Discrete Element Modeling of Soil-Geogrid Interactions

Alina Irsainova, B. Eng

Submitted in fulfillment of the requirements

for the degree of Masters of Science

In Civil Engineering

School of Engineering

Department of Civil Engineering

Nazarbayev University

53 Kabanbay Batyr Avenue,

Astana, Kazakhstan, 010000

Supervisors: Jong-Ryeol Kim, Sung Woo Moon

December 2018

Abstract

Geogrids are the geosynthetics of choice for soil reinforcement applications. To evaluate the efficiency of geogrid reinforcement, several methods are used including field tests, laboratory tests, and numerical modeling. Field studies consume a long period of time and conducting these investigations may become highly expensive because of the need for real-size structures. Laboratory studies present also significant difficulties: large-size testing machines are required to accommodate realistic geogrid designs. The discrete element method (DEM) may be used as a complementary tool to extend physical testing databases at a lower cost. Discrete element models do not require complex constitutive formulations and may be fed with particle scale data (size, strength, shape) thus reducing the number of free calibration parameters. The thesis reviews the different approaches followed to model soil-geogrid interaction in DEM and presents preliminary results from pull-out and triaxial conditions. Moreover, a numerical model of triaxial test with or without geogrid was developed and validated by laboratory test values that were provided by other researchers.

Acknowledgements

I would like to express my gratitude to my supervisors, Dr. Jong-Ryeol Kim and Dr. Sung Woo Moon. I am thankful for guidance and support they gave me throughout this research.

I am also thankful to Dr. Marcos Arroyo from Polytechnic University of Catalonia and his research team for helping and supporting me during my research stay in Spain.

I would like to thank faculty and staff of Nazarbayev University, especially Civil Engineering Department for giving me opportunity to study master program and guiding and educating me during this period.

Table of Contents

List of Abbreviations

DEM	Discrete Element Method
FEM	Finite Element Method
ICI	Investment Company Institute
FHWA	Federal Highway Administration
AASHTO	American Association of State Highway and Transportation Officials
M-E PD	Mechanistic-Empirical Pavement Design
FEA	Finite Element Analysis
PDE	Partial Differential Equation

List of Figur

List of Tabl

Chapter 1 – Introduction

1.1 Overview

The thesis includes a description of the research performed for the soil-geogrid interaction study. The literature review covers the introduction to geosynthetics and in particular to geogrids, to main functions that geogrids offer and to the application of geogrids in civil engineering. In order to evaluate the efficiency of the reinforcement by means geogrids, several methods are used including field tests, laboratory tests, and numerical modeling. The thesis focuses on the numerical method, discrete element method (DEM), for investigation of soil-geogrid interaction. Moreover, it is essential to consider the validity of this approach which can be performed by contrasting the results obtained by laboratory tests and DEM.

1.2 Objectives

One of the principal objectives of the thesis is to model a geogrid reinforced soil through numerical simulation using DEM for the analysis of soil-geogrid interaction. The results of the separate study of interface behavior between soil and geogrid from triaxial testing will be provided. Moreover, the thesis will cover the comparison of the results obtained by numerical simulation and laboratory testing. Thereby, the main purpose of this thesis research is to analyze the efficiency of numerical modeling method. Furthermore, the thesis

aims to provide additional analyses including soil-geogrid interaction outputs for different shapes and sizes of geogrids.

1.3 Thesis Layout

The thesis is composed of two main parts: Literature Review and Study of Geogrid Reinforcement in Triaxial Conditions. The first part includes general information on geosynthetics and geogrids (Chapter 2), applications of geogrids in geotechnical engineering (Chapter 2) and methods to study of geogrid-soil interactions (Chapter 3) which includes a description of the methods that are applied to study soil-geogrid interaction as experimental and numerical studies. Types of experimental and numerical methods are included in further subsections.

As for the second part of the thesis, it includes the main findings of the study. Chapter 4 provides a general introduction of Discrete Element Method (DEM) and Yade code. Moreover, it presents set-up of the numerical model including a numerical representation of soil and geogrid. The following chapter (Chapter 5) is on the feasibility study of proposed model which provided trough pull-out test. The purpose of this chapter is to make sure that the model is functional by means of a simple numerical test. In Chapter 6, the results of a triaxial test are given including calibration and validation tests. In the last chapter (Chapter 7) the conclusions of the thesis are provided including main investigations and future work.

Chapter 2 - Literature Review

2.1 Geosynthetics and Geogrids

Geosynthetics is defined as synthetically manufactured products used with soil, rock, and earth so overcome civil engineering problems. Geosynthetics can be used in a wide spectrum of fields such as transportation, geotechnical, environmental, hydraulics, and private development. Basic functions that geosynthetics offer are separation, reinforcement, drainage, filtration, containment, etc. Geosynthetics can be subcategorized into geotextiles, geogrids, geonets, geomembranes, geosynthetic clay liners, geopipe, geofoam, and geocomposites [1].

Geogrids are one of the types of geosynthetics that are growing in usage. Its structure consists of plastic ribs forming big apertures. Due to its open-like structure, it can be used only for reinforcement and stabilization. Transverse and longitudinal ribs of the geogrids are manufactured from high-modulus polymers; therefore, the strength of geogrid ribs is higher than the strength of geotextiles. Transverse members of the geogrids serve as an abutment or anchor due to their location parallel to the face of a structure as it can be seen in Figure 2.1. Therefore, the main function of longitudinal ribs is considered keeping the transverse ribs in position [2].

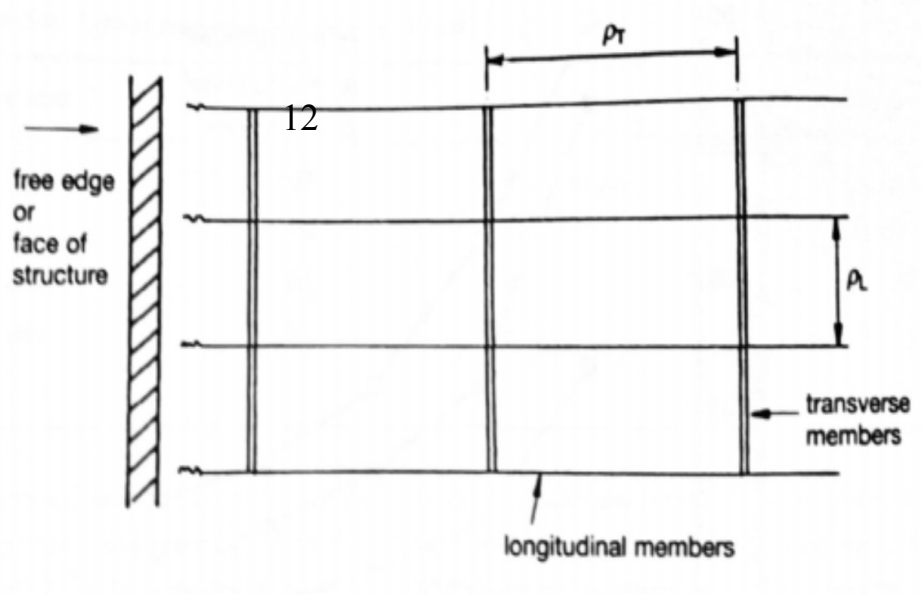


Figure 2.1: Grid reinforcement [3]

Unitized (homogeneous) geogrids were primarily produced in the United Kingdom by Netlon Ltd.. Afterward, Tensar corporation brought them to North America in 1982; whereas, geogrids with a similar design were manufactured by Tenax in Italy and spread into Asia. Using polypropylene-coated polyester fibers for bundles as reinforcing component was developed by Investment Company Institute (ICI) 1980 leading to the development of polyester yarn geogrids [1].

2.2 Advantages

The physical properties of geogrids that are used in describing them include dimensions of apertures and ribs, a density of the material, out-of-plane bending stiffness and in-plane torsional stiffness. Dimensions of geogrids can be measured straightforwardly, whereas density is dependent on the material of them. Out-of-plane and in-plane bending stiffnesses are measured by applying tests. They are required to evaluate how stiff or flexible geogrids are.

For the mechanical properties, single rib and junction strength, wide-width tensile strength, shear strength, anchorage strength from soil pullout and wall connections are included. Single rib's strength is measured by applying tension to a rib until failure occurs, whereas junction strength can be evaluated by applying force to longitudinal rib while transverse ribs are fixed. Wide-width tensile strength is measured by testing larger specimen under tension with alteration of rib number in the direction of a geogrid's width. The resultant value gives a strength value per unit width. For the determination of shear strength, geogrid slides over soil block under normal stress, and the maximum shear stress obtained is its shear strength [1].

Geogrids can be subdivided into uniaxial and biaxial based on the direction of stretching. In uniaxial geogrids, longitudinal ribs have higher tensile strength compared to transverse ribs. In biaxial geogrids, the stretching is provided in both directions; therefore, tensile strength is equally distributed in both directions. Moreover, they are categorized into three types: unitized polyolefin, coated yarn, and polyester rod. Unitized polyolefin geogrids are a homogeneous set of intersecting longitudinal and transverse ribs. Coated yarns tend to be more flexible and bundles of polyethylene coated polyester fibers are used. Polyester rods are manufactured by laser or ultrasonically bonding together rods in a grid pattern [3].

The opening size of geogrids is sufficient enough to allow soil communication and particles interlocking. The stiffness of a geogrid's ribs is higher than in geotextiles. It provides higher strength of soil mass and higher load bearing capacity. Geogrids are helpful in retaining soil from erosion. Moreover, the use of geogrids in construction reflects other advantages such as ease of construction, high durability, resistance to environmental issues, availability of the material, and low cost [4].

Geogrids are widely used in construction, and main areas that they are applied in are the construction of the pavement, construction of retaining walls, and stabilization of soil under the foundation.

2.3 Applications of Geogrids in Geotechnical Engineering

The applications of geogrids can be divided into main three categories: applications in road constructions, application in retaining walls, slopes and embankments, and application in foundation soils. Usage of geogrids in the construction of a road is to be described in the paragraph below.

2.3.1 Geogrids in Unpaved Roads

Geogrids can be useful not only in terms of reinforcement but also separation for roads construction. However, the primary function of geogrid is reinforcement for pavement. Geogrids' traditional applications in pavement include subgrade stabilization, aggregate base reinforcement, and asphalt concrete reinforcement. For an application of the first two functions above,

geogrid is required to be placed at the bottom of the pavement base layer with a thickness less than 14 in. In the case of base layer thickness is greater than 14 in., the geogrid is to be placed in the middle of the layer [5].

Performance of Unreinforced Unpaved Structure

In unpaved roads, a base layer carries a considerable number of loads in the parts where there is a temporary road. For these temporary roads, a surface rutting of such significant size as 50-100 mm is normally allowable if they can be easily reinforced by a supplement of material. Nevertheless, significant rutting in the subgrade layer may lead to the contamination of a base and subgrade material. This may need a replacement of a base layer soil. The mechanisms that may lead to surface rutting are:

• Aggregate compaction of the base layer and/or soil of subgrade under cyclic traffic loading.

• A failure of bearing capacity in base and subgrade layer from tangential and normal stresses due to the traffic loads.

• A failure of bearing capacity in base and subgrade layer from deterioration of soil, decreasing of base layer thickness due to the contamination of base, due to the cyclic traffic loads.

• Because of a cluster of incremental plastic strains by repeated loading, availability of lateral displacement in base and subgrade materials [6].

Influence of Geogrids on Subgrade Material Performace

The improvement of the subgrade material behavior due to geogrids is reached through the following mechanisms:

• *Preventing the local shear of a subgrade.* In the case where roads are not reinforced, small shear takes place because of vertical stress surpassing the elastic limit of road aggregate. The material of the base layer interrupts the subgrade; thus, leading to the permanent deformation. Shear zones increase, the base layer deteriorates, vertical stress value grows, and surface rutting appears due to the cyclic traffic loads. Ultimately, the limit of plasticity or final bearing capacity of the subgrade is achieved which causes a full shear failure. An appropriate reinforcement placed between the base and subgrade can help to avoid a formation and increase of local shear [6].

• *Load distribution improvement*. Placement of geogrid as reinforcement elevates the distribution of loads in the base layer and diminishes normal stress acting on the subgrade. Thereby, the safety factor due to bearing capacity failure is elevated.

• *Decrease or reorientation of shear stress at subgrade.* Reinforcing with geogrid at the interface among layers brings an advantageous result of carrying the shear stress due to traffic loads. It is essential to realize that the shear stress from the base layer transferred to the subgrade can be directed inward or outward. According to the theory of plasticity, subgrade bearing capacity

decreases with the outward direction of shear stress and bearing capacity increases due to the inward shear stress.

• *Tensioned membrane effect*. Rutting of a subgrade layer may lead to heaving when the soil begins to shear. A layer with geosynthetics placed at the interface will become wavelike that may be tensioned or stretched. When a flexible fabric takes a wavelike shape, normal stress acting on a concave side is larger than normal stress on a convex side. This action is named the "tensioned membrane effect" [6].

2.3.2 Geogrids in Flexible Pavements

There are three reinforcement mechanisms due to the geogrid application: (1) lateral resistant, (2) improved bearing capacity, (3) tensioned membrane effect. The first mechanism (Fig. 2.2 (a)) concerns restricting material from lateral flowing under applied load leading to an increase of base material's modulus. As a result, vertical stress distribution applied to a subgrade improves and vertical strain on the top of the subgrade reduces [7].

The second mechanism is described by the shift of the failure envelope of the pavement from the weak subgrade to the relatively strong base course material. Figure 2.2 (b) depicts the mechanism with failure envelope shift.

The third mechanism of tensioned membrane effect (Fig. 2.2(c)) refers to improved vertical stress distribution resulting from tensile stress in a membrane.

Moreover, Table 1 shows the minimum requirements for geogrids to be used in the subgrade layer of roads. According to [1], using geogrids for reinforcement of subgrade provides a set of benefits including initial stiffness increase, reduction of long-term vertical and horizontal deformation, tensile strength increase, diminishing cracking, etc.

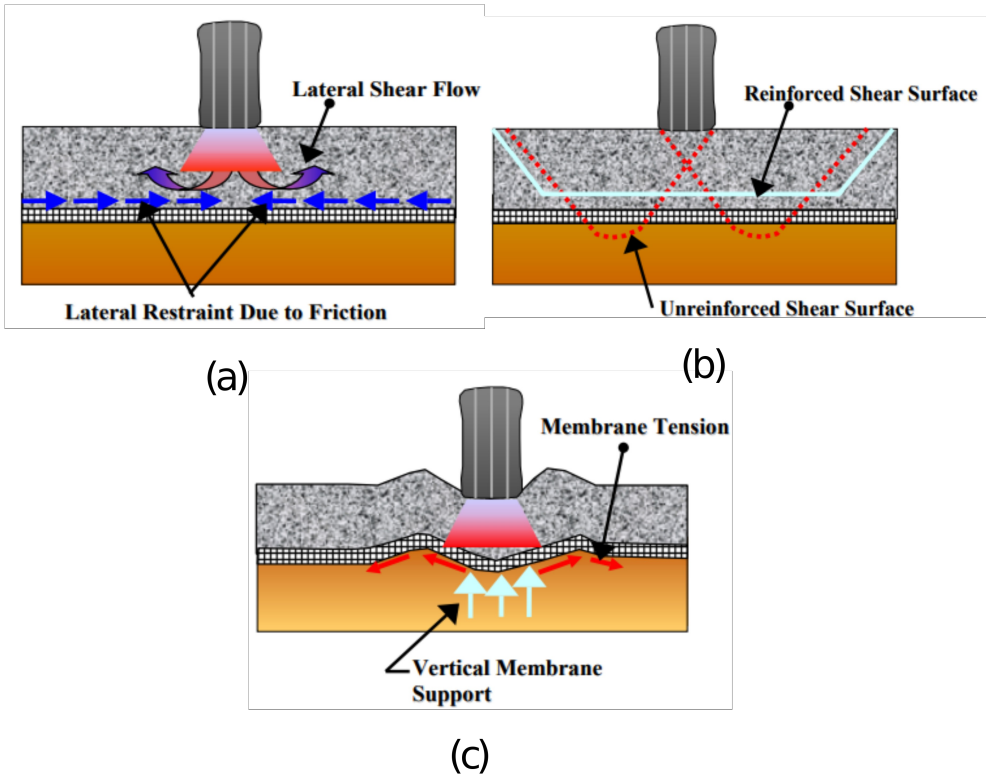


Figure 2.2: Reinforcement mechanisms: (a) lateral resistant mechanism, (b) improved bearing capacity mechanism, (c) tensioned membrane effect reinforcement [8]

2.3.3 Geogrids in Slope and Wall Reinforcement

In this part, applications of geogrids for slope and wall reinforcement is to be considered.

The soil mass that is reinforced by a geogrid can be referred to as mechanically stabilized earth (MSE). There are several methods in the reinforcement of walls by means of geogrids [9]:

- *Wraparound facing* represents soil-filled bags forming a facing of the slope and rest of the soil is behind the facing as a backfill. Moreover, the soil in the backfill is reinforced by geogrids. This method is effective to walls and slopes at the angle of up to 80 and with the height between 3 and 50 m. Figure 2.3 depicts wraparound facing of a retaining wall.
- *Timber facing* refers to geogrid located between timbers and held by friction and/or when geogrids tied to large timbers by batten strips.
- *Articulated precast concrete panels* consist of a set of precast concrete panels with inlets required for geogrid attachment.
- *Full height precast panels* represent a structure of precast concrete panels being supported prior to backfill completion.
- *Cast-in-place concrete panels* are walls with wraparound geogrids, and panels are covered around them.
- *Gabion facings* represent baskets made of steel with rocks inside, and geogrids are placed between the baskets.
- *Welded wire-mesh facing*s are analogous to gabion facings, but with no rock fills.
- *Masonry block faced walls* are when geogrids are located between the blocks and held by friction or/and pins, keyways.

Figure 2.3: Wraparound facing [10]

One of the widespread wall system's structure represents segmental blocks facings. There are three basic components: precast concrete segmental blocks, geosynthetic reinforcement, and soil backfill. Figure 2.4 depicts typical methods of connection between geosynthetics and concrete facing blocks. Regarding the size of the blocks, it is suggested to use small size blocks to achieve the facing stability of the wall. Facing stability guarantees resistance to cracking. Moreover, geogrid reinforcement is required for the reinforcement of backfill as well. Strength and spacing requirements for the design of reinforcement varies according to backfill material properties and wall dimensions [11].

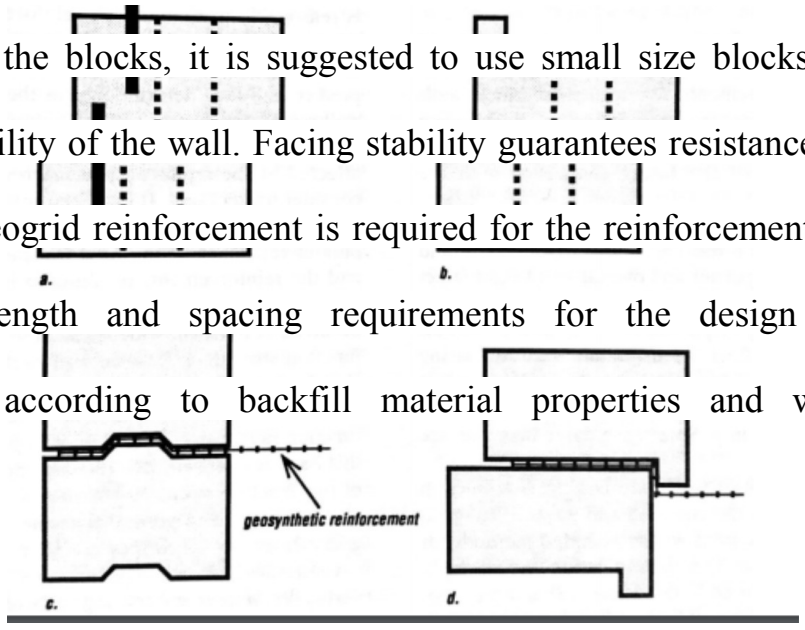


Figure 2.4: Connection of geosynthetic reinforcement and segmental blocks (Alen, 1992)

Chapter 3 – Methods of Studying Soil-Geogrid Interactions

3.1 Experimental Studies

3.1.1 Field Testing

One of the methods of obtaining data on geogrid-soil interaction and geogrid reinforcement efficiency is experimental studies. They involve testing of real life soil and geogrid. Experimental studies can be divided into field testing and laboratory testing. Field testing is provided in an actual environment of soil reinforcement. For example, investigation of unpaved road reinforcement can be done within the road. Moreover, the field tests may involve full-scale experiments which simulate actual behavior of geogrids. In addition, it is challenging to make adjustments for different environmental conditions. In the case of field testing, the cost of data gathering is high and, consequently, only the number of tests typically conducted is limited. Thus, the testing scope of the research is limited [13].

3.1.2 Laboratory Testing

There are numerous laboratory tests dedicated to the study of mechanisms of soils reinforced geosynthetically, reinforced flexible pavements particularly. The main purpose of the tests was quantifying the mechanism of soil and geosynthetics interaction. It was provided by the estimation of index properties

of geosynthetics or by repeating the field conditions. One of the most significant field conditions that are needed to be repeated is the shear of the interface which is developed from geogrid interlocking when it is placed in or below the base layer of the pavement. Based on the used method, the laboratory tests are categorized into: confined and unconfined tests. In confined laboratory tests, the properties of geosynthetics are measured in the soil confinement, whereas in unconfined tests the properties are estimated in-air [14]. As confined conditions are subjected to the geosynthetics in the base soil of pavements, only confined tests will be considered.

Confined tests

Geosynthetics are applied in the base layer confined by soil under repeated traffic loading. These field conditions cannot be replicated under the unconfined tests. The confinement of soil around geosynthetics is dependent on macroscopic properties and structure of geosynthetics, soil properties, and on soil-geogrid interaction. According to the study sponsored by FHWA, the response of geosynthetics under unconfined conditions is excessively conservative, while under confined conditions, the response is considerably improved.

In order to quantify shear stress development ability in soils under the repeated loads, the cyclic triaxial test (Fig. 3.1) was applied. M_R, the resilient modulus of soil particles was determined by this test and was applied to the M-E

design [15]. Furthermore, the following study [16] modified the test to measure the deviation in M_R and to quantify the behavior of permanent deformation with the inclusion of geosynthetics to the coarse aggregate layer. As a result of the tests, it was indicated that geosynthetic reinforcement does not influence the resilient modulus of soil. However, it reduces the permanent deformations of pavement considerably. Moreover, reinforcement of soil increases the stiffness of soil particles in areas near to the geosynthetic.

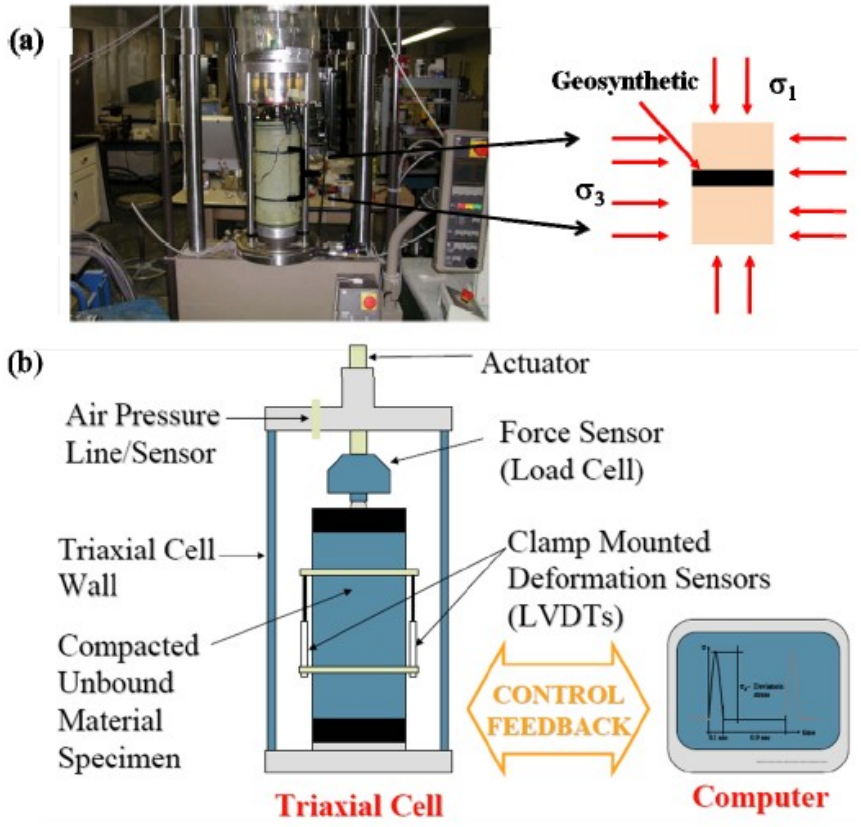


Figure 3.1: Cyclic triaxial test (a) Test equipment; (b) Schematic of test setup [17]

The test methods for evaluating geosynthetically reinforced pavements under confined conditions are summarized. The confined laboratory tests are able to develop quantification of the soil-geosynthetic interaction behavior;

however, they are more expensive and consume longer time compared to the unconfined laboratory tests. Confined tests are capable of measuring soil reinforcement performance by quantifying reduced deflections and increased modulus of confinement. Moreover, the confined test results are qualified as more appropriate for AASHTO and M-E design methods. Numerous studies showed that reinforcing soil leads to the improvement compared to the systems without geosynthetics. However, there are disadvantages of the proposed confined test methods. The confined tests need special equipment and the variability of the results was excessive. To summarize, confined test methods are more appropriate in order to quantify the enhancement of pavements due to geosynthetics compared to the unconfined test approaches. Relying upon this conclusion, it may be referred that laboratory test approach should:

a) be able to replicate lateral restraint mechanism;
b) output required parameters for M-E design;
c) produce solid stability of the testing results;
d) apply parameters distinguishing separate geosynthetics' performance;
e) be sensitive for small displacements;
f) be easy to perform.

3.2 Numerical Studies

The design of soil-geogrid system involves understanding the behavior and the interaction between various materials (asphalt, base course, subgrade, geosynthetic reinforcements). Current design methods are empirical in nature. This is partly because of the inability of available analytical tools to predict the time-dependent behavior of pavements under actual traffic loads [18]. However,

numerical methods can be used to provide insight into the mechanics of pavement systems. The most commonly used methods are the finite element method (FEM) and the discrete element method (DEM).

3.2.1 Continuum-Based Models

Continuum models are commonly used to simulate the behavior of the soil. The two methods that are usually applied are the finite difference method (FDM) or the finite element method (FEM).

FEM is a numerical method applied for resolving engineering problems and problems in mathematics. In addition, it may be adverted as Finite Element Analysis (FEA). An alternative to this numerical method, analytical solution, normally needs the resolution for problems with boundary values in PDEs. In FEM, the problem results are formulated in the algebraic equations system. FEM divides the problem into several smaller and simpler elements (finite elements) in order to resolve the problem. Thus, simple equations modeling these smaller parts are collected into a bigger equation system which is modeling the whole problem [19]. To clarify, a system of equations are applied to model a problem in FEM; whereas, simple equations represent only one finite element of the problem. By solving every simple equation one-by-one, simple problems are solved. The solutions of all simple equations will form a solution of the system of equations, thus, resolving the big problem.

In order to test a performance of geosynthetics applied to flexible pavements, FEM was used in several studies. Surface deformations of the reinforced pavements under the employed load were obtained as a result from finite element investigations. The surface deformation values of reinforced and unreinforced systems were compared. Finite element model with representative sections for pavements can be seen in Figure 3.2.

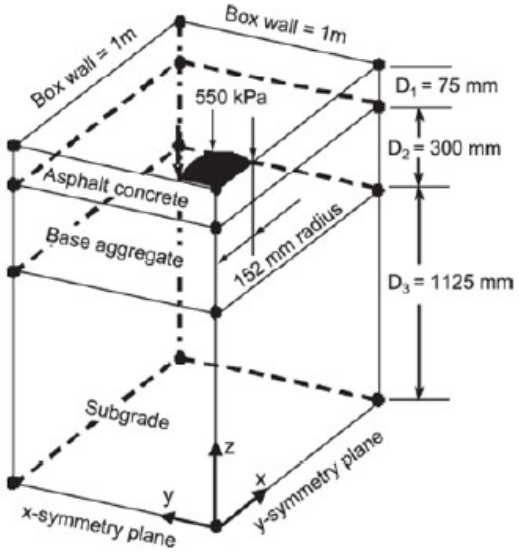


Figure 3.2: Flexible pavement: finite element model [19]

3.2.2 Discrete Models

The Discrete Element Method (DEM) has been latterly applied for modeling of geogrid-soil interrelation. Especially, evaluation of the interlocking quality of geogrid within base aggregate has been studied. Performance of soil and geogrid interaction is defined by deformation behavior and load transfer mechanism by this approach. DEM is a numerical method that includes tracking

the movement of each particle and every structure of a model system. In the DEM model, every particle-particle and structure-particle contacts and their contact laws are determined. During the numerous computation cycles that contain from small-step iterations, between some structures or bodies total loss of contacts and creation of new contacts may occur. Moreover, the forces at the contacts are computed regarding the contact force model [21].

Originally, DEM was introduced by Cundall and Strack [21]. DEM is applied in order to define the performance of particulate material by the interactions of separate particles. Discrete particles in the model have shapes such as spheres (3D) and disks (2D). The elements are considered rigid, although minor overlaps are permitted at the contact points. At these points, the contact model type that is used is a soft model. Furthermore, contacts are permitted to be lost when the overlap between the elements stop being available. DEM is capable of estimating values of micromechanical properties which can be determined also by experimental studies; however, DEM involves interactions at the particle scale [22]. This method defines discrete properties as particle displacements, particle stresses, forces on particles and velocities of particles. The boundaries of a model can be defined by applying wall elements, where displacements are employed. Contact model assigned in DEM influences on which parameters need to be calibrated. Moreover, through simulations by DEM dynamic data can be obtained including trajectories of particles and

transient forces on separate elements that are highly difficult to measure by experiments [22].

A study involved a DEM model for small part of reinforced with geogrid pavement [23]; and as a result, application of geogrids provided locked-in stresses during the phases of placing, compaction, and loading. Moreover, soil located above the geogrid became stiffer. However, further studies are needed to determine a relation between field behavior ad DEM pullout test results.

According to [23], application of DEM method issues the results defining the mechanisms of geosynthetics as locked-in stresses underwent in service life period of the pavement. According to the results, reinforcing the pavement with geosynthetics leads to the base layer to stiffen. Therefore, DEM may be fully applied for quantification of the interlocking effect of geogrid and for fundamental research. Nevertheless, the continuum approaches should be applied for further analyses due to the lack of sufficient studies on the efficiency of DEM for other analyses.

DEM models provide a description of interrelations of separate particles. As the DEM models are developing, they are being utilized for simulating granular soil and other materials. Moreover, it is extremely beneficial when the study involves erosion as a problem or when significant deformations take place.

Numerical Representation of the Granular Soil

In order to model granular soil by the software, soil type should be described. It is possible to define soil that requires reinforcement or soil type that is used for certain applications (e.g. ballast for railroads). All the required design parameters of soil will be chosen according to the characteristics of soil used for triaxial testing.

The discrete particles can be modeled as a single grain of soil, or for the purposes of saving time for model simulation, several soil grains can be represented by a single particle. In DEM, it is possible to model the shape for soil particles and they include spheres, ellipses, and even complex shapes can be generated. One of the instances of creating irregular particle shapes is obtained by connecting and overlapping several regular shaped particles [24].

A number of particles in the model also can be controlled and it may be thousands to millions of particles. Moreover, diameters of the grains can be selected according to the sample's particle size distribution. Furthermore, the soils can be defined as saturated or unsaturated depending on the liquids or gas content in the voids. The soil structure can be modeled as loose, medium or dense which depends on the packing of the particles. Other soil properties such as elastic modulus, Poisson's ratio, and friction angle values can be input.

Numerical Representation of the Geogrid

Modeling of geogrids can be categorized as soil-inclusion problems. Applying finite element method (FEM) for such case is widely practiced [25], [26]. However, using FEM to model soil-inclusion problems faces difficulties in the definition of crucial parameters that represent grid-soil interaction. Application of discrete element method (DEM) may be useful, particularly for cases involving large sized granular materials. There are several studies that describe a conventional method of modeling representing soil and soil inclusion as rigid spherical particles [27]. Some studies develop soil-inclusion model by using a mix of methods: where the soil was modeled by discrete element (DE) and geogrid was modeled by finite element (FE) [28]. A summary example of geogrid modeling methods is given in Table 3.1.

Table 3.1: Summary of geogrid modeling methods

#	Modeling approaches	Applications	Refs.
1	Model by using FEM: where both soil and geogrid was modeled by finite element	Pull-out behavior of the model was investigated	Sugimoto and Alagiyawanna [25]; Khedkar and Mandal [26]
2	DEM model representing soil and soil inclusion as rigid spherical particles.	Cyclic triaxial loading simulation with spherical ballast particles.	McDowell et.al [27]
3	DEM-FEM models: where the soil was modeled by spherical discrete element and geogrid was modeled by finite element (FE).	A pull-out test was performed to define relationships between pull-out force and displacements	Tran et. al [28]
4	DEM model by representing geogrids as deformable cylinders according to the Minkowski sum concept and representing soil as spherical particles.	A pull-out test was performed in order to check the effectiveness of the model.	Thoeni et. al [29]

There are different techniques of modeling the geogrids such as modeling as soil particles with the properties of geogrid material or modeling as a set of rigid meshes applying the geogrid dimensions. Modeling the geogrids as the rigid element can be efficient as it decreases computation time of simulation and makes the calibration procedure easier. Moreover, the shapes, dimensions, the aperture sizes of geogrids should be modeled according to the tested specimen properties. The characteristics for the geogrids as tensile strength, shear strength, normal stiffness, shear stiffness, and friction angle are chosen in the DEM software. More details are to be evaluated during the research process [20].

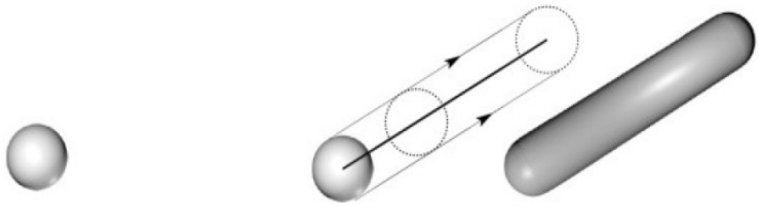
Chapter 4 – Numerical Model Set-up

4.1 Numerical Model

A method to model geogrid-soil interactions by representing geogrids as deformable cylinders according to the concept by Minkowski sum and representing soil as rigid spheres has been recently proposed by Thoeni et.al [29]. Main components of Minkowski sum include rigid spheres (Fig. 4.1a) and cylinders represented by sphere and line (Fig. 4.1b). Each rib of the grid can be modeled by one or more cylinders depending on the geometry of a grid.

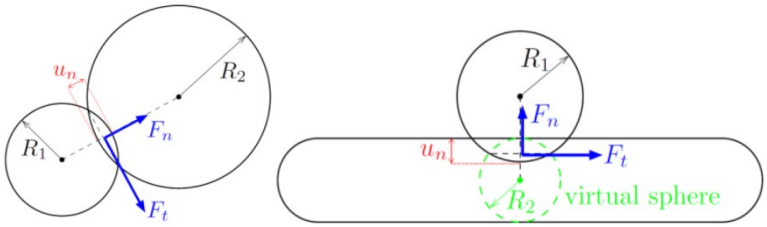
The possible interactions in the model are the rigid sphere-rigid sphere, rigid sphere-cylinder, and cylinder-cylinder. An idea of a virtual sphere inside the deformable cylinders was presented in order to deal with all possible interactions in the model. The same virtual spheres can be created at every contact nodes inside a cylinder (Fig. 4.2b). Therefore, the contacts between each component are treated as sphere-sphere interconnection allowing to use basic mathematical formulation for contact forces (Fig. 4.2). In this model, the cylinder deformation can be defined by the deformation of cylinder nodes. The mass of a cylinder is concentrated on its nodes. Each sphere particle in the model behaves as a rigid object, while each cylinder behaves as an almost rigid element with the deformation influenced by the conformity of the contacts

between a cylinder and other elements. If this conformity is neglected, the cylinders may appear as rigid but deformable in the longitudinal direction.



(a) (b)

Figure 4.1: Minkowski sum components: (a) sphere and (b) cylinder [29]



(a) (b)

Figure 4.2: Contact forces between components: (a) sphere-sphere and (b) sphere and virtual sphere of a cylinder [29]

4.2 Inter-particle Contact Law

A linear contact stiffness law and Mohr-Coulomb friction were used in the software to describe inter-particle interaction. This law implements the classical linear elastic-plastic law from [21]. The normal force is (with the convention of positive tensile forces) $F_n = \min(k_n u_n, 0)$, where u_n is the normal distance between two spheres. The shear force is $Fs = k_s u_s$, where u_s is the relative shear displacement. The plasticity condition defines the maximum value of the shear

force: $F_s(max) = F_n\tan(\varphi)$, with φ the friction angle. The linear contact model stiffness is derived from the normal and shear stiffness k_n and k_s assigned to the contacting objects. Linear contact model represents two contacting objects to be in series; hence, normal secant stiffness of contact is defined by following equation:

$$k_n = \frac{k_{n1}k_{n2}}{k_{n1}+k_{n2}} = \frac{2E_1R_1E_2R_2}{E_1R_1+E_2R_2} \quad (4.1)$$

Where, k_{n1}, k_{n2} = normal stiffness of contacting objects .Whereas shear tangent stiffness of the contact is defined by:

$$k_s = \frac{k_{s1}k_{s2}}{k_{s1}+k_{s2}} = \frac{2E_1R_1\nu_1E_2R_2\nu_2}{E_1R_1\nu_1+E_2R_2\nu_2} \quad (4.2)$$

Where, k_{s1}, k_{s2} = shear stiffness of contacting objects, E_1, E_2 = Young's modulus, R_1, R_2 = radii of the contacting spheres, and ν_1, ν_2 = Poisson's ratio.

When a soil particle contacts a grid component the same formulas apply, but the grid is assigned the radius of the virtual inscribed sphere (see Fig. 4.2).

4.3 Yade DEM Code

4.3.1 DEM Code for Pull-out Test

Yade DEM code for pull-out of geogrid can be referred in Appendix A. This code consists of several main parts including definition of materials, introduction of engines, creation of the geogrid, creation of soil and walls

constraining it, application of particle growth process, application of pull-out force and confining pressure, and obtaining the results.

First of all, materials of objects are defined in terms of macroscopic parameters such as Young's modulus, Poisson's ratio, density, and friction angle. The materials that were used are soil particles, geogrid material, and material for walls surrounding the soil.

The next step is the definition of engines. All the processes and interactions are defined in the engines section. In the InsertionSortCollider() the elements for approximate collisions are detected. In the code, they are Bo1_Sphere_Aabb() for sphere particles of soil and geogrid, Bo1_Box_Aabb() for the surrounding box formed from walls, and Bo1_GridConnection_Aabb() for the connection between the geogrid nodes. Furthermore, the possible interactions are defined in the InteractionLoop(). There are two functors: Ig2 where interactions between two elements or shapes are considered and Ip2 where interactions of two materials are considered. Moreover, the laws acting in the simulation are defined with the help of Law2 functor. In the code, CundallStrack law was applied.

The next part of the code is a creation of a geogrid. In Appendix A, the code lines for triangular and rectangular grids are presented. Firstly, the main parameters of the grid are defined including length, width, number of nodes in each dimension, and radius of the grid nodes. Afterward, the nodes are created

by nodesIds.append() function at the locations required. At last, the connections between the nodes are developed by using O.bodies.append(gridConnection()) function. In the code, the required connections between the nodes were created by indicating the required distance between the nodes.

Furthermore, the creation of soil and walls was performed. Walls were created by means of wallIds function and the soil was created by pack.inParallelepiped() function allowing to create soil pack in the assigned parallelepiped coordinates.

Next step is an application of a servo-control system code for growing or shrinking the soil particles in order to achieve the required stress inside the box. The system measures the stress on the top wall by O.forces.f() divided into the area of the wall. The grow factor by which the particles grow is depended on the stress value we want to control and the actual measured stress value. The servo-control system stops when the grow factor reaches the specified value.

Moreover, the pull-out force is applied by the application of velocity along the left side of the geogrid. As for the confining pressure application, another servo-control system was utilized. It also measures the force at the top wall where the pressure is applied and compares with the target force. The velocity is applied boundaryVel() function at the top wall. The value of velocity depends on the difference between the measured stress and the target stress.

The last part of the code is obtaining and recording the results. Pull out force is calculated by measuring the forces in each computational step at the nodes where the geogrid is pulled-out. Moreover, the pull out distance is calculated for each step by measuring the current position s1.state.pos[0] and subtracting from it the reference position s1.state.refPos[0]. The results of pull-out force and geogrid displacement at each computational step are saved by using plot.saveDataTxt() function.

4.3.2 DEM Code for Triaxial Test

Yade DEM code for the triaxial test can be referred in Appendix B. Triaxial code also consists of several parts including a definition of variables and materials, a creation of walls and geogrids, generation of particles, defining the engines, application of isotropic pressure, application of deviatoric pressure, and acquisition and saving the results. Some parts of the code are similar to the pull8-out test code. For example, the creation of walls and geogrids are the same as in previous code. Defining the variables and engines section is similar; however, other parameters were added to the triaxial test code including the number of spheres, friction degree for isotropic and deviatoric stages, strain loading rate, etc. Generation of soil particles with required particle size distribution was performed by the use of pack.SpherePack() function. Moreover, the engines are the same in O.engines=[] part including the InsertionSortCollider() and InteractionLoop() functions. However,

TriaxialStressController() engine was added which is able to control triaxial test processes. By the selection of stressMask value, it is possible to define in which axis stress needs to be controlled. It can be calculated by x*1 + y*2 + z*4, where x,y,z are the binary representation of the axes. For example, if the stress needs to be controlled in x and z directions, stressMask = 1*1 + 0*2 + 1*4 = 5.

The next section is the application of isotropic confining pressure. Confining stress value in three directions were defined by triax.goal1=triax.goal2=triax.goal3 functions. Afterward, deviatoric loading was performed where through triax.stressMask value the stress-controlled directions are defined. At last, the lines for recording and saving the results were written. Stress and strain values at each directions were measured by triax.stress[] and triax.strain[] functions respectively; they were saved by plot.saveDataTxt().

Chapter 5 – Feasibility Study of DEM Model

5.1 Pull-out test model

The discussed geogrid modeling technique with deformable cylinders needs to be analyzed in the test configurations. A pull-out test was initially modeled in order to observe the potential of the approach chosen. The contact properties of soil and geogrid material are assumed to be the same (Table 5.1). Note that, for simplicity, no rolling friction was included in the contact model. Square grid mesh of 9.5 cm x 9.5 cm dimensions with 1 cm openings was introduced to the model (Fig. 5.1a). The particle sizes used for the rectangular and triangular grid are 2 mm and 1 mm respectively; therefore, the diameters of cylinders are 2 mm and 1 mm. The number of nodes for the length and for the widths is 10 for rectangular geogrid. While in the case of triangular geogrid, the number of nodes in both directions was 8. The material type chosen for the grids is CohFrictMat with such properties as indicated in Table 5.1. Pull-out of the grid for cubic soil matrix with sides of 10 cm was performed applying a constant velocity of 0.06 m/s to the grid with the time step of $1e^{-04}$ s. The soil particles were generated inside the parallelepiped with indicated coordinates. The number of particles is unknown, although homogeneous particles with the radius of 2.5 mm and with the gap between particles of 1.25 mm were generated. The initial packing properties were as described above. However, in the simulation, the

particles grew or shrank according to the servo-control system. Grow factor was depended on the target stress inside the box and measured stress and was calculated for each loop. When grow factor reached a specified value, the servo-control system stopped. All numerical simulation was performed using Yade software [30].

Table 5.1: Summary of material properties

Parameter	Value
Young's modulus, *E*	5000 *kPa*
Density, ρ	2650 kg/m^3
Poisson's ratio, υ	0.3
Friction angle, *φ*	20˚

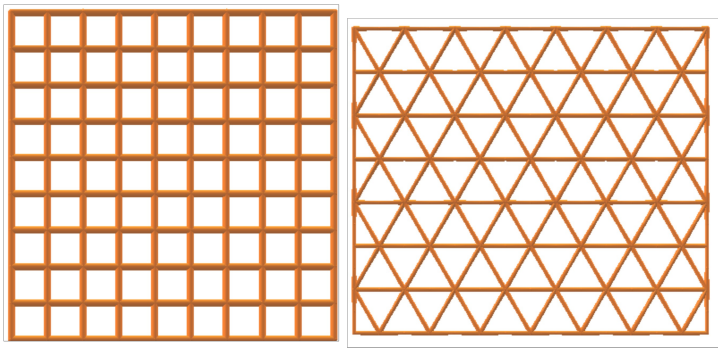


Figure 5.1: Geogrid mesh: (a) rectangular and (b) triangular

Before performing the pull-out, confining pressure was applied to the top wall. In order to reach the effect of confining pressure, particle growth was utilized with the confining pressure target of 10 kN. The particle growth system is described in section 4.3.1. Moreover, the application of pull-out velocity by means of servo-control is described in the section. After the generation of all required elements and application of processes, acquisition of results was considered. The pull-out force was calculated by measuring the forces at each 10

grid nodes at the left side of the grid for each computational step. The pull-out force is equal to the summation of forces at the grid nodes. Furthermore, displacement of the grid was also measured for each step and it is described in section 4.3.1. The required time for the simulation altered from 1 to 36 hours depending on the particle size, the smaller the size of particles results in the larger number of particles which leads to longer computational time.

Figure 5.2 shows a pull-out test at several stages, with the grid at different positions. Figure 5.2(a) shows when the displacement of a grid at the initial stage ($\Delta x=0$), while Figure 5.2 (b) and (c) present pull-out at the intermediate stage ($\Delta x=4.75$cm) and total pullout ($\Delta x=9.5$cm) respectively. The entrapment of soil particles can be observed as grey soil columns became mixed with green columns as geogrid is being pulled out. This is indicative of the interlocking properties of the grid because soil particles are captured in the grid openings and while it is pulled out, the particles move along the movement direction.

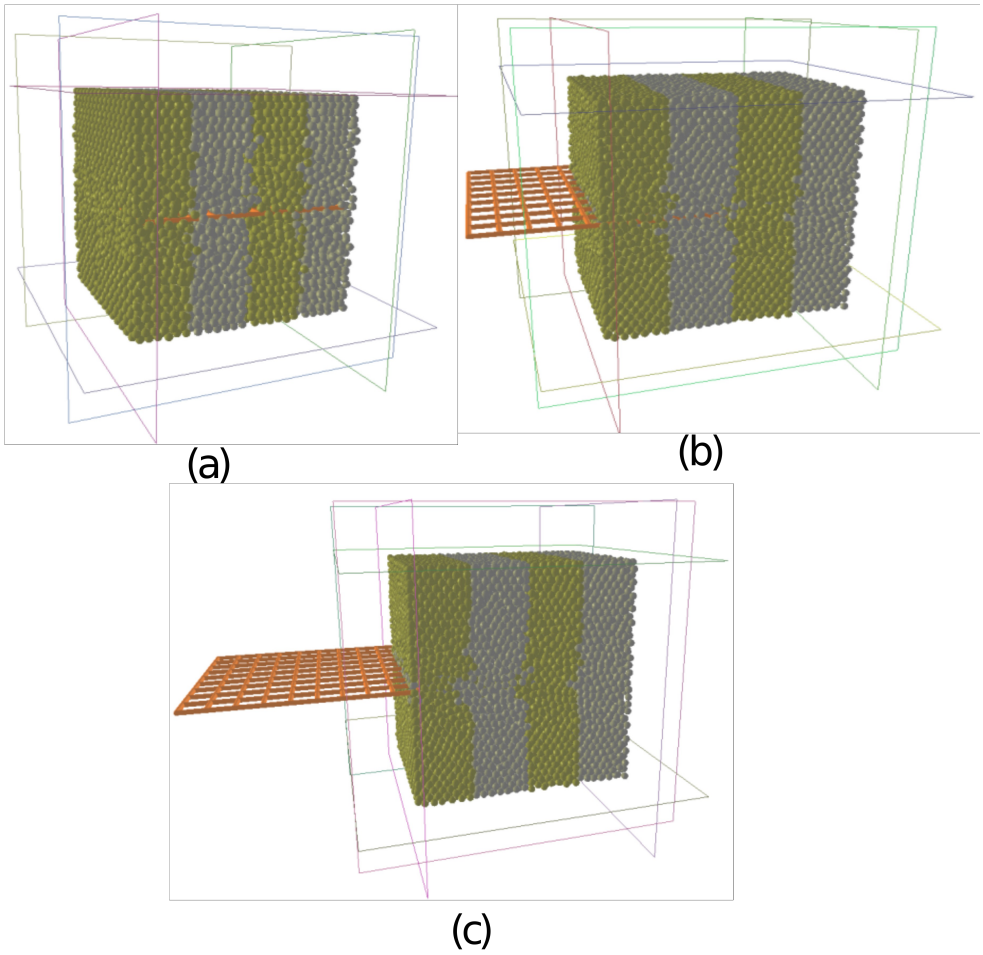


Figure 5.2: Pull-out of a grid: (a) Δx=0, (b) Δx=4.75cm and (c) Δx=9.5cm

5.2 Results of the pull-out test

A parametric study was conducted, in which the grid pull-out was performed under different conditions. The parameters explored included the vertical confining pressure at the top wall of the box, size of soil particles and the shape of the grid pulled out. The corresponding values of the parameters are presented in Table 5.2. It is noted that a uniform sized particle distribution was used in all cases. The schematic view of 10 cm x 8.7 cm triangular geogrid can be seen in Figure 5.1(b) and its geometry is more complex compared to the

rectangular. Triangles of the grid are equilateral with the sides of 1.43 cm and vertical components at both sides of the grid are 1.24 cm.

Table 5.2: Range of parameters considered in the study

Parameter	Value
Confining pressure, P (kPa)	75, 150 and 300
Radius of soil grains, r (m)	0.001, 0.0015 and 0.0025
Shape of geogrid	Rectangular and triangular

The resultant graph of pull-out force versus displacement is given in Figure 5.3 for various confining pressure conditions. This case considers rectangular geogrid with the middle (0.0015 m) soil particle dimension. The figure shows that pull-out response slightly increases with increasing vertical confining stress of 75 kPa, 150kPa, and 300 kPa. Average values of pull-out force are 20.11 N, 28.05 N and 31.20 N respectively. Interestingly there seems to be very little effect on the pullout force of the reduction of inserted grid length in the specimen; as long as there is one transversal rib in the box the pull-out force average is closely maintained.

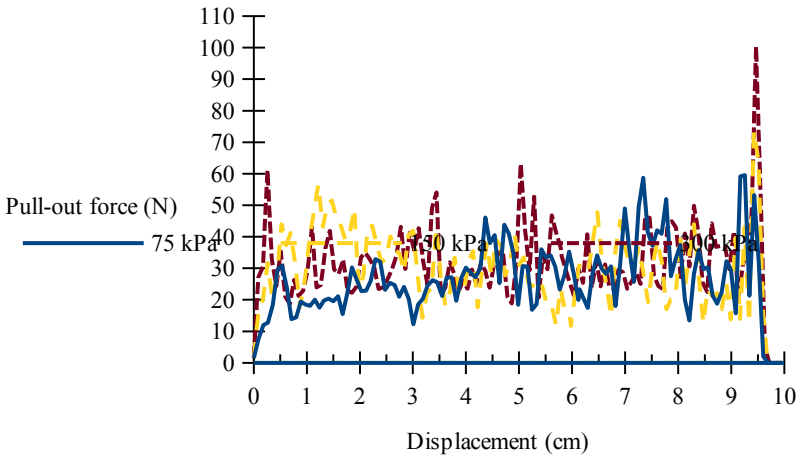


Figure 5.3: Pull-out response of square geogrid: r=0.0015 m

Another figure was built to illustrate the variance of a response of geogrid pull-out according to different shapes of geogrid. Figure 5.4 represents the case with vertical confining stress of 150 kPa and with soil grain radius of 0.0025 m. The figure shows that pull-out response for the rectangular grid is higher than the triangular grid case, at least until a large displacement has been achieved. Average pull-out forces for rectangular and triangular grid shapes are 36.66 N and 28.05 N respectively.

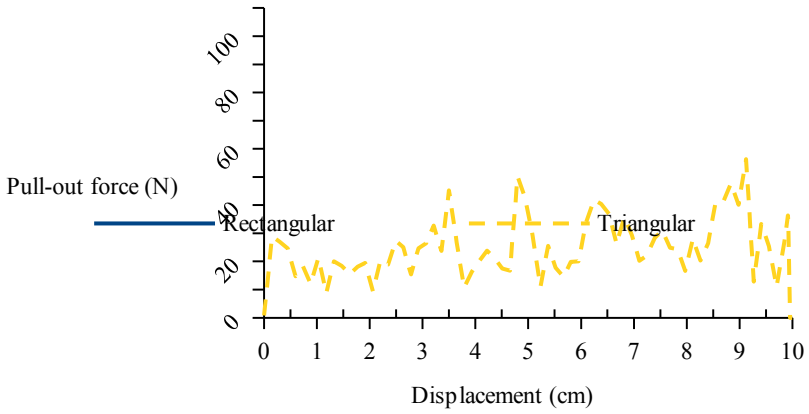


Figure 5.4: Pull-out of geogrids:P=150 kPa and r=0.0025m

Average pull-out force values for each case in the study are shown in Table 5.3. As expected, for all sizes of soil grains the pull-out response of the grid is higher for increased vertical confinement. In order to quantify variation of the pull-out response results, the coefficient of variation values was estimated. Table 5.4 includes the coefficient of variation for different particles sizes and for varied vertical confining stress. As a result, a variation coefficient decreases with the decreasing particle size for each confinement scenario. This indicates that a variation of the pull-out response is smaller for smaller grain size which leads to more precise results. One of the reasons may be that with smaller particles the model becomes more continuous compared to the model with larger particles. As the model becomes more continuous the variance of the results related to the average value reduces. Moreover, the size of particles may influence numerous properties of the soil material including flow and compaction properties. The larger particles may flow more easily and are likely to be compacted more easily due to the larger gaps between the particles in

comparison with the small particle model. Therefore, the variation of results for larger particles is higher than the variation of results for smaller particles. Furthermore, as the size of particles decreases, the number of particles increases in the model. Thus, the initial packing of the model with finer particles is denser. So, during the compaction phase, the particles have less movement in the small particle model. This may also affect the smaller variation of results.

Table 5.3: Average pull-out force values with different parameters

Confining pressure (kPa)	Radius of soil particles (m)			
	0.0025	0.0015	0.0010	Triangular shape (r=0.0025m
75	26.29	20.11	41.32	20.11
150	36.66	28.05	42.41	28.05
300	43.56	31.20	44.51	39.23

Table 5.4: Coefficient of variation of pull-out force with different parameters

Confining pressure (kPa)	Radius of soil particles(m)		
	0.0025	0.0015	0.0010
75	0.543	0.445	0.242
150	0.434	0.381	0.234
300	0.413	0.406	0.230

Chapter 6 – Application of DEM Model

6.1 Calibration tests

The proposed DEM model can be applied in a triaxial testing condition in order to observe triaxial test response with geogrid and to compare the results with no geogrid case. The first step is to define macroscopic parameters that will be applied to the DEM code in order to perform numerical simulations. Macroscopic soil parameters that are required for the code include Young's modulus, Poisson's ratio, and friction angle. These parameters were found by calibration of values with respect to the provided laboratory results for isotropic compression with 75 kPa pressure. For providing calibration, values of parameters were alternated and numerous combinations were simulated in order to obtain the best combination of parameters with the closest results to the laboratory test. The range of parameters used for calibration is given in Table 6.1. The table includes various Young's modulus values from 2500 to 5000 kPa, Poisson's ratio from 0.25 to 0.6 and friction angle from 28˚ and 34˚.

Table 6.1: Range of macroscopic parameters considered for calibration

Parameter	Value
Young's modulus, E (kPa)	2500, 3000, 3500, 4000, 4500, 4700, 4750, 4800 and 5000
Poisson's ration, ν	0.25, 0.30, 0.33, 0.35, 0.4, 0.45, 0.50, 0.55 and 0.6
Friction angle, φ (˚)	28, 29, 30, 31, 32, 33 and 34

The parameters of the numerical simulation need to be defined. The number of particles in the simulation was 15000. Soil particles were generated according to the particle size distribution shown in Figure 6.3. Number of computational steps assigned was 10000 with the time step of 1 s. Such a time step was chosen in order to reduce the time of simulation. For example, each simulation of data presented in Figure 6.1 was performed around in 1 hour. The geometric parameters of the geogrid used for the triaxial test are the same as described in section 5.1. The detailed description of triaxial test simulation can be seen in section 4.3.2.

Furthermore, numerical simulation results were compared with the provided graph and the best combinations are shown in Figure 6.1. The best combinations are for the Poisson's ratio of 0.33 and friction angle of 32˚**.** In the figure, it can be seen that the response is closest to the laboratory test results for Young's modulus of 4500 kPa. Therefore, these values for the macroscopic parameters were selected for further validation of the model.

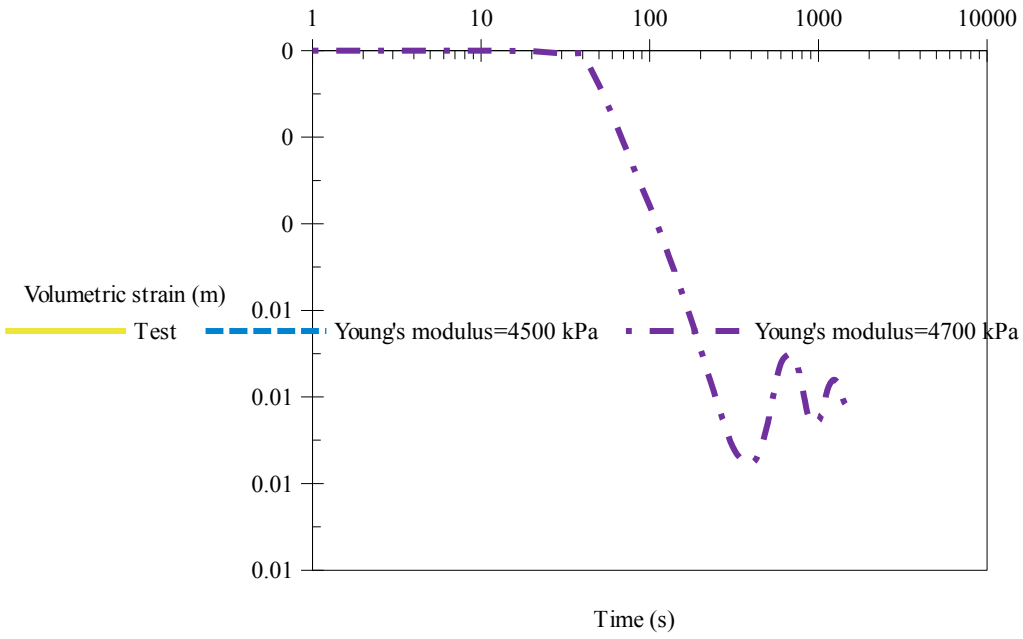


Figure 6.1: Best combinations of calibration for Poisson's ratio = 0.33 and friction angle = 32°

6.2 Validation tests

Validation simulation was performed with respect to the laboratory study provided by Polytechnic University of Catalonia. A triaxial test was performed in two stages: isotropic compression and deviatoric loading for different cell pressure cases. 75 kPa cell pressure case was used for validation tests. Schematic view of a triaxial cell is given in Figure 6.2 and its height is 0.6 m with a diameter of 0.3 m. Moreover, particle size distribution (PSD) of sample soil can be viewed in Figure 6.3. The same dimensions of the triaxial cell and the PSD of soil were used in the numerical simulations. Besides, the values of macroscopic parameters determined through calibration were applied for validation of the proposed DEM model.

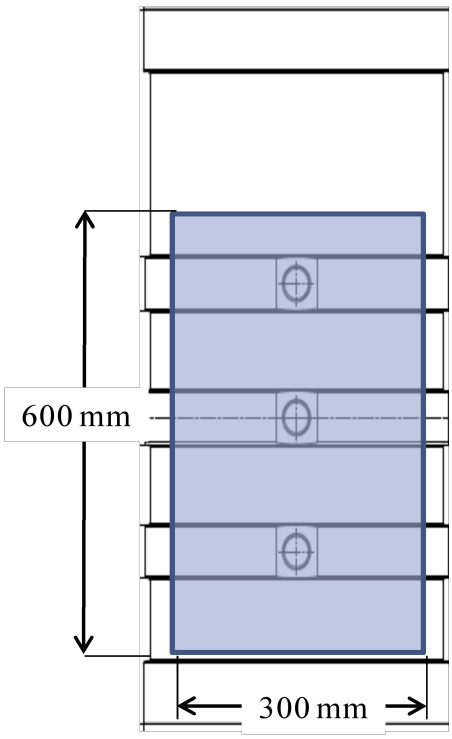


Figure 6.2: Triaxial cell dimensions

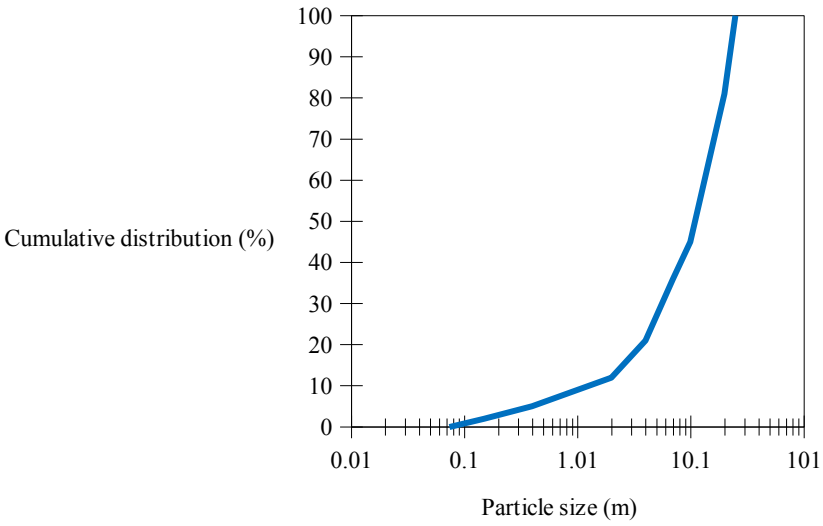


Figure 6.3: Particle size distribution of sample

Figure 6.4 provides the comparison of numerical simulation results with laboratory test results, volumetric strain vs. axial strain, for isotropic compression stage. Moreover, Figures 6.5 and 6.6 contain the comparison of

volumetric strain vs. axial strain and deviatoric stress vs. axial strain for deviatoric loading stage respectively. From Figures 6.4-6.6, it can be noticed that simulation values are similar to laboratory test values. Hence, it can be concluded that the numerical model is validated and ready for further simulations.

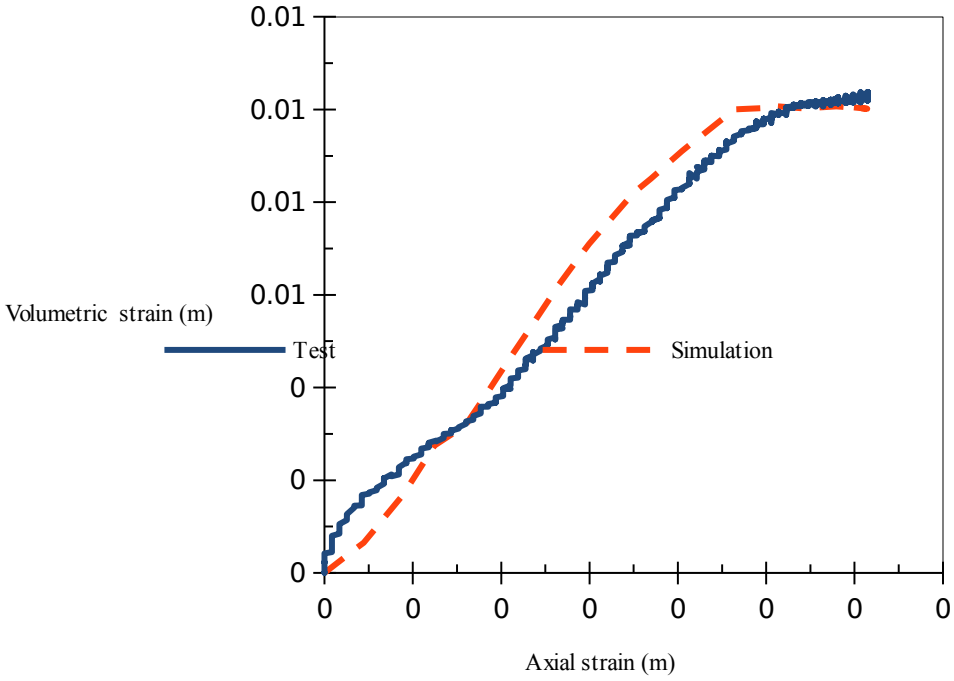


Figure 6.4: Isotropic compression simulation for validation

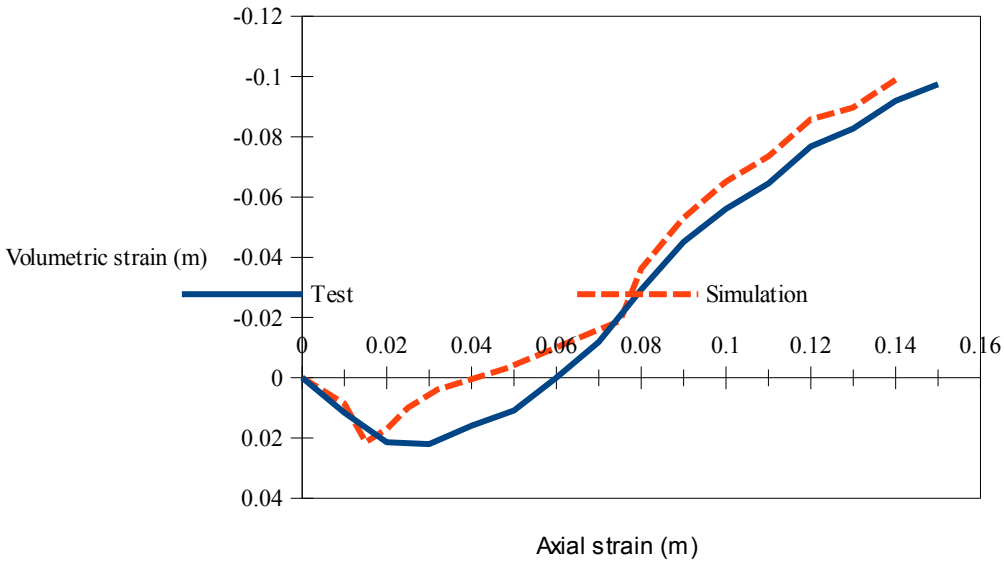


Figure 6.5: Deviatoric compression: volumetric strain vs. axial strain

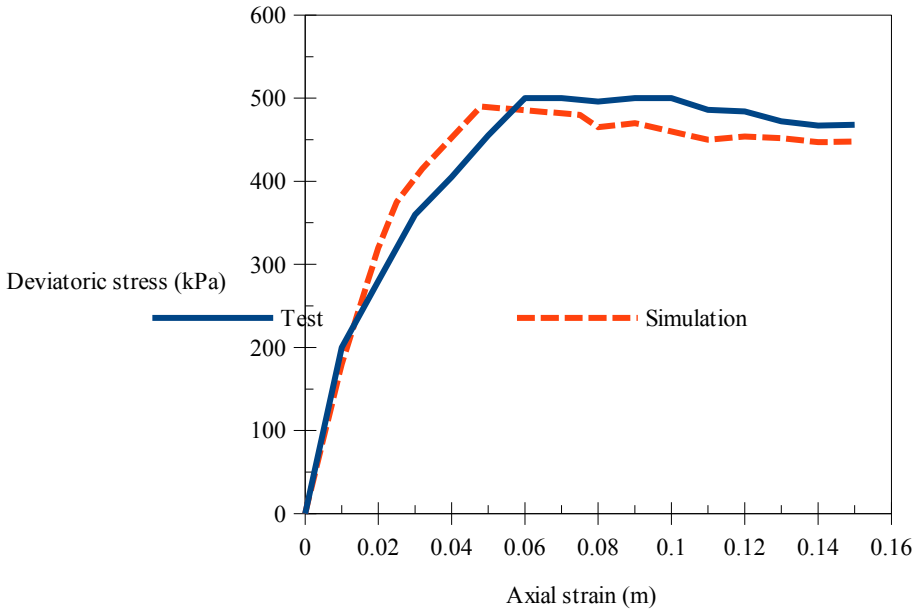


Figure 6.6: Deviatoric compression: deviatoric stress vs. axial strain

6.3 Further analyses

After the validation of the numerical model, further analyses need to be provided. First of all, the effect of geogrid in the triaxial response should be

considered. Figure 6.7 represents the volumetric strain vs. time graph for isotropic loading. The graph contains results for no geogrid case, triangular geogrid, and rectangular geogrid. In isotropic loading results, it can be observed that mainly for two cases with geogrid volumetric strain value is higher than with no geogrid case for a particular time. This indicates that samples with geogrids undergo higher compression faster than the sample with no geogrid. However, the final volumetric strain values are less for the case with geogrids compared to the no geogrid case, which indicates that geogrids help to reduce overall compression of the sample. As for a comparison of two shapes of geogrids, volumetric strain value for triangular shape is less than for rectangularly shaped geogrid for a particular time. Therefore, based on these simulations, triangular shaped geogrid show best performance among the three cases. This may be because of the number of connections of triangular geogrid. Each inner node of the triangular geogrid is connected with six other nodes, whereas the number of connections for inner nodes of rectangular geogrid is four. As it is more fixed it becomes more rigid.

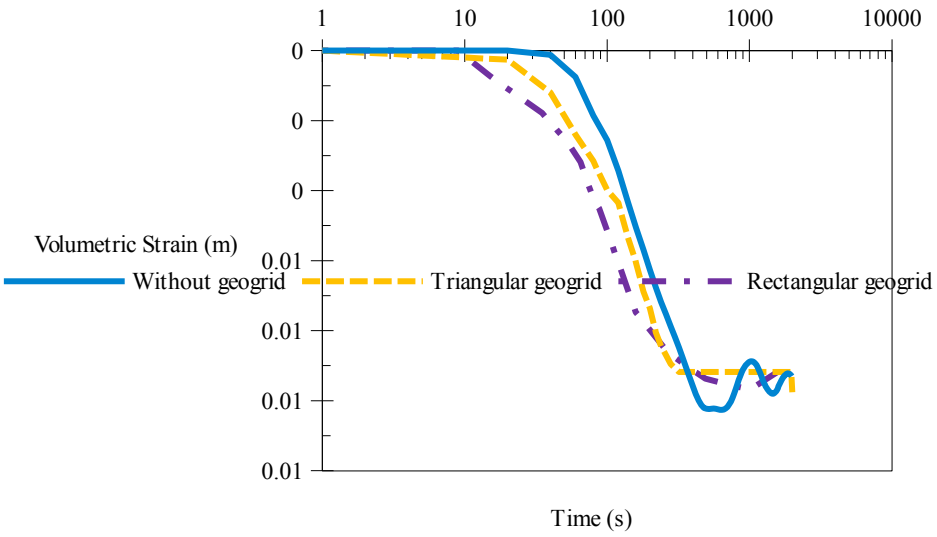


Figure 6.7: Comparison of effects during isotropic compression of rectangular and triangular shaped geogrids with no geogrid case

Figures 6.8 and 6.9 show deviatoric stress vs. axial strain and volumetric strain vs. axial strain graphs, respectively for deviatoric loading. Observing Figure 6.8, deviatoric stress final output is a little higher for cases where geogrid is located. However, it can be concluded that no considerable effect of geogrid to deviatoric stress is noticed. In Figure 6.9, the effect of geogrids to volumetric strain can easily be pointed. In the case of no geogrid, the sample experiences first compression, then it is swelled. However, when geogrids are included in a soil sample, it undergoes only compression. Moreover, volumetric strain value is higher for the case of triangular geogrid.

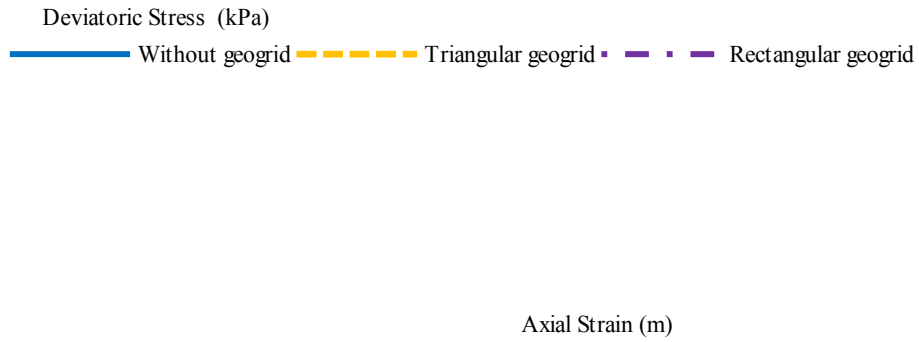


Figure 6.8: Comparison of effects during deviatoric compression of rectangular and triangular shaped geogrids with no geogrid case: deviatoric stress vs. axial strain

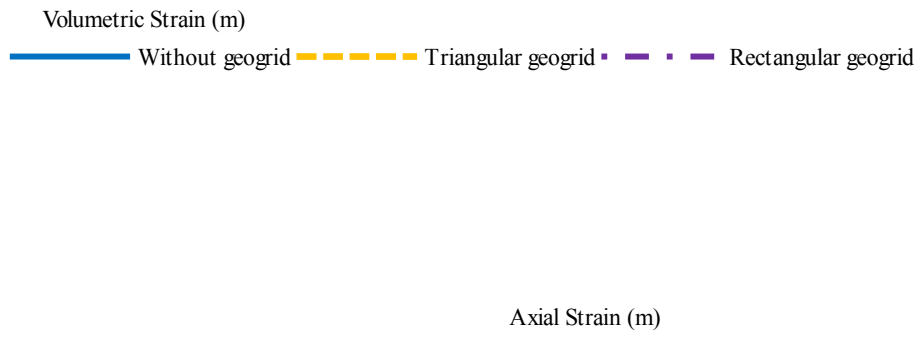


Figure 6.9: Comparison of effects during deviatoric compression of rectangular and triangular shaped geogrids with no geogrid case: volumetric strain vs. axial strain

Furthermore, different cell pressures were applied in order to observe the differences in outputs depending on cell pressure. Figure 6.10 shows the volumetric strain vs. time graph for different confinement values. It can be noted that with increasing confining pressure volumetric strain increases at the initial period. However, as time passes, with higher confining pressure the volumetric strain is smaller. This is due to the dilatancy angle of samples. The value of dilatancy angle is higher for smaller confining pressures. Therefore, soil with highest dilatancy angle results in the highest volumetric strain after a considerable amount of time. Observing Figure 6.11, deviatoric stress value increases with the increasing cell pressure as it was expected.

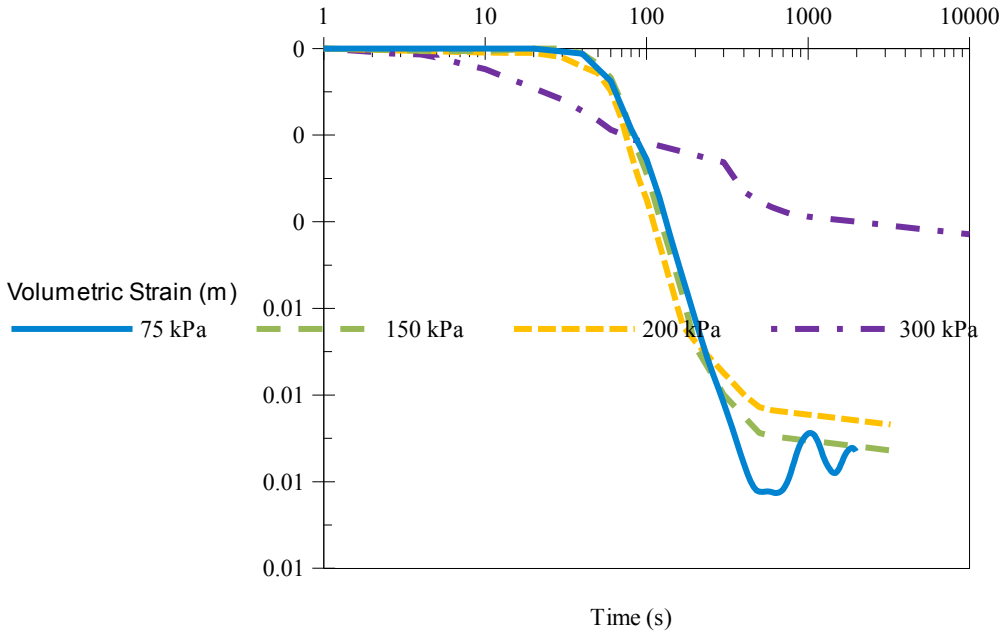


Figure 6.10: Comparison of effects during isotropic compression for different cell pressures

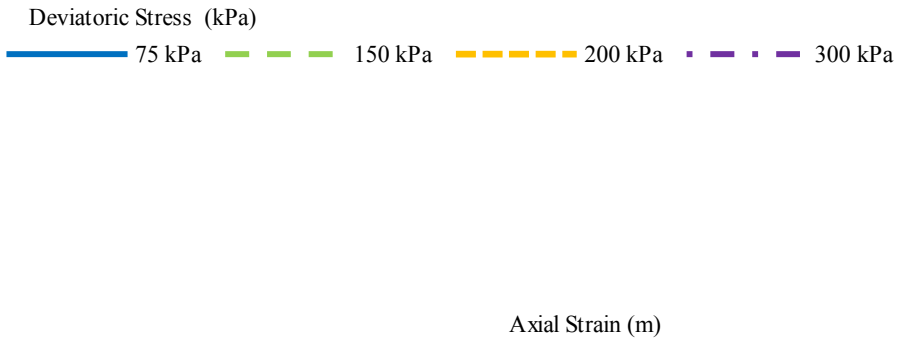


Figure 6.11: Comparison of effects during deviatoric compression for different cell pressures: deviatoric stress vs. axial strain

6.4 Summary

To summarize triaxial test simulations, calibration of macroscopic parameters was performed in order to apply for numerical simulation. Validation of the DEM model was the next step. Both for calibration and validation provided laboratory test results were used. As a result, the proposed model was validated through obtaining similar results as the laboratory tests. Besides, validation of the numerical model indicates that the DEM method and Yade software is effective for performing triaxial tests. Furthermore, the effect of geogrid in soil was analyzed. It was concluded that availability of geogrid improves the behavior of soil. The interlocking qualities of the geogrid is likely providing reinforcement for soil; thus, improving the soil performance. The

results indicate that triangularly shaped geogrid shows a better effect compared to the rectangular geogrid due to its higher number of connections. Moreover, cell pressures were alternated and it was noted that triaxial output is higher for higher confining pressures.

Chapter 7 – Conclusions

7.1 Summary

To conclude, work that is completed by the period and presented in the thesis is the literature review, methodology, preliminary results of the pull-out test and results from triaxial test simulation. The literature review concentrated on geosynthetics and geogrids, in particular, the applications of geogrids, and available techniques that can be applied for evaluation of soil-geogrid interactions. Among these methods, DEM was stated as an approach used for the research and open source software YADE was selected. Furthermore, numerical modeling was explained. The numerical method part included DEM representation of soil and geogrid, contact laws that were applied and other aspects that should be considered. Moreover, Yade codes were written in order to simulate pull-out and triaxial tests. Preliminary results of the pull-out test included parametric study where pull-out was performed under various conditions where confining pressure, size of soil particles and shape of geogrid was altered. As for the triaxial simulations, varied parameters included cell pressure, availability of geogrids (without geogrid or with geogrid) and shape of geogrid. In order to execute these analyses, the first calibration and validation of the numerical model were performed.

7.2 Main Investigations

1. A new code was developed for triangular geogrid shape.

For this study, Yade code for pull-out and the triaxial test was developed from existing codes by adapting those considering specifications of these tests. However, the new code was written for triangularly shaped geogrid.

2. The efficiency of DEM was determined.

There are different methods for analyses of soil-geogrid interaction including field studies, laboratory studies, and numerical methods. Field studies consume a long period of time and conducting these investigations may become highly expensive because a real-size structure is built. For laboratory studies, it also may take a long period of time. Moreover, large-size testing machines are required as geogrids have major apertures and considerable area needs to be tested. Therefore, standard machines are not effective for geogrids testing. Furthermore, ordering specially assembled machine may be expensive. So, numerical methods may be assumed as the most effective due to time and money saving. Moreover, small design changes in the field or laboratory tests require considerable effort, whereas, it is possible to alternate design aspects by a simple procedure in numerical modeling. The importance of this research can be emphasized by the potential that numerical methods provide insight into the mechanics of soil-geogrid systems.

The thesis concentrates on the efficiency of DEM approach on investigations of the behavior of a soil-geogrid interface. In order to do this, the

results obtained by DEM and laboratory tests were compared; and as a result, the numerical simulation results were similar to laboratory test results. Moreover, DEM technique may be effective in terms of time consumption. The time required for the simulation of one test is comparable with the time required for a laboratory test. However, the laboratory method requires a long period of time for the order and construction of appropriately sized apparatus, for its assembly and calibration.

3. The efficiency of geogrid is studied.

This research included DEM analyses of geogrid effect on soil. The analyses included simulating a triaxial test soil sample without geogrid and with geogrid and the results were compared. Based on the preliminary study, availability of geogrid improves soil behavior. However, further studies need to be provided to support this conclusion.

4. The efficiencies of rectangular and triangular shapes of geogrid were compared.

Besides, the research included a comparison between triangular and rectangular shaped geogrids. As a result of numerical simulations, triangular shaped geogrids are more effective than rectangular geogrids. Nevertheless, future studies should be added to prove this conclusion.

7.3 Future Works

Future works are required for this research. For example, other shapes geogrids can be used for further analyses. Thus, the effectiveness of each shape can be evaluated. Moreover, various sizes of geogrids should be analyzed to find either the optimal size of geogrid or optimal aperture to geogrid length ratio. Also, simulations for diverse PSD cases can be added to the research. Thus, the effects of various PSDs on triaxial output can be analyzed. Moreover, DEM code for a pull-out test that was used as a feasibility study could be applied with determined macroscopic parameter values.

Bibliography

[1] Koerner, R. M.: Designing with geosynthetics (fifth edition). Xlibris Corporation, 2005.

[2] Jones, C. J. Earth reinforcement and soil structures. Elsevier, 2013.

[3] Shukla, S. K. (Ed.). (2002). *Geosynthetics and their applications*. Thomas Telford.

[4] Chan, D. H., & Law, K. T. (Eds.). *Soft Soil Engineering: Proceedings of the Fourth International Conference on Soft Soil Engineering, Vancouver, Canada, 4-6 October 2006*. CRC Press.

[5] Giroud, J. P., & Han, J. Design method for geogrid-reinforced unpaved roads. I. Development of Design Method. *Journal of Geotechnical and Geoenvironmental Engineering, 130*(8), 787-797, 2004.

[6] Giroud, J. P., & Noiray, L. Geotextile-reinforced unpaved road design. *Journal of Geotechnical and Geoenvironmental Engineering, 107*(ASCE 16489), 1981.

[7] Zornberg, J. G., & Gupta, R. Geosynthetics in pavements: North American contributions. In Theme Speaker Lecture, *Proceedings of the 9th International Conference on Geosynthetics, Guarujá, Brazil,* May (Vol. 1, pp. 379-400), 2010.

[8] Haas, R., Walls, J., & Carroll, R. G. *Geogrid reinforcement of granular bases in flexible pavements* (No. 1188), 1988.

[9] Chen, X., Zhang, J., & Li, Z. Shear behaviour of a geogrid-reinforced coarse-grained soil based on large-scale triaxial tests. *Geotextiles and Geomembranes, 42(4),* 312-328, 2014.

[10] Yang, G., Ding, J., Zhou, Q., & Zhang, B. Field behavior of a geogrid reinforced soil retaining wall with a wrap-around facing. 2010.

[11] Collin, J. G., & Berg, R. R. Connection strength criteria for mechanically stabilized earth walls. *Transportation research record*, (1414), 1992

[12] Allen, T. M. Issues regarding design and specification of segmental block-faced geosynthetic walls. *Transportation research record*, (1414), 1992

[13] Indraratna, B., Salim, W., & Rujikiatkamjorn, C. *Advanced rail geotechnology–ballasted track*. CRC press, 2011

[14] Mehrjardi, G. T., Ghanbari, A., & Mehdizadeh, H. Experimental study on the behaviour of geogrid-reinforced slopes with respect to aggregate size. *Geotextiles and Geomembranes*, 44(6), 862-871, 2016

[15] NCHRP. NCHRP Project 1-28A, *Harmonized Test Methods for Laboratory Determination of Resilient Modulus for Flexible Pavement Design, Volume 1*. Unbound Granular Material, 198p, 2000.

[16] Perkins, S.W., Christopher, B.R., Cuelho, E.L., Eiksund, G.R., Hoff, I., Schwartz C.W., Svanø, G. and Want, A. Development of design methods for geosynthetic-reinforced flexile pavements. *FHWA-DTFH61-01-X-00068*, Final report, 263p, 2004.

[17] Tutumluer, E. and Dawson, A. Resilient characterization of compacted aggregate. PowerPoint from TRB Workshop on describing aggregate behavior for today's pavements, Washington, DC, 2004

[18] Zornberg, J. G. Advances in the use of geosynthetics in pavement design.

Geosynthetics India'11, 23-24 September 2011, IIT Madras, Chennai PP 1, 21, 2011.

[19] Perkins, S.W. and Edens, M.Q. Finite element and distress models for geosynthetic-reinforced pavements. *International Journal of Pavement Engineering, Vol. 3*, No.4, pp. 239-250, 2002

[20] Qian, Y., Mishra, D., Tutumluer, E., & Kwon, J. Comparative evaluation of different aperture geogrids for ballast reinforcement through triaxial testing and discrete element modeling. In *Proceedings of the 2013 Geosythetics Conference*, 2013.

[21] Cundall, P. A., & Strack, O. D. A discrete numerical model for granular assemblies. *Geotechnique, 29*(1), 47-65, 1979.

[22] O'Sullivan, Catherine. *Particulate discrete element modelling: a geomechanics perspective*. CRC Press, 2014.

[23] Kwon, J., Tutumluer, E., & Konietzky, H. Aggregate base residual stresses affecting geogrid reinforced flexible pavement response. *International Journal of Pavement Engineering, 9*(4), 275-285, 2008.

[24] Ngo, N. T., Indraratna, B., & Rujikiatkamjorn, C. Modelling geogrid-reinforced railway ballast using the discrete element method. *Transportation Geotechnics, 8,* 86-102, 2016.

[25] Sugimoto, M., Alagiyawanna, A.M.N.:Pullout behavior of geogrid by test and numerical analysis. Journal of Geotechnical and Geoenvironmental Engineering ASCE 129 (4), 361–371, 2003.

[26] Khedkar, M.S., Mandal, J.N.:Pullout behaviour of cellular reinforcements. Geotextiles and Geomembranes 27 (4), 262–271, 2009.

[27] McDowell, G. R., Harireche, O., Konietzky, H., Brown, S. F., & Thom, N. H.: Discrete element modelling of geogrid-reinforced aggregates. Proceedings of the Institution of Civil Engineers-Geotechnical Engineering, 159(1), 35-48, 2006.

[28] Tran, V. D. H., Meguid, M. A., &Chouinard, L. E.: A finite–discrete element framework for the 3D modeling of geogrid–soil interaction under pullout loading conditions. GeotextilesandGeomembranes, 37, 1-9, 2013.

[29] Thoeni, K., Effeindzourou, A., Chareyre, B. and Giacomini: A. Discrete Modelling of Soil-Inclusion Problems. In Applied Mechanics and Materials (Vol. 846, pp. 397-402). Trans Tech Publications, 2016.

[30] ŠmilauerV.: Yade Documentation 2nd ed. The Yade Project. DOI 10.5281/zenodo.34073 (http://yade-dem.org/doc/). 2015.

Appendix A

Pull-out test code:

```
# encoding: utf-8

from yade import pack,geom,qt

from yade.gridpfacet import *

from yade import utils

from pylab import *

from yade import plot

from pprint import pprint




rad,gap=.0025,.00125

O.materials.append(CohFrictMat(young=5e6,poisson=0.3,density=2.
65e3,frictionAngle=20,normalCohesion=1e7,shearCohesion=1e7,momentRot
ationLaw=True,label='spheremat'))
```

```
O.materials.append(FrictMat(young=5e6,poisson=0.3,density=2.65e
3,frictionAngle=20,label='sphere'))

O.materials.append(FrictMat(young=5e6,poisson=0.5,frictionAngle
=0,density=0,label='walls'))



O.engines=[

      ForceResetter(),

      InsertionSortCollider([


Bo1_Sphere_Aabb(),Bo1_Box_Aabb(),Bo1_GridConnection_Aabb(),

      ]),

      InteractionLoop(


[Ig2_Sphere_Sphere_ScGeom(),Ig2_Box_Sphere_ScGeom(),Ig2_GridNode_Gri
dNode_GridNodeGeom6D(),Ig2_Sphere_GridConnection_ScGridCoGeom(),Ig2_
GridConnection_GridConnection_GridCoGridCoGeom()],
```

```
[Ip2_FrictMat_FrictMat_FrictPhys(),Ip2_CohFrictMat_CohFrictMat_CohFr
ictPhys(setCohesionNow=True,setCohesionOnNewContacts=False)],



[Law2_ScGeom_FrictPhys_CundallStrack(),Law2_ScGeom6D_CohFrictPhys_Co
hesionMoment(),Law2_ScGridCoGeom_FrictPhys_CundallStrack()]

          ),


GlobalStiffnessTimeStepper(timestepSafetyCoefficient=0.7),


     NewtonIntegrator(gravity=(0,0,0),damping=0.3,label='newton')

     ]



     #### Parameters of a triangular grid ###

     #L=0.3 #length [m

     #l=0.3     #width     [m]
```

```
#nbL=8     #number of nodes for the length [#]

#nbl=8     #number of nodes for the width  [#]

#r=L/1000. #radius

#D=0.001 #Diameter of a hexagon [m]

#t=D/2 #Sides of hexagon [m]

#color=[255./255.,102./255.,0./255.]

#nodesIds=[]

##Create all nodes first :

#list1=[a for a in range(nbl) if a%2==0]

#list2=[b for b in range(nbl) if b%2>0]

#list3=[c for c in range(nbl-1) if c%2==0]

#list4=[d for d in range(nbl-1) if d%2>0]



#for i in range(nbL):

     #for j in list1:
```

```
                    #nodesIds.append(O.bodies.append(gridNode([i*L/
nbL,j*l/nbl*sqrt(3)/2,0.35],r,wire=False,fixed=False,material='spher
emat',color=color)) )#central node

                #nodesIds.append(O.bodies.append(gridNode([i*L/nbL-
t*sin(pi/3),j*l/nbl*sqrt(3)/2-
t/2,0.35],r,wire=False,fixed=False,material='spheremat',color=color)
) )#6 hexagon nodes



#nodesIds.append(  O.bodies.append(gridNode([i*L/nbL,j*l/nbl*sqrt(3)/
2-
t,0.35],r,wire=False,fixed=False,material='spheremat',color=color)))



#nodesIds.append(  O.bodies.append(gridNode([i*L/nbL+t*sin(pi/3),j*l/
nbl*sqrt(3)/2-
t/2,0.35],r,wire=False,fixed=False,material='spheremat',color=color)
))



#nodesIds.append(  O.bodies.append(gridNode([i*L/nbL+t*sin(pi/3),j*l/
nbl*sqrt(3)/2+t/2,0.35],r,wire=False,fixed=False,material='spheremat
',color=color)) )
```

```
#nodesIds.append(  O.bodies.append(gridNode([i*L/nbL,j*l/nbl*sqrt(3)/
2+t,0.35],r,wire=False,fixed=False,material='spheremat',color=color)
))

                #nodesIds.append(  O.bodies.append(gridNode([i*L/nbL-
t*sin(pi/3),j*l/nbl*sqrt(3)/2+t/2,0.35],r,wire=False,fixed=False,mat
erial='spheremat',color=color)))



     #for i in range(nbL-1):

          #for j in list2:

                    #nodesIds.append(O.bodies.append(gridNode([(i+0
.5)*L/nbL,j*l/nbl*sqrt(3)/2,0.35],r,wire=False,fixed=False,material=
'spheremat',color=color)) )

                    #nodesIds.append(O.bodies.append(gridNode([(i+0
.5)*L/nbL-t*sin(pi/3),j*l/nbl*sqrt(3)/2-
t/2,0.35],r,wire=False,fixed=False,material='spheremat',color=color)
) )



#nodesIds.append(  O.bodies.append(gridNode([(i+0.5)*L/nbL,j*l/nbl*sq
```

```
rt(3)/2-
t,0.35],r,wire=False,fixed=False,material='spheremat',color=color)))



#nodesIds.append(  O.bodies.append(gridNode([(i+0.5)*L/nbL+t*sin(pi/3
),j*l/nbl*sqrt(3)/2-
t/2,0.35],r,wire=False,fixed=False,material='spheremat',color=color)
))



#nodesIds.append(  O.bodies.append(gridNode([(i+0.5)*L/nbL+t*sin(pi/3
),j*l/nbl*sqrt(3)/2+t/2,0.35],r,wire=False,fixed=False,material='sph
eremat',color=color)) )



#nodesIds.append(  O.bodies.append(gridNode([(i+0.5)*L/nbL,j*l/nbl*sq
rt(3)/2+t,0.35],r,wire=False,fixed=False,material='spheremat',color=
color)))


     #nodesIds.append(      O.bodies.append(gridNode([(i+0.5)*L/nbL-
t*sin(pi/3),j*l/nbl*sqrt(3)/2+t/2,0.35],r,wire=False,fixed=False,mat
erial='spheremat',color=color)))
```

```
      #for i in range(1):

               #for j in list2:

                        #nodesIds.append(O.bodies.append(gridNode([i*L/
nbL,j*l/nbl*sqrt(3)/2,0.35],r,wire=False,fixed=False,material='spher
emat',color=color)) )



      #for i in range(nbL-1):

               #for j in list3:

                        #nodesIds.append(O.bodies.append(gridNode([(i*L
/nbL+L/nbL/4),j*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4,0.35],r,wire=False,f
ixed=False,material='spheremat',color=color)))



#nodesIds.append(O.bodies.append(gridNode([i*L/nbL+3*L/nbL/4,j*l/nbl
*sqrt(3)/2+l/nbl*sqrt(3)/4,0.35],r,wire=False,fixed=False,material='
spheremat',color=color)))



      #for i in range(nbL-1):
```

```
            #for j in list2:



#nodesIds.append(O.bodies.append(gridNode([(i+0.5)*L/nbL+L/nbL/2,j*l
/nbl*sqrt(3)/2,0.35],r,wire=False,fixed=False,material='spheremat',c
olor=color)))

    #for i in range(nbL-1):

            #for j in list4:



#nodesIds.append(O.bodies.append(gridNode([((i+0.5)*L/nbL+L/nbL/4),j
*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4,0.35],r,wire=False,fixed=False,mate
rial='spheremat',color=color)))

    #for i in range(nbL-2):

            #for j in list4:

                    #nodesIds.append(O.bodies.append(gridNode([(i+0
.5)*L/nbL+3*L/nbL/4,j*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4,0.35],r,wire=F
alse,fixed=False,material='spheremat',color=color)))

    #for i in range(nbL-1):

            #for j in list1:
```

```
#nodesIds.append(O.bodies.append(gridNode([(i*L/nbL+L/nbL/2),j*l/nbl
*sqrt(3)/2,0.35],r,wire=False,fixed=False,material='spheremat',color
=color))) #node between the connections



      #for i in range(1):

               #for j in range(nbL-1):



#nodesIds.append(O.bodies.append(gridNode([i*L/nbL,j*l/nbl*sqrt(3)/2
+l/nbl*sqrt(3)/4,0.35],r,wire=False,fixed=False,material='spheremat'
,color=color)))

      #for i in range(nbL-1,nbL):

               #for j in range(nbL-1):



#nodesIds.append(O.bodies.append(gridNode([i*L/nbL,j*l/nbl*sqrt(3)/2
+l/nbl*sqrt(3)/4,0.35],r,wire=False,fixed=False,material='spheremat'
,color=color)))
```

```
      #for i in range(1):

               #for j in list4:

                        #nodesIds.append(O.bodies.append(gridNode([(i*L
/nbL+L/nbL/4),j*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4,0.35],r,wire=False,f
ixed=False,material='spheremat',color=color)))




      ##Create connection between the nodes



      #for i in range(0,len(nodesIds)):

            #for j in range(i+1,len(nodesIds)):

                                      #dist=(O.bodies[i].state.pos   -
O.bodies[j].state.pos).norm()

                  #if(dist<=t*1.1):

                        #O.bodies.append( gridConnection(i,j,r,color=co
lor))
```

```
	#for i in range(0,len(nodesIds)):

		#for j in range(i+1,len(nodesIds)):

					#dist=(O.bodies[i].state.pos   -
O.bodies[j].state.pos).norm()

				#if(dist>t*20.1):

					#if  (dist<=1.0001*0.5*sqrt(t*t+(L/nbL-
2*t*sin(pi/3))*(L/nbL-2*t*sin(pi/3)))):#horizontal distance

					#O.bodies.append( gridConnection(i,j,r,colo
r=color))



	### Parameters of a rectangular grid ###

	L=0.10 #length [m]

	l=0.10     #width     [m]

	nbL=10#number of nodes for the length [#]

	nbl=10#number of nodes for the width  [#]
```

```
	r=L/100    #radius

	color=[255./255.,102./255.,0./255.]

	nodesIds=[]

	#Create all nodes first :

	for i in range(0,nbL):

		for j in range(0,nbl):



nodesIds.append(  O.bodies.append(gridNode([i*L/nbL+0.005,j*l/nbl+0.0
05,0.05],r,wire=False,fixed=True,material='spheremat',color=color))
)



	#Create connection between the nodes

	for i in range(0,len(nodesIds)):

		for j in range(i+1,len(nodesIds)):

			dist=(O.bodies[i].state.pos                    -
O.bodies[j].state.pos).norm()
```

```
                if(dist<=L/nbL*1.01):


    O.bodies.append( gridConnection(i,j,r,color=color))



    ## create walls around the soil packing

    mn,mx=Vector3(0,0,0),Vector3(0.1,0.1,0.1)  #  corners  of  the
initial packing

    walls=aabbWalls([mn,mx],thickness=0,material='walls')

    wallIds=O.bodies.append(walls)

    O.bodies.append(pack.regularHexa(pack.inParallelepiped((0,0,0),
(0.025,0,0),(0,0.1,0),
(0,0,0.1)),radius=rad,gap=rad/2.0,color=(0.5,0.5,0.1),material='sphe
remat'))

    O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.025,0
,0),(0.05,0,0),(0.025,0.1,0),
(0.025,0,0.1)),radius=rad,gap=rad/2.0,color=(0.5,0.5,0.5),material='
spheremat'))

    O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.05,0,
0),(0.075,0,0),(0.05,0.1,0),
```

```
(0.05,0,0.1)),radius=rad,gap=rad/2.0,color=(0.5,0.5,0.1),material='s
pheremat'))

      O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.075,0
,0),(0.1,0,0),(0.075,0.1,0),
(0.075,0,0.1)),radius=rad,gap=rad/2.0,color=(0.5,0.5,0.5),material='
spheremat'))



      ##Create packing

      ### create walls around the packing

      #mn,mx=Vector3(0,0,0),Vector3(0.05,0.05,0.05) # corners of the
initial packing

      #walls=aabbWalls([mn,mx],thickness=0,material='walls')

      #wallIds=O.bodies.append(walls)

      #O.bodies.append(pack.regularHexa(pack.inParallelepiped((0,0,0)
,(0.0125,0,0),(0,0.05,0),
(0,0,0.05)),radius=rad,gap=rad/3.0,color=(0.5,0.5,0.1),material='sph
eremat'))

      #O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.0125
,0,0),(0.025,0,0),(0.0125,0.05,0),
```

```
(0.0125,0,0.05)),radius=rad,gap=rad/3.0,color=(0.5,0.5,0.5),material
='spheremat'))

     #O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.025,
0,0),(0.0375,0,0),(0.025,0.05,0),
(0.025,0,0.05)),radius=rad,gap=rad/3.0,color=(0.5,0.5,0.1),material=
'spheremat'))

     #O.bodies.append(pack.regularHexa(pack.inParallelepiped((0.0375
,0,0),(0.05,0,0),(0.0375,0.05,0),
(0.0375,0,0.05)),radius=rad,gap=rad/3.0,color=(0.5,0.5,0.5),material
='spheremat'))



     #Set a fixed node

     O.bodies[0].state.blockedDOFs='xyzXYZ'



     #utils.growParticles((2),updateMass=True, dynamicOnly=True)

     area=1e-2

     F1=10000*area

     print "F1:",F1
```

```
totVol = 0.001

density=2.65e3

K=5e7

while True:

#for i in range(100):

        mass=sum([b.state.mass for b in O.bodies])

      porosity=(totVol-mass/density)/totVol

        print "Porosity:",porosity

      stress=O.forces.f(285)[2]/area

      growFactor = 1+((F1-O.forces.f(285)[2])/area/K)

        print "stress:",stress, "growFactor", growFactor



      if abs(1-growFactor)>1e-6:

                  utils.growParticles(growFactor,updateMass=True,
dynamicOnly=True)
```

```
            O.run(20,1)

        else: break



#Apply pull-out force

for i in range (0,101):

            O.bodies[i].state.blockedDOFs='xyzXYZ'

            O.bodies[i].state.vel[0]=-0.06



#Apply confining pressure:

normalStress=300e3

stiff=5e7

cte=normalStress*area

print "current force: ",O.forces.f(285)[2],", target: ",cte

#sum(O.forces.f(id)[2] for id in (285,286))
```

```
	#boundaryVel=copysign((normalStress-O.forces.f(285)
[2]/area)/stiff/O.dt, O.forces.f(285)[2]-cte)

	#O.engines	=	O.engines+[PyRunner(command='boundaryVel;
O.bodies[285].state.vel=(0,0,boundaryVel)',iterPeriod=1)]



	#plot force in grid :

	for i in range (0,101):

		O.bodies[i].state.blockedDOFs='xyzXYZ'

		O.bodies[i].state.vel[0]=-0.06



	O.dt=1e-04

	qtr = qt.Renderer()

	qtr.bgColor=[1,1,1]

	qt.View()



	plot.resetData()
```

```
    for i in range(102000):

        F=(abs(O.forces.f(91)[0])+abs(O.forces.f(100)[0])
+abs(O.forces.f(99)[0])+abs(O.forces.f(98)[0])+abs(O.forces.f(97)
[0])+abs(O.forces.f(96)[0])+abs(O.forces.f(95)[0])
+abs(O.forces.f(94)[0])+abs(O.forces.f(93)[0])+abs(O.forces.f(92)
[0]))

        boundaryVel=copysign((normalStress*area-
abs(O.forces.f(285)[2]))/stiff/O.dt, abs(O.forces.f(285)[2])-cte)

        O.bodies[285].state.vel[2]=boundaryVel

        print "boundaryVel:",boundaryVel

        print "current force: ",O.forces.f(285)[2],", target:
",cte

        print "F:",F

        s1=O.bodies[0]

          ep=((s1.state.pos[0]-s1.state.refPos[0]))

        plot.addData(F=F,ep=ep)
```

```
            plot.saveDataTxt('run     102000     F=300,     rad=0.0025,
gap=2.txt.bz2',vars=('F','ep'))


            O.run(1,1)


      O.saveTmp()
```

Appendix B

Triaxial test code:

```
# -*- coding: utf-8 -*-

from yade import pack

from yade.gridpfacet import*



#  DEFINING VARIABLES AND MATERIALS

nRead=readParamsFromTable(

      num_spheres=15000,# number of spheres

      compFricDegree = 32, # contact friction during the confining phase

      key='_triax_base_', # put you simulation's name here

      unknownOk=True

)
```

```
from yade.params import table



num_spheres=table.num_spheres# number of spheres

key=table.key

compFricDegree = table.compFricDegree # initial contact friction during the confining phase (will be decreased during the REFD compaction process)

finalFricDegree = 32 # contact friction during the deviatoric loading

rate=-0.02 # loading rate (strain rate)

damp=0.2 # damping coefficient

stabilityThreshold=0.01 # we test unbalancedForce against this value in different loops (see below)

young=4.5e6 # contact stiffness

mn,mx=Vector3(0,0,0),Vector3(0.3,0.3,0.6) # corners of the initial packing
```

```
## create materials for spheres and plates

O.materials.append(CohFrictMat(young=young,poisson=0.33,density
=2.65e3,frictionAngle=20,normalCohesion=1e7,shearCohesion=1e7,moment
RotationLaw=True,label='spheremat'))

O.materials.append(FrictMat(young=young,poisson=0.33,frictionAn
gle=radians(compFricDegree),density=1810,label='spheres'))

O.materials.append(FrictMat(young=young,poisson=0.33,frictionAn
gle=0,density=0,label='walls'))

## create walls around the packing

walls=aabbWalls([mn,mx],thickness=0,material='walls')

wallIds=O.bodies.append(walls)

#### Parameters of a triangular grid ###

#L=0.3 #length [m]
```

```
#l=0.3     #width     [m]

#nbL=8     #number of nodes for the length [#]

#nbl=8     #number of nodes for the width  [#]

#r=L/200.  #radius

#D=0.001 #Diameter of a hexagon [m]

#t=D/2 #Sides of hexagon [m]

#color=[255./255.,102./255.,0./255.]

#nodesIds=[]

##Create all nodes first :

#list1=[a for a in range(5,nbl+5) if a%2==0]

#list2=[b for b in range(5,nbl+5) if b%2>0]

#list3=[c for c in range(5,nbl+4) if c%2==0]

#list4=[d for d in range(5,nbl+4) if d%2>0]




#for i in range(5,5+nbL):
```

```
            #for j in list1:

                        #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color)) )#central node

                #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL-t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)) )#6 hexagon nodes

                #nodesIds.append(      O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-5)*l/nbl*sqrt(3)/2-
t+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=colo
r)))

                #nodesIds.append(      O.bodies.append(gridNode([(i-
5)*L/nbL+t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)))

                #nodesIds.append(      O.bodies.append(gridNode([(i-
5)*L/nbL+t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)) )
```

```
                #nodesIds.append(       O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+t+0.015,0.3],r,wire=False,fixed=True,material='sp
heremat',color=color)))

                #nodesIds.append(       O.bodies.append(gridNode([(i-
5)*L/nbL-t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)))



     #for i in range(5,4+nbL):

          #for j in list2:

                        #nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color)) )

                        #nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL-t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)) )

                #nodesIds.append(       O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-5)*l/nbl*sqrt(3)/2-
```

```
t+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=colo
r)))

                #nodesIds.append(       O.bodies.append(gridNode([(i-
4.5)*L/nbL+t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)))

                #nodesIds.append(       O.bodies.append(gridNode([(i-
4.5)*L/nbL+t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)) )

                #nodesIds.append(       O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+t+0.015,0.3],r,wire=False,fixed=True,material='sp
heremat',color=color)))

                #nodesIds.append(       O.bodies.append(gridNode([(i-
4.5)*L/nbL-t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)))



     #for i in range(5,6):

             #for j in list2:
```

```
                        #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color)) )



      #for i in range(5,4+nbL):

               #for j in list3:

                        #nodesIds.append(O.bodies.append(gridNode([((i-
5)*L/nbL+L/nbL/4+0.015),(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))

                  #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL+3*L/nbL/4+0.015,(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))



      #for i in range(5,4+nbL):

               #for j in list2:
```

```
                #nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL+L/nbL/2+0.015,(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color)))

     #for i in range(5,4+nbL):

             #for j in list4:

                #nodesIds.append(O.bodies.append(gridNode([((i-
4.5)*L/nbL+L/nbL/4+0.015),(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))

     #for i in range(5,3+nbL):

             #for j in list4:

                      #nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL+3*L/nbL/4+0.015,(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))

     #for i in range(5,4+nbL):

             #for j in list1:
```

```
                #nodesIds.append(O.bodies.append(gridNode([((i-
5)*L/nbL+L/nbL/2+0.015),(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color))) #node between the connections



    #for i in range(5,6):

            #for j in range(5,4+nbL):

                #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))

    #for i in range(4+nbL,5+nbL):

            #for j in range(5,4+nbL):

                #nodesIds.append(O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))



    #for i in range(5,6):
```

```
            #for j in list4:

                    #nodesIds.append(O.bodies.append(gridNode([((i-
5)*L/nbL+L/nbL/4+0.015),(j-
5)*l/nbl*sqrt(3)/2+l/nbl*sqrt(3)/4+0.015,0.3],r,wire=False,fixed=Tru
e,material='spheremat',color=color)))

     ##for i in range(3+nbL,4+nbL):

            ##for j in list1:

                    ##nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+0.015,0.3],r,wire=False,fixed=True,material='sphe
remat',color=color)) )

                    ##nodesIds.append(O.bodies.append(gridNode([(i-
4.5)*L/nbL-t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)) )

              ##nodesIds.append(     O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-5)*l/nbl*sqrt(3)/2-
t+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=colo
r)))

              ##nodesIds.append(     O.bodies.append(gridNode([(i-
4.5)*L/nbL+t*sin(pi/3)+0.015,(j-5)*l/nbl*sqrt(3)/2-
```

```
t/2+0.015,0.3],r,wire=False,fixed=True,material='spheremat',color=co
lor)))

                ##nodesIds.append(      O.bodies.append(gridNode([(i-
4.5)*L/nbL+t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)) )

                ##nodesIds.append(      O.bodies.append(gridNode([(i-
4.5)*L/nbL+0.015,(j-
5)*l/nbl*sqrt(3)/2+t+0.015,0.3],r,wire=False,fixed=True,material='sp
heremat',color=color)))

                ##nodesIds.append(      O.bodies.append(gridNode([(i-
4.5)*L/nbL-t*sin(pi/3)+0.015,(j-
5)*l/nbl*sqrt(3)/2+t/2+0.015,0.3],r,wire=False,fixed=True,material='
spheremat',color=color)))




    ##Create connection between the nodes

    #m=0.5*sqrt(t*t+(L/nbL-2*t*sin(pi/3))*(L/nbL-2*t*sin(pi/3)))
```

```
      #for i in range(5,5+len(nodesIds)):

            #for j in range(i+1,6+len(nodesIds)):

                                        #dist=(O.bodies[i].state.pos   -
O.bodies[j].state.pos).norm()

                  #if(dist<=t*1.1):

                        #O.bodies.append( gridConnection(i,j,r,color=co
lor))



      #for i in range(5,5+len(nodesIds)):

            #for j in range(i+1,6+len(nodesIds)):

                                        #dist=(O.bodies[i].state.pos   -
O.bodies[j].state.pos).norm()

                  #if(dist>t*20.1):

                        #if (dist<=1.0001*m):#horizontal distance

                              #O.bodies.append( gridConnection(i,j,r,colo
r=color))
```

```
#### Parameters of a rectangular grid ###

#L=0.30 #length [m]

#l=0.30    #width     [m]

#nbL=10#number of nodes for the length     [#]

#nbl=10#number of nodes for the width [#]

#r=L/100   #radius

#color=[255./255.,102./255.,0./255.]

#nodesIds=[]

##Create all nodes first :

#for i in range(5,5+nbL):
```

```
            #for j in range(5,5+nbl):

                #nodesIds.append(      O.bodies.append(gridNode([(i-
5)*L/nbL+0.015,(j-
5)*l/nbl+0.015,0.4],r,wire=False,fixed=True,material='spheremat',col
or=color)) )



      ##Create connection between the nodes

      #for i in range(5,5+len(nodesIds)):

            #for j in range(i+1,6+len(nodesIds)):

                #dist=(O.bodies[i].state.pos                          -
O.bodies[j].state.pos).norm()

                #if(dist<=L/nbL*1.01):



      #O.bodies.append( gridConnection(i,j,r,color=color))



      ## use a SpherePack object to generate a random loose particles
packing
```

```
psdSizes,psdCumm=[0.000074,0.00015,0.0004,0.002,0.004,0.007,0.0
1,0.02,0.025],[0.,0.02,0.05,0.12,0.21,0.36,0.45,0.81,1.0]
#[0.000074,0.00015,0.0004,0.002,0.004,0.007,0.01,0.02,0.025],
[0.,0.,0.,0.,0.,0.,0.,0.8,1.0]

sp=pack.SpherePack();




#DEFINING ENGINES



triax=TriaxialStressController(

        ##  TriaxialStressController  will  be  used  to  control
stress and strain. It controls particles size and plates positions.

        maxMultiplier=1.+2e4/young,  #  spheres  growing  factor
(fast growth)

        finalMaxMultiplier=1.+2e3/young, # spheres growing factor
(slow growth)

        thickness = 0,

        stressMask = 7,
```

```
            internalCompaction=False,   #   If   true   the   confining
pressure is generated by growing particles

      )



      newton=NewtonIntegrator(damping=damp)



      O.engines=[

            ForceResetter(),


InsertionSortCollider([Bo1_Sphere_Aabb(),Bo1_Box_Aabb(),Bo1_GridConn
ection_Aabb()]),

            InteractionLoop(


[Ig2_Sphere_Sphere_ScGeom(),Ig2_Box_Sphere_ScGeom(),Ig2_GridNode_Gri
dNode_GridNodeGeom6D(),Ig2_Sphere_GridConnection_ScGridCoGeom(),Ig2_
GridConnection_GridConnection_GridCoGridCoGeom()],
```

```
[Ip2_FrictMat_FrictMat_FrictPhys(),Ip2_CohFrictMat_CohFrictMat_CohFr
ictPhys(setCohesionNow=True,setCohesionOnNewContacts=True)],



[Law2_ScGeom_FrictPhys_CundallStrack(),Law2_ScGeom6D_CohFrictPhys_Co
hesionMoment(),Law2_ScGridCoGeom_FrictPhys_CundallStrack(),Law2_Grid
CoGridCoGeom_FrictPhys_CundallStrack()]

                ),


GlobalStiffnessTimeStepper(active=1,timeStepUpdateInterval=100,times
tepSafetyCoefficient=0.8),

           triax,


TriaxialStateRecorder(iterPeriod=100,file='WallStresses'+table.key),

           newton

     ]



     #  APPLYING CONFINING PRESSURE
```

```
##the value of (isotropic) confining stress defines the target stress to be applied in all three directions

triax.goal1=triax.goal2=triax.goal3=-75000

while 1:

  O.run(1000, True)

    #the global unbalanced force on dynamic bodies, thus excluding boundaries, which are not at equilibrium

  unb=unbalancedForce()

      print 'unbalanced force:',unb,' mean stress: ',triax.meanStress

        if unb<stabilityThreshold and abs(-75000-triax.meanStress)/75000<0.001:

    break



O.save('confinedState'+key+'.yade.gz')

print "### Isotropic state saved ###"
```

```
# DEVIATORIC LOADING


#We move to deviatoric loading, let us turn internal compaction
off to keep particles sizes constant

if O.iter>=1500:

    triax.internalCompaction=False

    #setContactFriction(radians(finalFricDegree))

    triax.stressMask = 5

    triax.stressMask = 5

    triax.goal2=rate

    triax.goal1=-75000

    triax.goal3=-75000
```

```
O.saveTmp()




# a function saving variables

def history():

        plot.addData(e11=-triax.strain[0],  e22=-triax.strain[1],
e33=-triax.strain[2],

                                ev=-triax.strain[0]-triax.strain[1]-
triax.strain[2],#volumetric strain

        s11=-triax.stress(triax.wall_right_id)[0],

        s22=-triax.stress(triax.wall_top_id)[1],

        s33=-triax.stress(triax.wall_front_id)[2],

        i=O.iter)

      plot.saveDataTxt('strainvtime.txt.bz2',vars=('ev','e11','e22
','e33','s11','s22','s33'))
```

```
                                        O.engines=O.engines[0:5]+
[PyRunner(iterPeriod=1,command='history()',label='recorder')]
+O.engines[5:7]

          O.engines[4]=PyRunner(iterPeriod=1,command='history()',labe
l='recorder')



      O.dt=1

      O.run(10000,True)



      O.saveTmp()
```