



**LIMITS ON THE VALIDITY OF INFINITE WIDTH AND LENGTH ASSUMPTIONS
FOR MODELLING SHALLOW LANDSLIDES**

by

Carlos Tapia Cabrera

Student Number 180367894.

Supervised by

Dr D. G. Milledge.

Submitted in partial fulfilment of the requirements for the degree of Master of Science in
Geotechnical Engineering in the Faculty of Science, Agriculture and Engineering.

Civil Engineering and Geospatial

School of Engineering

Second Floor, Cassie Building

Newcastle University

Newcastle upon Tyne

NE1 7RU

United Kingdom

Tel: +44 (0) 191 208 6445

Email: soe.pgt.civil@newcastle.ac.uk

AUGUST 2020

NEWCASTLE UPON TYNE, UNITED KINGDOM

© I hereby certify that this work is my own, except where otherwise acknowledged, and that it has not been previously submitted for a degree at this, or any other university. The copyright of this Dissertation rests with the author, any information derived from it should be acknowledge.

A handwritten signature in black ink, appearing to read 'Carlos Tapia Cabrera', with a horizontal line drawn through the middle of the signature.

Carlos Tapia Cabrera
SN 180367894

ACKNOWLEDGEMENT

On the very outset of this project, I would like to extend my strongly and sincere gratitude to the personages who succoured and walked besides me in this adventure. Without their active guidance, cooperation, and engagement, I would not have achieved the end of the project.

I am incredibly indebted to my project supervisor **Dr D Milledge** for his endless guidance and encouragement to accomplish this assignment.

I am thankful as well to **Dr M Rouainia** for his help in the Numerical Modelling and advice to enhance the process.

I have to pay gratitude to *The School of Engineering* for giving me the opportunity to embrace the project, and from the bottom of my hearth to *The National Scholarship and Educational Loans Program – PRONABEC (PERU)* for the scholarship to study at *Newcastle University*.

I also acknowledge with a deep sense of reverence, my gratitude towards my **Mother** and members of my family, who has always supported me morally as well as economically, this gratitude is also extended to my friends who directly or indirectly helped me to keep strong while miles away from home.

Any omission in this brief acknowledgment does not mean lack of gratitude.

**Thanking you all,
Carlos Tapia.**

COVID-19 Impact Review

Student Name : Carlos Tapia Cabrera

Student Number : 180367894

Supervisor's Name : Dr David Milledge

Current Project Title:

“LIMITS ON THE VALIDITY OF INFINITE WIDTH AND LENGTH ASSUMPTIONS FOR MODELLING SHALLOW LANDSLIDES”.

New Project Title if applicable:


Not required.

Has COVID-19 impacted your work?

Yes, it has.

Please provide details of how COVID-19 has impacted your project in discussion with your supervisor:

This project is primarily numerical analysis, so access to laboratory equipment is not a concern. However, the project does require numerical analysis using Plaxis 3D, a finite element software package for which the School holds several licenses. These licenses cannot be installed locally and access to School of Engineering buildings is limited due to COVID19. We plan to mitigate this impact by using university computers via remote access. Carlos has tested the remote access to Plaxis 3D and is able to gain access to the software. The other impacts of COVID19 are related to access to expertise, particularly in-person supervisory support. We are mitigating this impact through online supervisory meetings.

Student Signature:  _____

Supervisor Signature: David Milledge

Date: 03 – 06 – 2020

ABSTRACT

Mechanistic modelling of landslides at any scale larger than an individual hillslope rely almost exclusively on the infinite slope stability model. The model's central assumption that landslides are infinitely long (down slope) and wide (across slope) is usually considered valid since most natural slides are shallow and planar. However, this is rarely justified, because the critical length/depth (L/H) and width/depth (W/H) ratios below which edge effects become important are poorly constrained. We identify the critical L/H and W/H ratios by benchmarking infinite slope stability predictions against finite element predictions for a set of synthetic two- and three-dimensional slopes. In each case we assume that the difference between the predictions is due to error in the infinite slope method. We use a 15-noded triangular finite element mesh in PLAXIS 2D to examine the length effects in the slope stability of ~3000 2D models across the range of geometric and geotechnical conditions typically found on natural slopes. Furthermore, to examine the width effects in the slope stability a 10-noded tetrahedral finite element mesh in PLAXIS 3D was used, in this case ~1000 3D models across the range of geometric and geotechnical conditions were simulated. We find that: L/H ratios of >14 lead to <10% error in FoS and of >50 lead to <5% error in FoS for 95% of parameter combinations, consistent with previous studies. Critical W/H ratios are shorter, with ratios of 10 and 20 leading to <10% and <5% error, respectively. To put these results in context we compare critical L/H and W/H ratios with measured ratios at ~300 shallow landslides from six separate inventories across a range of landscapes. Although critical W/H is narrower than critical L/H for any given error threshold, more landslides fail the width than length criteria because most landslides are narrower than they are long.

Keywords: Width, Length, convergence, Finite Elements, Infinite Slope.

TABLE OF CONTENTS

ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES.....	viii
LIST OF TABLES.....	xii
CHAPTER 1.- INTRODUCTION	1
1.1. Aim of the Project.....	4
1.2. Objectives of the Project.....	4
1.3. Scope of the Project	4
CHAPTER 2.- LITERATURE REVIEW	6
2.1. Stability Factor of Safety for Shallow Landslides	7
2.1.1. Infinite Slope Method (IS).....	8
2.1.2. Finite Element Methods (FE)	13
2.2. Geotechnical Variability for Soil Properties.....	21
2.2.1. Monte Carlo Simulation	22
2.3. PLAXIS Remote Scripting interface Via Python	23
CHAPTER 3.- METHODOLOGY	25
3.1. Infinite Slope Method for Slope Stability	25
3.2. Finite Element Method for Slope Stability	26
3.3. Data Generation by Random Analysis.....	28
CHAPTER 4.- RESULTS	32
4.1. Phase 1, Exploratory investigation	32
4.1.1. Slope translational slide simulation.....	32
4.1.2. FoS normalization to the FE FoS	33
4.1.3. Meshing influence in the FoS FE normalization.....	36
4.1.4. Observations in the dilatancy angle.....	39
4.1.5. 3D settings for the models	40
4.2. Phase 2, Monte Carlo Simulation	43
CHAPTER 5.- DISCUSSION.....	48
CHAPTER 6.- CONCLUSIONS AND RECOMMENDATIONS	54
6.1. Conclusions.....	54

6.2. Recommendations for Future Works	57
REFERENCES	58
APPENDIX – A/ MODEL SCRIPTS	61
a. Code for Plaxis 2D Models/ Includes Monte Carlo Simulation runs	61
b. Code for Plaxis 3D Models/ Includes Monte Carlo Simulation runs	61

LIST OF FIGURES

Figure 1.- Natural landslides, size, and effects.....	1
Figure 2.- Shallow landslide, Cajamarca, Peru – 2019.....	6
Figure 3.- Infinite slope failure in dry sand, vertical depth [H] uniform across the slope layer, [β] is the slope angle, [b] is the length of the slope and [W] is the weight of the analysed slice of slope.	9
Figure 4.- Pore water pressure head.	10
Figure 5.- Infinite slope failure $c - \phi$ soil with parallel seepage, the soil properties are expressed in terms of effective parameters.....	11
Figure 6.- Finite element mesh view of a simple geometry (generated by PLAXIS 2D 2018)	13
Figure 7.- Distribution of nodes and stress points in interface elements and their connection to soil elements (PLAXIS 2D Manuals, 2018).....	14
Figure 8.- 3D Soil element / 10 – noded tetrahedrons (PLAXIS 2D Manuals, 2017).....	14
Figure 9.- Soil Constitutive models, H is the vertical depth pf the soil layer allowed to slide, L is the horizontal length of the proposed slope and beta (β) is the inclination angle of the slope.	15
Figure 10.- Command anatomy for PLAXIS and Python equivalent in PLAXIS (PLAXIS 2D Reference Manual, 2018)	23
Figure 11.- PLAXIS 2D settings for the numerical modelling.....	26
Figure 12.- PLAXIS 3D settings for the numerical modelling.....	27
Figure 13.- Activation of the remote server in PLAXIS Software.	29
Figure 14.- Flowchart for 2D modelling in PLAXIS through Python scripting.....	30
Figure 15.- Flowchart for 3D modelling in PLAXIS through Python scripting.....	31
Figure 16.- Shallow landslide simulated in PLAXIS 2D for translational failure mechanism, $L/H = 8$	33
Figure 17.- FoS for different L/H ratios predicted from IS and FE methods, termed here a convergence curve, in this case for a 6-noded mesh with the parameters shown in Table 5. ...	34

Figure 18.- FoS for different L/H ratios predicted from IS and FE methods, termed here a convergence curve, in this case for a 15-noded mesh with the parameters shown in Table 5.	34
Figure 19.- Normalization of the curves in Figure 17 & Figure 18 to the L/H ratio of 128....	35
Figure 20.- Very coarse mesh in PLAXIS 2D model, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e).....	36
Figure 21.- Very fine mesh in PLAXIS 2D model, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e).....	37
Figure 22.- Normalised ratio to the FE FoS at different mesh sizes.....	38
Figure 23.- Combined Mesh in PLAXIS 2D model, very fine for the Mohr – Coulomb layer and very coarse for the Linear Elastic layer, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e)	38
Figure 24.- Combined Mesh in PLAXIS 2D model, a very fine mesh for the top layer, and a very coarse, with coarseness factor of 4, for the bottom layer. In the plot, the very fine mesh curve is not visible because it is always below the combined mesh curve.	39
Figure 25.- Dilatancy angle variation in the Slope FE Model, both of the plotted models use the geometrical and geotechnical parameter from Table 4.....	39
Figure 26.- PLAXIS 3D Slope model settings and expected displacements from the analysis. For the model, it has been used the values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = 64$	41
Figure 27.- Cross section of the 3D model in Figure 26, although the displacements are not notorious, it is appreciated that the displacements follow the same behaviour as in Figure 16.	41
Figure 28.- Slope FE FoS and IS FoS for a 10-noded mesh in PLAXIS 3D, the model uses geometrical and geotechnical values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = 64$. convergence curve for 3D models	42
Figure 29.- Normalized curve to the W/H ratios of 64.....	42

Figure 30.- Variability of W/H convergence curves at different L/H ratios, the model uses geometrical and geotechnical values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = [2, 4, 8, 16, 32 \text{ \& } 64]$	43
Figure 31.- Random data generated by the Monte Carlo Simulation for the infinite Length Analysis (~3000 2D models), all curves are normalized to the L/H ratio of 128.	44
Figure 32.- Normalized curves for the 50 th , 90 th , 95 th percentile of the random data for the infinite length analysis.....	44
Figure 33.- Normalized curve to the FE FoS for L/H ratios at the 95 th Percentile of the random data generated by the Monte Carlo Simulation. This NRL curve is crossing the 10% threshold at L/H ratio of 14 and the 5% threshold at the L/H ratio of 50.....	45
Figure 34.- Random data generated by the Monte Carlo Simulation for the infinite Width Analysis (~1000 3D models), all curves belong to the L/H ratio of 64 and are normalized to the W/H ratio of 64.....	46
Figure 35.- Normalized curves for the 50 th , 90 th , 95 th percentile of the random data for the infinite width analysis.....	46
Figure 36.- Normalized curve to the FE FoS for W/H ratios at the 95 th Percentile of the random data generated by the Monte Carlo Simulation. This NRL curve is crossing the 10% threshold at W/H ratio of 10 and the 5% threshold at the W/H ratio of 20.....	47
Figure 37.- Slope failure after a suddenly – heavy rain, Cajamarca, Peru/2019.....	48
Figure 38.- Landslide size recurrence in terms of L/H & W/H ratios, these data is a collection from the following authors: [1] The Appalachian Mountains (Montgomery, et al., 2000); [2] Hakoishi, Japan (Paudel, et al., 2003); [3] San Gabriel Mountains, California (Rice, et al., 1969); [4] Santa Barbara County, California (Gabet & Dunne, 2002); [5] Cumbria, England (Warburton, et al., 2008); and [6] Oregon Coast Range (Montgomery, 1991).....	49
Figure 39.- Normalized curves to the FE FoS, crossing the 5% threshold at W/H ratio of 20, L/H ratio of 50 and 10% threshold at W/H ratio of 10, L/H ratio of 14.....	50
Figure 40.- Scatter plots showing the random selection of the IS parameters at which the IS predictions converge to the FE predictions within 5% and 10% threshold (L/H_{crit}), the involved geotechnical and geometrical parameters are: effective soil cohesion [kN/m^2], effective soil unit weight [kN/m^3], normalized free surface height [-], effective friction angle [$^\circ$], slope angle [$^\circ$] & soil depth [m].	51

Figure 41.- Scatter plots showing the random selection of the IS parameters at which the IS predictions converge to the FE predictions within 5% and 10% threshold (W/H_{crit}), the involved geotechnical and geometrical parameters are: effective soil cohesion [kN/m^2], effective soil unit weight [kN/m^3], normalized free surface height [-], effective friction angle [$^\circ$] & slope angle [$^\circ$] 52

LIST OF TABLES

Table 1.- Soil parameter for the Infinite Slope Equation.....	22
Table 2.- Limits of the model geotechnical parameter for the IS and FE Methods.....	25
Table 3.- PLAXIS Modelling properties	27
Table 4.- Finite Element Method, trial model.....	34
Table 5.- Size elements influence in the FE FoS	37

CHAPTER 1.- INTRODUCTION

Shallow landslides are extremely frequent in nature, they can be regarded as the main source of landforms on the Earth's surface, e.g. mountains, hills, plateaus; nevertheless, it is also possible to appreciate them in minor landforms included buttes, canyons, valleys and basins. Although shallow landslides are mostly caused by natural disturbance of a slopes, they can be triggered by heavy rains or follow draughts, volcanic activity, or earthquakes.

The importance of study shallow landslides is mostly associated with the hazard they represent to the population or human – made structures within the vicinity, they can cause destruction on roadways, railways, dams, or at bigger scales they can cause catastrophic events burying houses or even villages (Figure 1). To minimize the impact of shallow landslides, warning agencies usually have Hazard Maps with the distribution of areas where landslides have occurred before, they include the location of steep slopes that have been altered for the construction of roads or buildings, in addition, these could mention areas where wildfire or human modification of land have destroyed vegetation and now they are susceptible to slide (Montgomery, *et al.*, 2000).



Figure 1.- Natural landslides, size, and effects.¹

¹ This picture can be found at http://www.weatherwizkids.com/?page_id=1326

The information in hazard maps are usually gathered by surveying the areas of interest, however, when those areas become greatly extensive, the budget for surveying becomes extremely expensive as well, therefore, instead of surveying, the hazard maps are done by satellite imagery, which work with Digital Terrain Models to predict shallow landslides by solving equation (Simplified Newmark, SHALSTAB). The methodology behind those models is frequently based on the Infinite Slope Method (IS) as its central component (Montgomery, 1991), the central component, in a hazard model, determines which areas are at risk of sliding and which are safe for human settlements or engineering structures.

When addressing the hazard of a landslide, the size is always important, not only because of the amount of mass waste in movement, but also because of the events it can trigger afterwards, sometimes an small mass movement is a sign of instability, therefore, a warning sign for evacuation before something bigger happens. The models for shallow landslide predictions based on the Infinite Slope Method have been used for many years, although they have been useful and quite chip to implement, they can incur in model bugs, e.g. determining very long slides rather than wide or avoiding the determination of small scale landslides, which could trigger bigger ones. These misperceptions rise warmings in the formulation of those models, calling back the question, how wide and long a landslide should be, to be considered as infinite? and hence analysed by the Infinite Slope Method. This question was addressed first in 2012 by Dr Milledge, who determined that Length/Depth (L/H) ratios should be greater than 25 for a slope to be considered as infinite, however, for the Width/Depth (W/H) ratios the analysis has not been done yet.

Using a hazard model to predict shallow landslides, which uses the IS Method as its central component, is always worth, they are quite chip to implement and do not require very big deals of computational memory to analyse large amounts of data in short periods of time. Nevertheless, nowadays the computational analysis through Finite Elements (FE) is one of the

most accurate calculation methods for almost all engineering problems, thus the slope stability analysis; the only limitations of the FE Methods are mostly related to the computation cost and tightly restricted by the limitation of data that it can handle in certain intervals of time, e.g. to process a model with the IS as its central component it could take just hours rather than days that could take for the FE software to process the same model. In despite of the FE drawbacks, it is possible to benchmark the IS Method (central component) by the FE Method and tell the degree of accuracy of the Infinite Slope Method at non infinite slope dimensions, thus from the models containing the IS method as its central component, e.g. the Hazard Models for predicting shallow landslides or Catchment landslide models.

The current research project will address once more the question, how wide and long a landslide should be, to be considered as infinite?, although the L/H analysis was done in 2012, this time it will be addressed by a different Numerical Software (PLAXIS), and mesh element distribution (triangular/2D and tetrahedral/3D), so as to able to make comparisons and stablish relationships between the length and width analysis.

The project is distributed in six chapters, Chapter 1, will carry out an introduction into the problem, giving a brief understanding about the importance of the analysis by stablishing the aim, objectives, and scope of the research. Chapter 2, will go through the wide range of literature about the problem, making clear that the Infinite Slope Method is well recognised and accepted in most of the analysis for shallow and translational landslides.

Moving into Chapter 3, it will stablish the proposed methodology for addressing the investigation step by step, in this chapter, it is also stablished the software characteristics aimed to the investigation. Chapter 4 will present the results of the analysis at different stages of the investigation as the proposed methodology. In Chapter 5 will be a discussion comparing the findings with the ones in the literature review, although it is not expected huge differences in the L/H analysis with the one done in 2012 by Dr Milledge, it is worth to notice that this time

it is being used a different Numerical Software (PLAXIS 2D & 3D), mesh systems (Triangular/tetrahedral instead of rectangular), and the Numerical Approach to the random data. Finally, in Chapter 6, it is presented the conclusions of the investigation in accordance with the aim and objectives, in the same chapter, it is also presented some recommendation for future works in this field.

1.1. Aim of the Project

The aim of the project is to determine the limits on the validity of infinite width and length assumptions for modelling shallow landslides.

1.2. Objectives of the Project

In order to achieve this, the objectives are to:

- Accurately establish a failure processes as close as possible to the Infinite Slope (IS) Model with the exception of the finite dimensions by examining 2D & 3D Finite Element Models in PLAXIS Software.
- Build a 2D (for the length analysis), and a 3D (for the width analysis) Finite Element Model in PLAXIS Software capable to determine The Stability Factor of Safety (FoS) of a slope under the variation in all its soil and geometrical parameters.
- Examine the sensitivity of the findings to a wider range of conditions by developing a system or methodology which randomly chooses soil and geometrical parameters for the models in 2D & 3D.
- Analyse the outputs from the 2D & 3D models in relationship with the existent data related to shallow landslides analysis.

1.3. Scope of the Project

The aim of the project is to develop a system that could tell us how wide and long a landslide should be, to be considered as infinite and hence analysed by the Infinite Slope Method. However, to do so, by Finite Element Analysis we need other parameters, those

called strength values of the model, for example, Young's modulus and Poisson's ratio, for those, previous investigations have demonstrated little or negligible influence in the Factor of Safety, therefore, this time the research will not go into a deeper analysis of the values and will be taken as constant along the analysis. Furthermore, when analysing the 3D models for determine the infinite width in a slope, the Length/Depth ratio will be consider constant at certain ratios, e.g. L/H of 64, these ratios will be tightly linked to the 2D analysis, where firstly it will be determined a proper ratio since which the slope dimensions can be considered infinite in length.

The importance of the findings of the current research is attached to the development of Hazard, Geomorphic and Landscape Evolution Models, which usually solve the Infinite Slope Equation as its central component of the calculations, however, this research project is an attempt to stablish conditions under which a central component (IS) is suitable, rather than an attempt to develop such models for an specific location.

CHAPTER 2.- LITERATURE REVIEW

Landslides also called landslips are a common geomorphic process characterized by land forming most of the natural scenarios on Earth. Furthermore, landslides are considered as hazards for population or human – made structures located at the toe of steep slopes or close to them. Landslides can be triggered by rainfall, snowmelt, changes in water level, stream erosion, changes in ground water, earthquakes, volcanic activity, disturbance by human activities, or any combination of these factors.

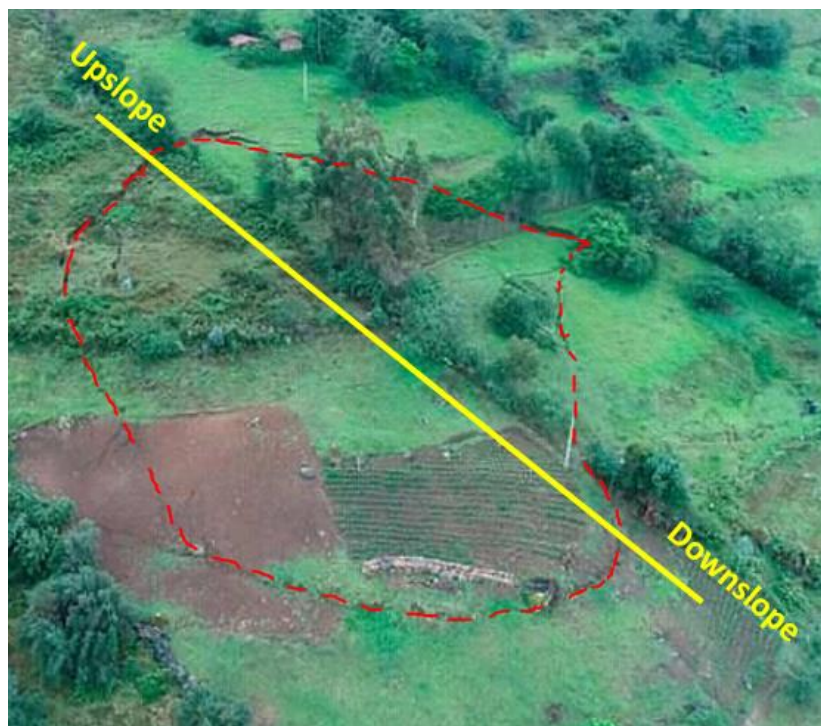


Figure 2.- Shallow landslide, Cajamarca, Peru – 2019.

Although most of the landslides have multiple causes, slope movement occurs when forces acting down – slope (mainly due to gravity) exceed the strength of the earth materials that compose the slope (Figure 2), landslides can severely affect roadways, railways, bridges, and similar, leaving people uncommunicated for several days. Despite the fact that landslides can be of any dimension, it is natural shallow landslides the most recurrent events, in most of the scenarios this type of landslides present a translational behaviour (Milledge, et al., 2014). As the shallow landslides happen due to forces acting downslope and the resistance forces

produced by the material that compose the slope, it can be calculated ratios between those forces that would tell us about the stability of a slope. There are many methods which have been developed through the years for computing the stability of a slope, however, this research project will focus on two of them only, the Infinite Slope Method (IS) and the Finite Element Method (FE), both of them measure the slope stability factor of safety (FoS) as the ratio of the shear strength of the soil to the shear stress required for the equilibrium.

2.1. Stability Factor of Safety for Shallow Landslides

The slope Stability Factor of Safety (FoS) for shallow landslides has been widely studied across the years, however, many assumptions have been made, e.g. steady ground water behaviour, uniform soil properties across the slope, infinite dimensions across the length & width of the slope, other similar. According to “Abramson, *et all.*, (2002, p.331) the main items required to evaluate the stability of a slope are: [1] Shear strength, [2] Slope geometry, [3] Pore pressure or Seepage forces and [4] Environmental conditions.”

The items listed in the last paragraph lead to a purely equilibrium analysis in a slope, however, natural slopes are mostly saturated, where the drainage plays an important role in the calculation of the FoS. Although the undrained failure is more likely to occur in a manmade slope (Ladd, 1991), the effective stress state in drained conditions governs most of the failure scenarios in nature. What is more, if the pore pressure parameters are available the theoretical undrained FoS may be determined from the more readily computed drained FoS values (Moore, 1970).

From the great amount of available methods to analyse the FoS, shallow landslides are analysed by equilibrium methods making its length and width infinite, thus the name of the method “Infinite Slope Method (IS)”. Although in recent years, the analysis is moving to a Numerical Solutions by computational programs, the Infinite Slope Method is still widely used and accepted (Abramson, *et al.*, 2002).

2.1.1. Infinite Slope Method (IS)

The Infinite Slope Method (IS) is one of the many Limit Equilibrium Method available to analyse the Slope Stability Factor of Safety (FoS), the central assumption in the calculation is based on an Infinite Slope, which can be defined as those ones which extend its size to relatively long distances and have a consistent subsoil profile. The failure mechanism is typically parallel to the surface and can be produced by either to a sudden or gradual loss of strength by the soil or to a change in geometry conditions, for example, the steepening of an existing slope (Abramson, *et al.*, 2002). The IS analysis can be contemplated in two scenarios, e.g. [a] for dry sand, where the slope presents high ranges of permeability and the shear resistance depends only of the shear angle, and [b] for cohesive soils where the drainage is typically slow and needs to be accounted in the calculation, here the FoS depends on the effective soil parameters.

The drainage condition in a slope are usually determined by the phreatic surface present in the soil layer, this surface can be delineated, in the field, by using open standpipes as monitoring wells, ending up with a continuous line representing the ground water table levels. A uniform phreatic surface is typically represented as a line in 2 dimensions (linear variations), then as the water have similar flow properties in most of the slope points, it can be assumed that the flow is steady and the head pressure increases linearly with depth across the soil layer.

a) Infinite slope in dry sand

Figure 3 shows a typical slice in dry sand, the forces acting on it are mainly due to gravity, the polygon in the bottom right corner shows the static equilibrium applied to solve those forces. The weight of the slice of soil is defined by the Eq. 1, and it follows the gravity direction as indicated in Figure 3.

$$W = \gamma_s b H (1) \qquad \text{Eq. 1}$$

The parameter γ_s in equation 1 is the Dry Sand Unit Weight in kN/m^3 and the number (1) represents a unit dimension into the paper. The normal, N , and driving, T , forces are determined as follows,

$$N = W \cos \beta \quad \text{and} \quad T = W \sin \beta \quad \text{Eq. 2}$$

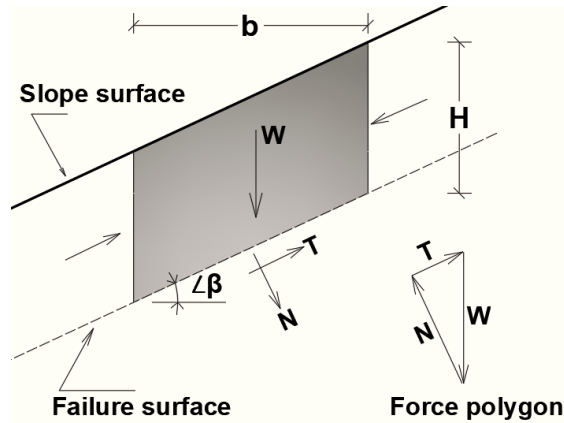


Figure 3.- Infinite slope failure in dry sand, vertical depth $[H]$ uniform across the slope layer, $[\beta]$ is the slope angle, $[b]$ is the length of the slope and $[W]$ is the weight of the analysed slice of slope.

Then the resistance that the soil offers along the failure plane will depend on the frictional angle of the soil and is given by

$$S = N \tan \phi \quad \text{Eq. 3}$$

The FoS is usually consider as the ratio between the available strength of soils in the failure plane to the strength required to maintain stability, therefore, we can write the following expression

$$FoS = \frac{N \tan \phi}{W \sin \beta} = \frac{W \cos \beta \tan \phi}{W \sin \beta} = \frac{\tan \phi}{\tan \beta} \quad \text{Eq. 4}$$

Notice that in the equation 4, the FoS is dependent only of the frictional angle (ϕ) and the slope angle (β), neglecting the importance of the slice high (H), which in reality is one of the causes to increase the weight of the slice, thus the instability. In the same equation also is clearly appreciated that unstable conditions ($FoS < 1$) are met when the slope angle (β) is greater than the soil frictional angle (ϕ).

b) Infinite slope in $c - \phi$ soil with seepage

To have a full understanding of the slope stability in shallow landslides the analysis must include cohesion (cohesive soils) and seepage (phreatic surface), allowing the surface failure to behave under saturated conditions, thus leading the analysis in terms of effective stress parameters.

The phreatic surface is usually represented as a line in two dimensions, therefore, the pore water pressures in a phreatic surface is calculated for steady – state seepage conditions (Figure 4), this concept is based in the assumption that all equipotential lines are straight and perpendicular to the phreatic surface passing through a similar soil – slice element presented in Figure 3.

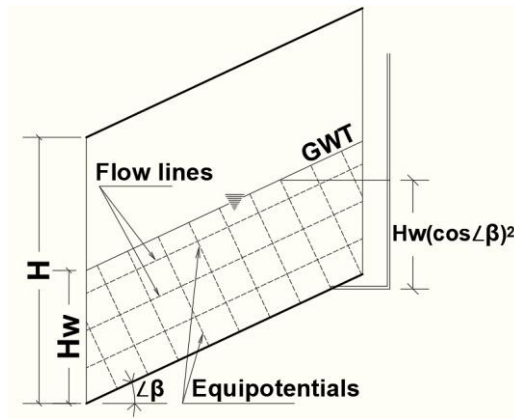


Figure 4.- Pore water pressure head.

Assuming that the phreatic surface is located at “ H_w ” distance from the bottom of the slice “ H ” in high, the pore pressure from the phreatic surface is given by (Figure 4),

$$\mu = \gamma_w H_w \cos^2 \beta \quad \text{Eq. 5}$$

Where “ β ” is the slope angle and “ γ_w ” is the water unit weight, it is also possible to introduce a new variable call “ m ”, which will be the ratio between the height of the phreatic surface and the height of the soil – slice element, this ratio will be used to make simplifications later on.

$$m = \frac{H_w}{H} \quad \text{Eq. 6}$$

With all the elements in place, it is possible to use the same Limit Equilibrium Method as for slopes in dry sand [a)], then determine the FoS under saturated conditions. Figure 5 shows the equilibrium state for this scenario, notice that all the elements are analysed in terms of effective parameters. The available friction strength in failure surface is dependent on the effective cohesion angle (c'), the effective frictional angle (ϕ'), and the effective normal (N') defines the effective stress (σ') perpendicular to the failure surface, therefore, the available resistance strength is given by

$$S = c' + \sigma' \tan \phi' \quad \text{Eq. 7}$$

Where,

- c' : Soil effective cohesion at the failure surface
- σ' : Soil effective stress at the failure surface (N')
- ϕ' : Soil effective frictional angle

The total pore pressure (u) generated along the base of the analysed soil slice will be defined by the Eq. 5 times the base length of the slice, then from Figure 5, and Eq. 2 the effective stress (σ') is given by,

$$\sigma' = N' = N - u = W \cos \beta - \gamma_w H_w \cos \beta^2 \left(\frac{b}{\cos \beta} \right)$$

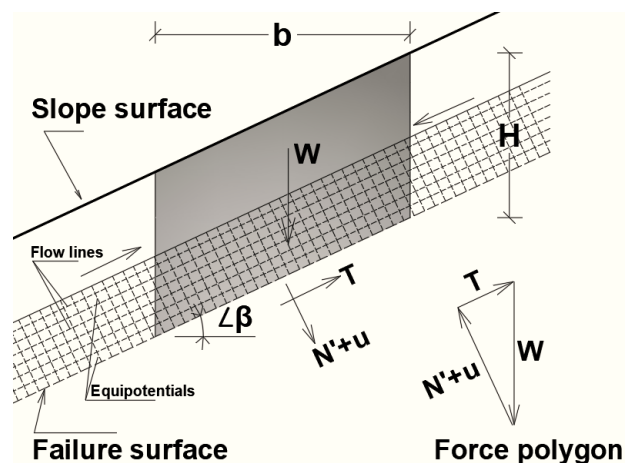


Figure 5.- Infinite slope failure $c - \phi$ soil with parallel seepage, the soil properties are expressed in terms of effective parameters.

The weight (W) of the soil slice is defined by the Eq. 1 with $\gamma_s = \gamma_{sat}$ for the saturated scenario, thus,

$$\sigma' = \gamma_{sat} b H \cos\beta - \gamma_w H_w \cos\beta^2 \left(\frac{b}{\cos\beta} \right) = b \cos\beta (\gamma_{sat} H - \gamma_w H_w) \quad \text{Eq. 8}$$

The available effective cohesion along the base of the analysed soil slice is defined by the effective cohesion times the base length of the slice [b].

Substituting the Eq. 8 in Eq. 7 and considering all the available cohesion as defined lines above,

$$S = c' \left(\frac{b}{\cos\beta} \right) + b \cos\beta (\gamma_{sat} H - \gamma_w H_w) \tan\phi'$$

Then the FoS is defined as the ratio between the available strength of soils in the failure plane to the strength required to maintain stability, therefore,

$$FoS = \frac{S}{T} = \frac{b(c' + \cos\beta^2 (\gamma_{sat} H - \gamma_w H_w) \tan\phi')}{W \sin\beta \cos\beta}$$

Where \mathbf{W} is the slice weight defined by Eq. 1 and replacing Eq. 6 for the ratio between the height of the phreatic surface and the height of the soil – slice element, leads to,

$$FoS = \frac{c' + H \cos\beta^2 (\gamma_{sat} - m \gamma_w) \tan\phi'}{H \gamma_{sat} \sin\beta \cos\beta} \quad \text{Eq. 9}$$

The Eq. 9 gives a wider understanding about the slope behaviour without disregarding any of the soil or geometric parameters, in addition, this equation allow us to work under saturated conditions at different levels for $\mathbf{m} > \mathbf{0}$, and under dry conditions for $\mathbf{m} = \mathbf{0}$, notice that the maximum value that \mathbf{m} can reach is $\mathbf{1}$ and indicates a fully saturated state.

2.1.2. Finite Element Methods (FE)

The finite element method is one of the most widely used methods among the many numerical methods available for computing the stress – strains in geotechnical engineering problems (Abramson, et al., 2002). This method performs its calculations based in the discretization of the whole domain into sub domains also called finite elements, these elements are usually connected to each other through nodes where forces and stresses are calculated in accordance with the Constitutive Model used for the calculations. The discretization of the domain is called a “Mesh” and the meshing system will determine the performance and accuracy of the calculations and solutions expected for the geotechnical problem.

An example of the Mesh elements is presented in Figure 6 for a simple slope geometry generated by PLAXIS 2D 2018 at medium size distribution.

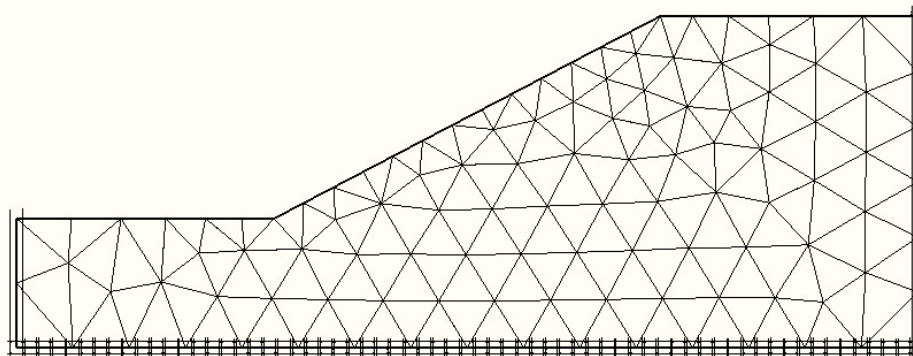


Figure 6.- Finite element mesh view of a simple geometry (generated by PLAXIS 2D 2018)

Normally each element is composed by 15 – noded (cubic - strain), plane – strain triangular elements with 12 Gauss quadrature points as shown in Figure 7 [b], however, PLAXIS 2D – 2018 also allows to work with a 6 – noded mesh and 3 Gauss quadrature points [a] for problems which require less accuracy and better calculations performance. For geotechnical problems which require solutions in three dimensions PLAXIS 3D performs its calculations based on a mesh discretization constituted by 10 – noded

tetrahedral elements with 6 Gauss quadrature points as shown in Figure 8, this is the only available option in PLAXIS 3D.

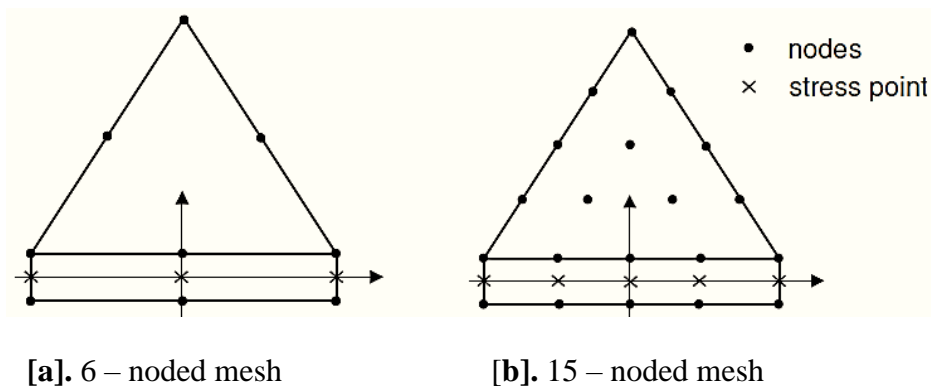


Figure 7.- Distribution of nodes and stress points in interface elements and their connection to soil elements (PLAXIS 2D Manuals, 2018)

The size of the mesh elements play a very important role in the calculations and must be chosen carefully, if they are too small the time needed to perform the calculations will be very large with little or no improvement in the results after certain mesh refinement, on the other hand, if the mesh is too coarse (very large elements) the results might not be accurate enough. Therefore, the size of the mesh goes according to the required performance and the accuracy of the Numerical Calculations, typically the analysis can begin with a very coarse mesh then by making refinements in the areas of interest the final mesh can be considered accurate enough for the facing problem (Brinkgreve, *et al.*, 2017).

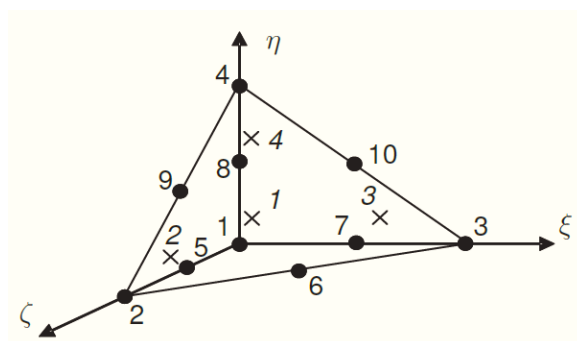


Figure 8.- 3D Soil element / 10 – noded tetrahedrons (PLAXIS 2D Manuals, 2017)

To obtain the Safety Factor through numerical analysis in PLAXIS, the “Strength Reduction Method” is commonly used, this method reduces simultaneously the strength of each soil parameter by a small increment until failure occurs, then, the resulting “Strength Reduction Factor, SRF” is comparable with the Safety Factor (Dawson, et al., 1999), furthermore, PLAXIS also uses the Phi – c Procedure (described later in this chapter), for computing the Safety Factor along with the “Strength reduction Method”.

The Geotechnical Soil Modelling accuracy can be acquired by choosing the right Constitutive Model and Failure Criteria, it also can be refined by selecting the best fit for the discretization size elements. In the present research it will be used a Linear Elastic Model to represent the bedrock and the Elastic – Perfectly – Plastic / Mohr – Coulomb Model to represent the layer allowed to slide across the slope, notice that this configuration has been chosen regarding the preference of PLAXIS Software for fixing its own axis along X – horizontal and Y – vertical rather than inclined axis along the slope. The combination of layers with different model behaviours is used to simulate shallow landslides, where the failure plane is usually the bedrock and parallel to surface allowed to slide (**Figure 9**).

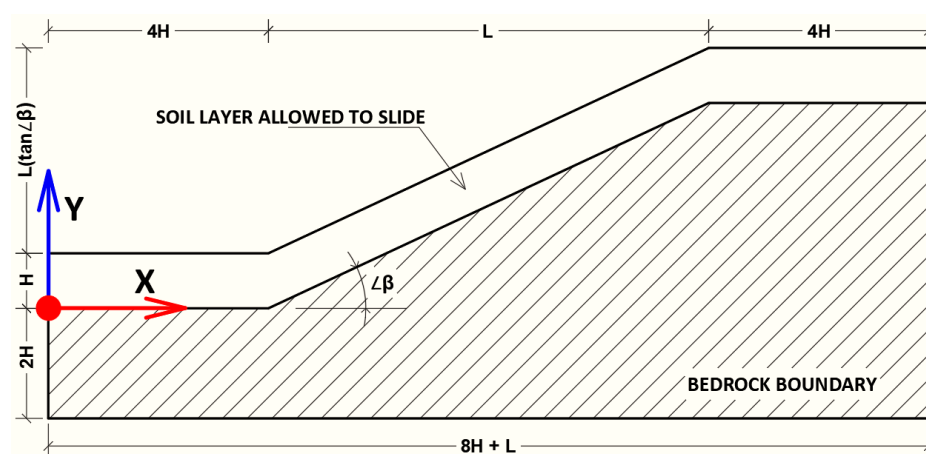


Figure 9.- Soil Constitutive models, H is the vertical depth of the soil layer allowed to slide, L is the horizontal length of the proposed slope and beta (β) is the inclination angle of the slope.

a) Linear Elastic Model

The Linear Elastic Model is based on Hooke's Law of Isotropic Elasticity, it involves two basic elastic parameters, i.e. Young's Modulus E and Poisson's ratio ν . Although the Linear Elastic Model is not suitable to model soils, it may be used to model stiff volumes in the soil, like concrete walls, or intact rock formations (Brinkgreve, et al., 2018).

The Linear Elastic Model is the simplest available stress – strain relationship in PLAXIS, it is generally considered too crude to capture essential deformations or features of soils or rock behaviours, however, it is the best representation for very stiff layers where deformations are not allowed but the interaction with the above layers is important. Shallow landslides generally represent soil translational behaviours, therefore, a bedrock layer which do not allow deformations but rather resist them is the closest approximation to translational landslides thus the Infinite Slope Method.

b) Elastic – Perfectly – Plastic / Mohr – Coulomb Model (MC)

Although non – linearity is a common behaviour in soils (stress – strain curves), the Mohr – Coulomb Model is a first approximation to the real behaviour of soils in most geotechnical problems. The MC Model is based in the principle of elastoplasticity that all strains and strain rates are decomposed into an elastic part and a plastic part. Commonly the elastic strain rates are described or related to the Hooke's Law and the plastic strain rates are proportional to the derivate of the yield function with respect to the stresses (Hill, 1950). This means that plastic strain rates can be represented as vectors perpendicular to the yield surface, however, for the MC Model in PLAXIS this approach generates excessive dilatancy, to control this, the model uses a plastic potential additional to the yield function, denoted in this case as non – associated plasticity contrary to the classic theory of plasticity.

To generate the MC model, requires five soil parameters, which are usually obtained from basic tests on soil samples, the parameters and their standard units are listed below (Brinkgreve, et al., 2018),

E	: Young's modulus	[kN/m ²]
v	: Poisson's ratio	[-]
c	: Cohesion	[kN/m ²]
φ	: Friction angle	[°]
ψ	: Dilatancy angle	[°]
σ _t	: Tension cut – off strength	[kN/m ²] #not used in safety calculation.

Notice that instead of using the Young's Modulus as the stiffness parameter, alternative stiffness parameters can be entered, these parameters are also listed below,

G	: Shear modulus	[kN/m ²]
E _{oed}	: Oedometer modulus	[kN/m ²]

The Shear modulus can be related to the Young's modulus through the following equation, E is the young's modulus in [kN/m²], and v is the Poisson's ratio [-]:

$$\mathbf{G} = \frac{E}{2(1 + v)} \quad \mathbf{Eq. 10}$$

In a numerical analysis with the MC model to duplicate the limit – equilibrium factor of safety results (at least for unreinforced slopes), with the Shear Strength Reduction Method, it is only necessary to: (i) Use the same Young's modulus value for the materials in a multiple – material model, (ii) Assume a single valid Poisson's ratio for the materials, (iii) Assume dilatation angle equal to zero, and, (iv) Use the Elastic – Perfectly – Plastic assumption for a post – peak behaviour (Hammanh, et al., 2005).

From the last paragraph it is directly inferable that the Young's modulus and Poisson's ratio have negligible influence in the factor of safety when analysing a slope with the Shear Strength Method, however, the choice of values for this strength parameters will influence the performance of the calculations. For example, choosing

high Young's modulus and Poisson's ratio will lead to highly un-deformable models, which will consume considerable computational memory for numerical modelling, on the other hand, very low values of Young's modulus would not represent accurately the geotechnical problem.

Another important parameter in the numerical analysis is the dilation (dilatancy) angle (ψ), which controls the amount of plastic volumetric strain developed during plastic shearing and is assumed to be constant during plastic yielding (Brinkgreve, et al., 2018). The choice of the dilatancy angle (ψ) is directly related to the frictional angle of the soil, usually the clay soils tend to show little dilatancy ($\psi \approx 0$), the dilatancy of sands in the contrary depends on both the density and the frictional angle. In general, the dilatancy angle is much smaller than the frictional angle, as a thumb rule, we can say that $\psi \approx \phi - 30^\circ$, however, Hammanh, *et al.*, (2005), said that "the angle of dilation does not have significant impact in slope problems due to generally low confinement environment", therefore, the dilation can be approximate to zero for most of the Numerical Analysis in slope problems, confirming in this way with great assertion the findings of Dawson, *et al.*, 1999.

c) **Meshing settings and influence in the models**

For a single slope, the meshing frame was presented in Figure 6 together with the size distribution and the available options in PLAXIS 2D and 3D, furthermore, it was described that the meshing choice will be reflected in accuracy of the results and the performance of the calculation.

To generate the finite element, PLAXIS software offers global and local setting, in the global configurations the mesh generator requires a global meshing parameter that represents the Target Size of the Elements (I_e), this parameter is usually calculated from the outer geometry dimensions of the model (X_{\min} , X_{\max} , Y_{\min} , Y_{\max} , Z_{\min} , Z_{\max}). The

Target Element dimension or average element size is based on a parameter called Relative Element Size Factor (r_e), this parameter goes from 0.50 to 2.00, as listed below (Brinkgreve, et al., 2018).

Very coarse	: $r_e = 2.00$	30 – 70 elements
Coarse	: $r_e = 1.33$	50 – 200 elements
Medium	: $r_e = 1.00$	90 – 350 elements
Fine	: $r_e = 0.67$	250 – 700 elements
Very fine	: $r_e = 0.50$	500 – 1250 elements

The presented number of elements is an approximation, because the real number of elements within the model will strongly depend on the geometry and optional refinement settings. Notice that with a 15-noded mesh we can have a finer distribution of the nodes, thus more accuracy in the results than a similar mesh composed by an equal number of 6-noded elements, contrary, the use of the 15-noded elements in more time consuming than using a 6-noded elements. The local refinement is mostly referred to those areas of the model which the analysis is aimed, e.g., by applying a refinement factor of 0.50 in certain layer the element size is reduced by half, therefore the accuracy of the displacements in the noded contained in the selected area. The opposite happens when selecting values greater than 1.00, in those areas where the nodal displacement is not required as the outputs in the calculations. The available values for refinement and coarseness in PLAXIS range from 0.03125 to 8.00.

d) Drainage conditions

The introduction of pore pressure in the Numerical Model is aimed to represent the effective soil response, i.e. the interaction of stresses and strains associated with the soil skeleton, in addition, the infinite slope method [pag. 8] introduces pore pressure in the model as well (Eq. 9), through the parameter “ m ” defined by the Eq. 6.

Two main types of drainage conditions are discussed in this section, they are available in PLAXIS software in the main settings of the material properties.

➤ **Non – porous behaviour**

Along this type of drainage neither initial nor excess pore pressure is generated in the soil model, some application can be found when modelling stiff bodies into the soil, e.g. concrete or structural behaviours. Non – porous behaviour is often used in combination with the Linear Elastic Model (LE) to simulate intact rocks or bedrocks beneath the interested soil layers (Brinkgreve, et al., 2018).

➤ **Drained behaviour**

The drained behaviour is used to represent the case of dry soils and also for full drainage due to permeability conditions, this option is also used to simulate long – term soil behaviours without the need to model the precise history of undrained loading and consolidation (Brinkgreve, et al., 2017).

The phreatic calculations related to this type of drainage are similar to the expressed for the seepage conditions in the Infinite Slope Method [pag. 8], which basically represent uniform phreatic conditions across the slope with steady – state flows in the soil layers.

e) **Gravity loads calculations**

This is a special procedure implemented in PLAXIS software for the numerical calculation of the initial stresses based mainly on the volumetric weight of the soil. When selecting this time of calculations for the MC Model, the initial stresses generated strongly depend on the Poisson's ratio selected and the ratio of the horizontal effective stress over the vertical effective stress is the constant known as the Coefficient of Earth at Rest, K_0 . This coefficient could be assumed as a first check – up that the Poisson ratio selected is giving realistic K_0 during the calculations (Brinkgreve, et al., 2018).

The selection of this type of calculation is suitable for soil layers which are non - horizontal, and it is required a complete stress field equilibrium for the computation of factor of safeties through Shear Strength reduction Methods.

f) Safety Calculations (Phi/C reduction)

This procedure is also known as Shear Strength Reduction (SSR), traditionally, the Factor of Safety is defined as the ratio of the actual shear strength to the minimum shear strength required to prevent failure (Bishop, 1955). The safety calculations through this procedure is the reduction of the soil shear strength until collapse occurs, the resulting factor of safety is the ratio of the soil's actual sheer strength to the reduced shear strength at failure (Dawson, et al., 1999).

In PLAXIS software this procedure is implemented and called “Phi – C reduction”, from this, it is inferable that the shear factors “tan(phi)” and “c” are reduced until failure occur, notice that it is also reduced the tensile strength and the dilatancy angle (psi), this last parameter is not reduced at the beginning if set a value lower that the shear angle (phi), then when the reduction of “Phi” reaches the value of “Psi”, it is also reduced until failure occurs.

2.2. Geotechnical Variability for Soil Properties

The Infinite Slope equation for the factor of safety (Eq. 9) calls for six geotechnical and geometrical parameters, expressed in Table 1. The soil is well known to vary its properties not only vertically but also horizontally making each site unique and susceptible to any possible failure mechanism, although geotechnical engineers would normally carry out site investigations for determining those properties, the study gets reduced to a very specific sites, or a predefined project.

Table 1.- Soil parameter for the Infinite Slope Equation

Parameter	Symbol
Friction angle	ϕ (<i>Phi</i>)
Cohesion	c
Soil depth	H
Normalised free surface height	m
Soil unit weight	γ
Slope angle	β

To wider the analysis, it is possible to set boundaries where each of the soil parameters would normally vary, for example, the slope angle, it can vary from 0° to 90°, nevertheless, slope angles of 90° and shallow landslides is a combination that would rarely happen in nature, what is more, when running the Finite Element Analysis this type of models might collapse under gravity loads due to the unstable geometry or the presence of inaccurate values. Once set the extreme limits of each of the geotechnical parameters, the question comes to the values between those limits, to solve this problem, a probabilistic analysis with random values selection can give a first understanding of the slope behaviour under different geometrical and geotechnical conditions. There are many tools available to randomly select values from a predefined range, the analysis of the selected values can be treated with a Monte Carlo Analysis for normal distribution.

2.2.1. Monte Carlo Simulation

This method is based on randomized values selection, each of the geotechnical parameters follow the same process and give a non – repetitive unique value along the number of simulations. The process is repeated many, many times to obtain an approximate result of the FoS convergence to the IS, the component random variable for each calculation is set independent so as to have more reliable selections. The Monte Carlo simulations follow a five – step process:

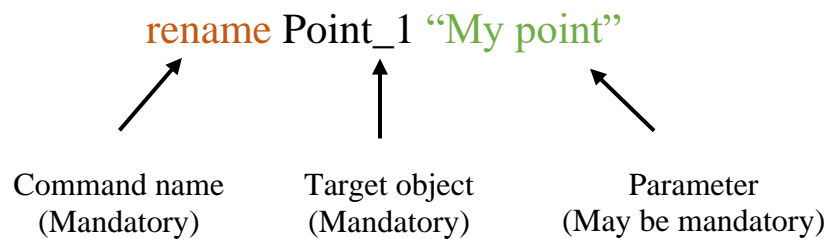
1. For each component, a random value is selected along each geotechnical and geometrical parameter.

2. Calculate the FoS for the IS method using the adopted values from the step 1.
3. Calculate the FoS for the FE method using the adopted values from the step 1 for each of the geometrical ratios selected.
4. Calculate the convergence (NR) from of the FE to the IS FoS for each geometrical ratio selected.
5. Repeat the steps 1 to 4 many times, storing the NR for each trial calculation.

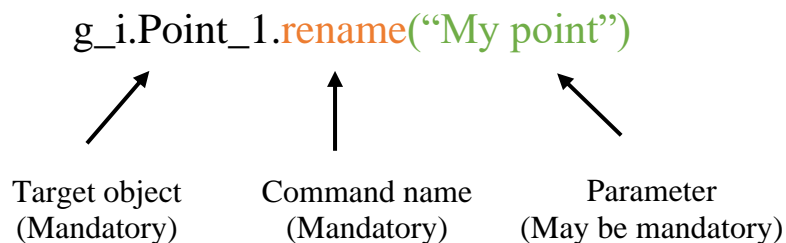
2.3. PLAXIS Remote Scripting interface Via Python

When many calculations of similar procedure are needed, it is possible to develop a Python script and connect with PLAXIS for the numerical calculations, the remote scription is currently available for the latest versions of PLAXIS with VIP licences, nevertheless, it needs great deals of time and memory for the storage of the calculations.

The structure of the command line is not complicated to understand, and basic knowledge of programming is needed to get its equivalents in Python language, some references for using Phyton in engineering problems come from Langtangen, 2016; Bautista, 2014 and Summerfield, 2008. As an example, from PLAXIS 2D manuals, the following anatomy of the command line is converted into Python language.



a) PLAXIS command



b) Python equivalent in PLAXIS

Figure 10.- Command anatomy for PLAXIS and Python equivalent in PLAXIS (PLAXIS 2D Reference Manual, 2018)

Notice that PLAXIS rather spaces between commands than periodic separations as Python does. The command name is basically an action to be executed, and it is given as the first word in the command line, however, when scripting into Python, this command is invoked as a global command (e.g. after `g_i`). The target is an object for which the action is to be implemented, it can also be a group of objects.

Lastly, the parameter, it is extra information passed to a command when it is called, in other words they are arguments for the specified method, for example, it can be indicated the name of the action or the name of an specific variable, have into account that some of the parameters require quotation marks inside the parenthesis, and some of them does not need it, e.g. when giving a name to an object, as the name of the object is physically a phrase and based on a string argument it need quotation marks to be understood, in the contrary, when meshing an object we need to specify the size of the mesh, this is a command and at the same time is a number (variable) to be used in the calculations, thus do not need quotations marks.

The geometry, waterflow, and calculation mode are usually called as global commands, nevertheless, we must be careful when naming them at any of the modes, because PLAXIS store them according to the general index rule (Brinkgreve, et al., 2018), this means that a curve or a shape crated under the name of `Shape_1` in the geometry mode, now will be called `Shape_1_1` in the mesh or the calculation mode, this is to avoid intersections between objects with different properties and execute the calculations in a cleaner manner.

CHAPTER 3.- METHODOLOGY

This chapter summarizes all the procedures and assumptions made through the project; the treatment of the obtained data can be processed in two different ways. [1] A sensitive analysis, where the gap between the IS FoS to FE FoS expressed as percentage of the FE FoS is called error or inaccuracy, this type of analysis is straightforward and depends on there being no other differences between the FE Method and IS Method that result in appreciable differences even at large lengths or widths, this method was used by Milledge, *et al.*, (2012); and Griffiths, *et al.*, (2010) by developing an specific Numerical Method for only the analysis of Slope failure. Furthermore, an [2] insensitive analysis can be done, in this case, the first step is to find an insensitivity ratio beyond the critical length or width in a slope, this means, to express as percentage the gap between the FE FoS at any ratio to the FE FoS at the ratio where insensitivity occurs, this type of analysis is more general for any Numerical Software, and represents in a more comprehensive manner the variability or appreciable differences that software can include.

The sensitivity of the prelaminal findings will be wider through random sampling, which will be part of the Monte Carlo Simulation for the infinite length and width determination.

3.1. Infinite Slope Method for Slope Stability

The Infinite Slope method is the direct application of the Eq. 9 through each of the selected geotechnical parameters. Each Factor of Safety is stored and reused later to obtain the convergence values once calculated the FoS by the FE Method. The variation of each of the geotechnical parameters have been selected across a reasonable range, listed in Table 2, which will be held constant through all the investigation.

Table 2.- Limits of the model geotechnical parameter for the IS and FE Methods.

Parameter	Value range
Friction angle (ϕ')	20° - 37°

Cohesion (c')	0 – 20 kN/m ²
Soil depth (H)	0 – 5 m
Normalised free surface height (m)	0 – 1
Soil unit weight (γ)	10 – 22 kN/m ³
Slope angle (β)	15° - 45°

3.2. Finite Element Method for Slope Stability

The main objective from constructing a FE Model is to simulate as close as possible a translational failure mechanism in a slope, this type of failure is typical in shallow landslides. In order to achieve it, a very stiff soil layer, representing the bedrock, has been used in the model to avoid displacement in the bottom boundary. This soil layer is simulated with a Linear Elastic Model, described in the literature review [a)]. In addition, the material in the soil layer is characterized as a non-porous material in terms of the drainage condition, this is aimed to avoid water flows in the bottom soil layer, thus behave as a bedrock would do.

The soil layer aimed to be analysed is localized in the top of the model [Figure 11 and Figure 12], and has the same depth (H defined in Figure 9) at any point along the soil layer. The soil performs an Elastic – Perfectly – Plastic behaviour with a Mohr coulomb (MC) failure criterion for drained conditions, drained or long – term material, behaviour in which stiffness and strength are defined in terms of effective properties, this type of drainage was selected due to its closer representation of colluvial natural slopes.

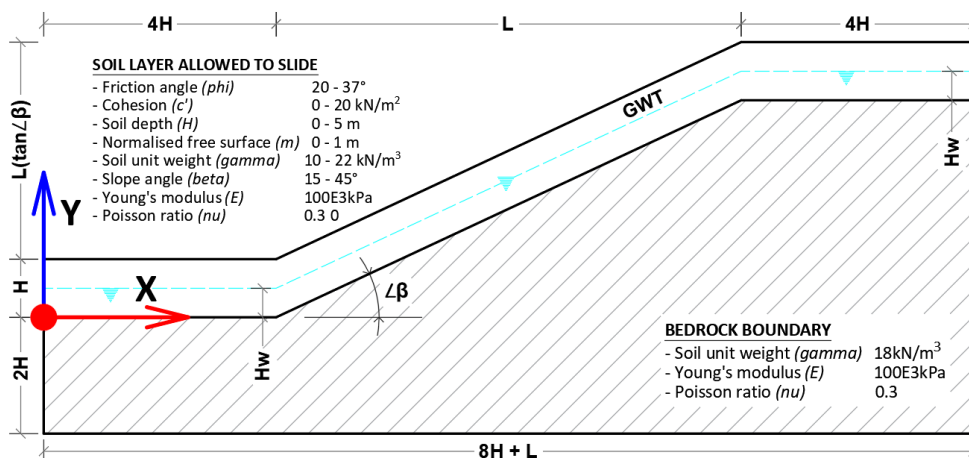


Figure 11.- PLAXIS 2D settings for the numerical modelling.

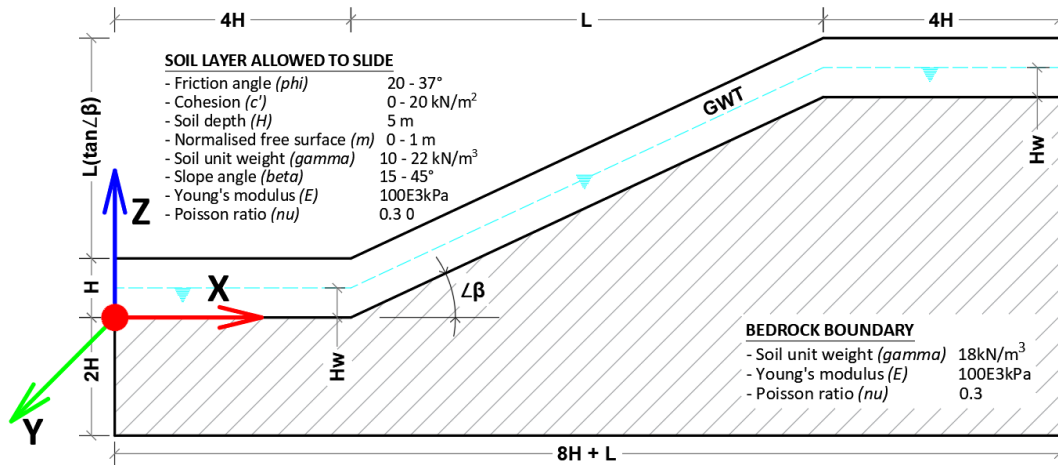


Figure 12.- PLAXIS 3D settings for the numerical modelling.

The geotechnical variation of the soil properties was defined in Table 2, and the PLAXIS modelling properties are presented in Table 3.

Table 3.- PLAXIS Modelling properties

Parameter	2D Model	3D Model
Soil Parameters		
Friction angle (ϕ')	20° - 37°	20° - 37°
Cohesion (c')	0 – 20 kN/m ²	0 – 20 kN/m ²
Soil unit weight (γ)	10 – 22 kN/m ³	10 – 22 kN/m ³
Young's modulus (E)	100e3kPa	100e3kPa
Poisson's ratio (ν)	0.30	0.30
Dilatancy angle (ψ)	0°	0°
Geometrical properties		
Normalised free surface height (m)	0 – 1	0 – 1
Soil depth (H)	0 – 5 m	0 – 5 m
Soil length (L)	variable	variable
Soil width (W)	variable	variable
Slope angle (β)	15° - 45°	15° - 45°
Numerical calculations (2D&3D)		
	Top layer	Bottom layer
Numerical Model	Mohr - Coulomb	Linear Elastic
Drainage type	Drained	Non – porous
Meshing system	15-noded [Very Fine]	10-noded [Very Coarse]
Calculation type		
Loads generation stage	Gravity	Gravity
Safety stage	Safety	Safety
Drainage loads generation	Phreatic	Phreatic
Steps number (Ph2)	User defined	User defined

From Figure 11 & Figure 12, the geometry of the models has been set, based on the vertical depth of the layer (H defined in Figure 9), this is to ensure that all models follow similar geometrical patterns and are analysed under similar conditions. However, this setting might change depending on the size of the model, e.g. for very small models the boundaries are defined closer to the slope ($2H$ instead of $4H$), and when modelling very large models the boundaries go further enough ($16H$ instead of $4H$), so as not to affect or influence in the resultant displacements in the slope, thus the calculated FE FoS.

3.3. Data Generation by Random Analysis

To examine the sensitivity of the results to a wider range of conditions, random samples from Table 2 have been selected. Based on the time constrain and the time dedicated to the modelling stage, the research ambitious is 3000 samples for the infinite length analysis (2D models), and 1000 samples for the infinite width analysis (3D models). For the 3D-models the number of variables from Table 2 is reduced to 5, this is because along the width of the slope it is needed to held constant the Length/Depth ratio, so as to observe the influence of only one geometric parameter at the time.

Understanding that the number of models is not possible to analyse for a single person, it was decided to write a script able to read and interpret all the modelling process. To do this, PLAXIS Software works in connection with Python language through the SciTE cross – platform text editor, the general process is presented in Figure 14 & Figure 15, the detailed scripts is attached in the Appendix – A. If the attached script is going to be re-use in a new computer, it must be changed the localhost and the password from the new server, these are usually found when starting the server in the expert menu in PLAXIS software (Figure 13), and changed in the following commands at the beginning of the scrip.

```
#BOILERPLATE ACTIVATION AND PYTHON CONNECTIVITY
from plxscripting.easy import*
s_i, g_i = new_server('localhost', 10001, password = '1+KuS=tG3G>fK1z$')
```


Further in the same sequence of the script, before running it, the user have to set the number of simulations required and the soil parameters from Table 2, as an example it is presented, lines below, the section of the script where those changes must be done.

```
#MONTE CARLO DATA GENERATION
import random
numberofsimulations = 1000
for i in range (numberofsimulations):
    #Random values selection
    slope_angle = 15 + (45 - 15)*random.random()
    soil_high = 0 + (5 - 0)*random.random()
    ... similar for all soil parameters in Table 2.
```

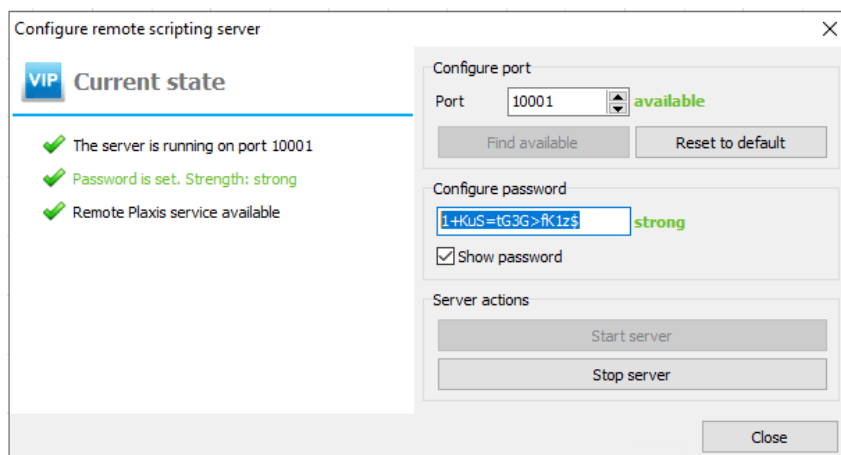


Figure 13.- Activation of the remote server in PLAXIS Software.

PLAXIS Software rather the input of the Shear Modulus of Rigidity (G), instead of the Youngs modulus (E), for that it is used the equation which defines the G in terms of E (Eq. 10), together with the other soil parameters in the following section of the script,

```
def make_topsoilmaterial(soilinfo):
    #Mohr Coulomb soil type conditions
    nu = 0.3 # Poisson's ratio
    soilmodel = 2 # Mohr - Coulomb model
    drainagetype = 0 # (0) Drained (1) Undrained (A)
    Gref = soilinfo["E"]/(2*(1+nu)) # Shear modulus (G) in terms of Young's modulus (E)
```

In this part of the script it is possible to change from Drained to undrained type A conditions by changing the number 0 to 1, nevertheless, the analysis is aimed to natural landslides, where Drained (0) conditions is a closer approximation to the mechanics in the infinite slope analysis.

Monte Carlos Simulation Flowchart (Python scripting with Plaxis 2D Interphase Connection)

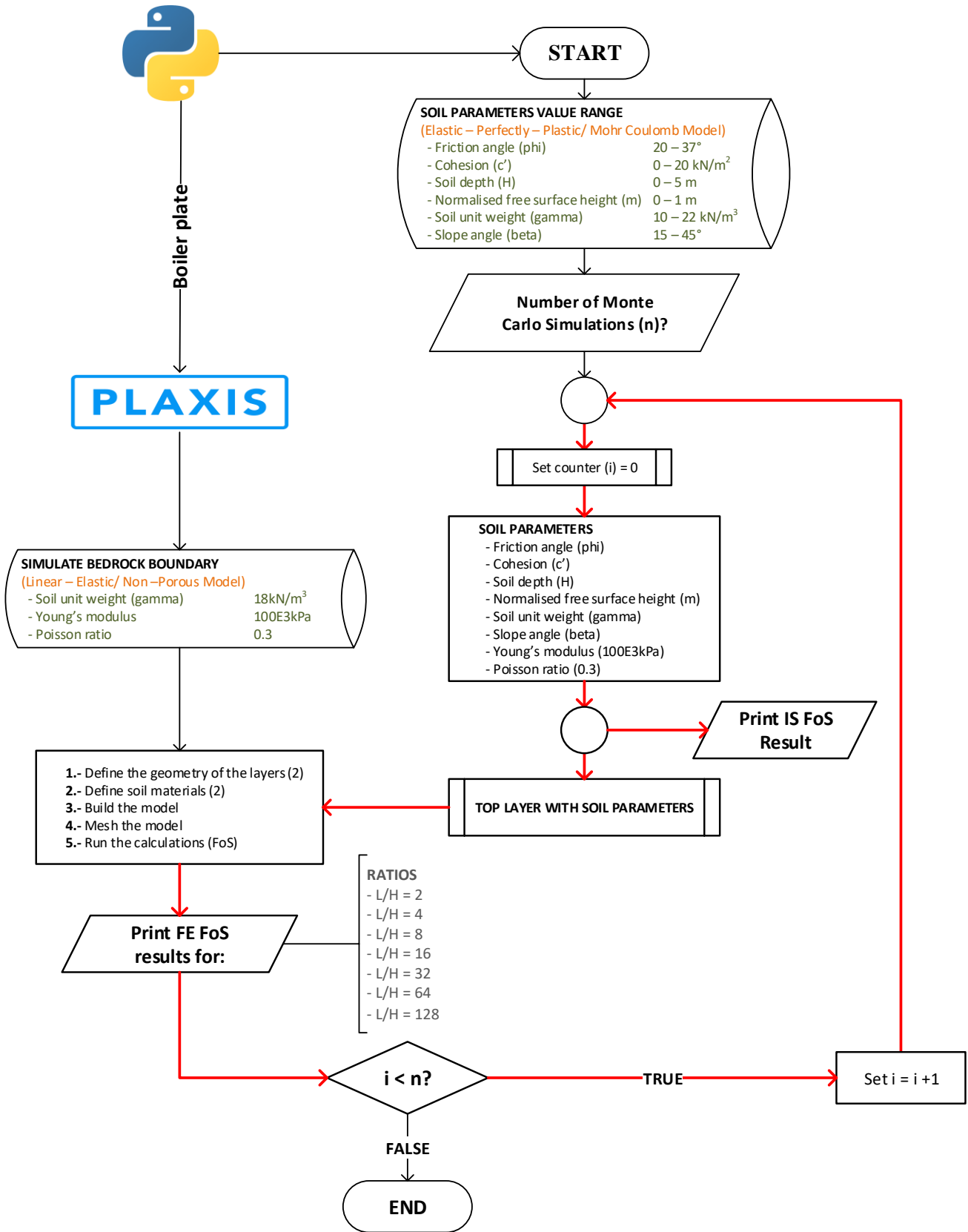


Figure 14.- Flowchart for 2D modelling in PLAXIS through Python scripting

Monte Carlos Simulation Flowchart (Python scripting with Plaxis 3D Interphase Connection)

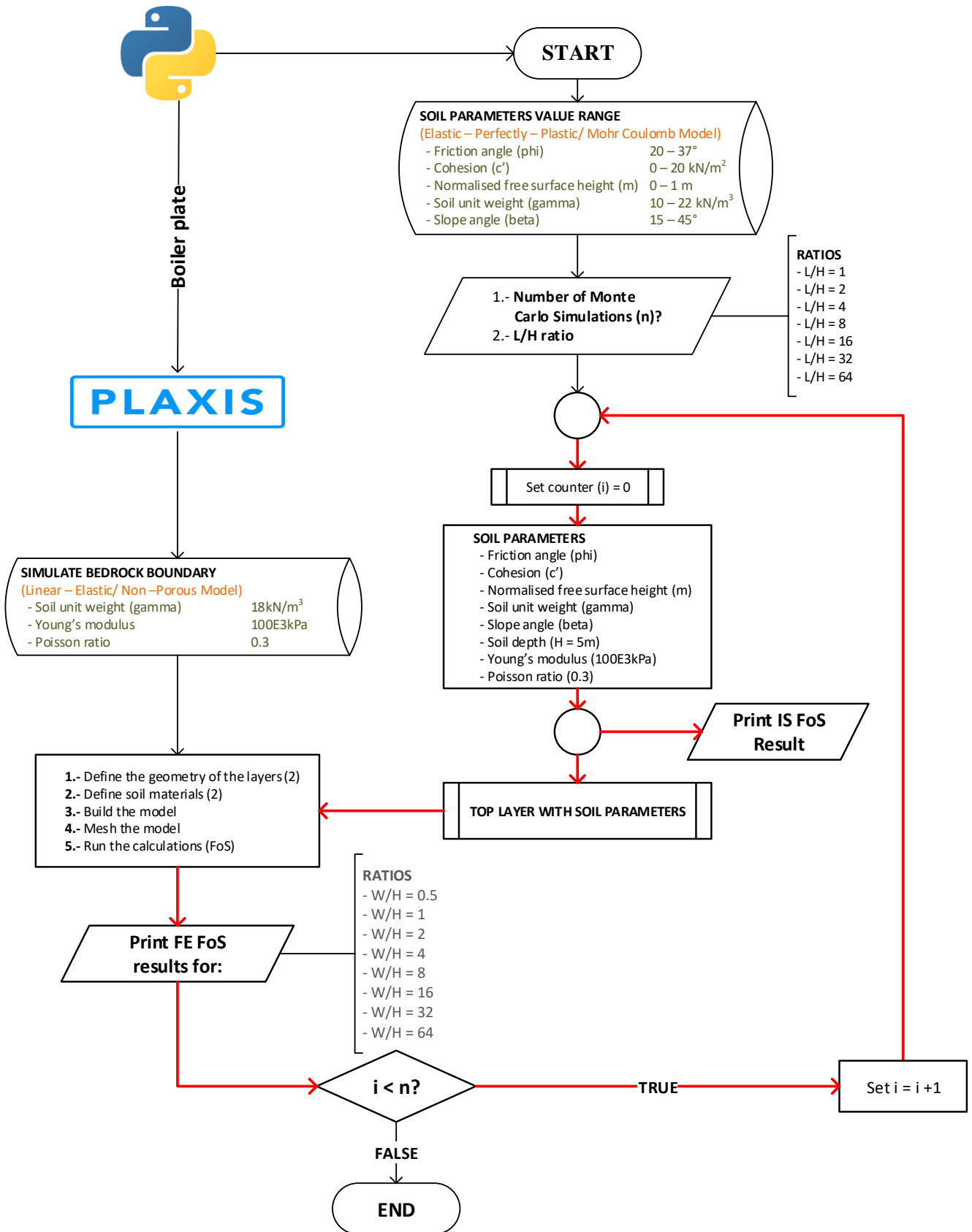


Figure 15.- Flowchart for 3D modelling in PLAXIS through Python scripting

CHAPTER 4.- RESULTS

An exploratory investigation phase (Phase 1) has been established to understand the slope behaviour under different software settings, e.g. shallow landslides simulation with translational failure mechanism, FE FoS convergence ratios, meshing and dilatancy angle influence in the modelling process. The analysis is mostly done in 2D due to the great amount of literature about the problem, then, the settings in 3D will be a replication of the models in 2D adding constant dimensions in one of the extra axes (z). Later (Phase 2), in the analysis of the sensitivity of the results to a wider range of conditions through The Monte Carlos Simulations it will be run ~3000 (2D) & ~1000 (3D) models, based on the time constrains dedicated to the modelling stage.

4.1. Phase 1, Exploratory investigation

4.1.1. Slope translational slide simulation

To ensure that the failure mechanism is translational and it is not affecting the bottom layer simulated as bedrock in Figure 11, it was set a pack of values from the defined range in Table 2. In addition, Milledge, *et al.*, (2012), determined that the geotechnical parameter that influence the most in the slope strength is the effective cohesion (c'), therefore, by restricting this parameter to the maximum possible value in the established range in Table 2 ($c' = 20\text{kN/m}^2$), it can be observed if the failure mechanism is translation and do not affect the bottom layer, proposed as bedrock.

In Figure 16 it can be observed that the model successfully simulates a translational behaviour in the slope, where the only layer allowed to slide is located in the top, in addition, to have a closer agreement with the Infinite Slope Method, the fixities of the Finite Element Model is fully in the button (Y_{\min}) and free in the top (Y_{\max}), also normal in the X_{\min} and X_{\max} , it worth to notice that the boundaries are not affecting the displacements, thus they will not the FoS. The model in in Figure 16

corresponds to a L/H ratio of 8, with effective cohesion (c') equal 20kN/m^2 , effective shear angle (ϕ') of 30° , unit weight of soil (γ) equal to 19 kN/m^3 , soil depth (H) of 5m , slope angle (β) of 25° and the normalised free surface height (m) equal to 0.40 , this is also called ground water table (GWT).

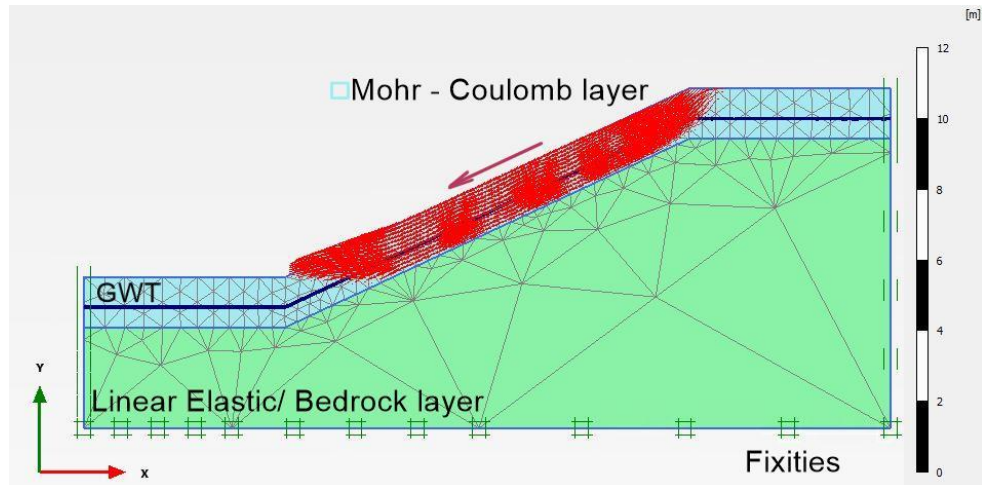


Figure 16.- Shallow landslide simulated in PLAXIS 2D for translational failure mechanism, $L/H = 8$.

4.1.2. FoS normalization to the FE FoS

As expected and explained in the methodology each pack of geotechnical values from the range in Table 3 will give a different point of convergence to the IS FoS, to normalize this, it was explained that could be done a sensitive analysis where the gap between the FE FoS and the FE FoS can be expressed as a percentage, however, this analysis depends on there being no other differences between the FE FoS and the IS FoS that could result in appreciable differences even at large lengths or width, which is not the case when working with PLAXIS Software. To avoid this misperception it will be done an insensitive analysis by choosing the ratio where the gradient of the convergence curve becomes nearly zero and normalize all the other ratios to this point, this can be done on the available options of 15-noded mesh and 6-noded mesh in PLAXIS 2D, as a way of showing a convergence point where insensitive occurs. For the models presented in since Figure 17 to Figure 23, the geotechnical parameters from Table 4 will be used,

furthermore, the of ratios to be analysed have been selected as exponential numbers, e.g. $L/H = 2, 4, 8, 16, 32, 64 \text{ \& } 128$, so as to have most of the calculations at ratios where it is believed to be greater mismatch between the FE FoS and IS FoS, e.g. $L/H < 25$.

Table 4.- Finite Element Method, trial model.

Parameter	Value range
Friction angle (ϕ')	30°
Cohesion (c')	20 kN/m^2
Soil depth (H)	5 m
Normalised free surface height (m)	0.40
Soil unit weight (γ)	19 kN/m^3
Slope angle (β)	25°

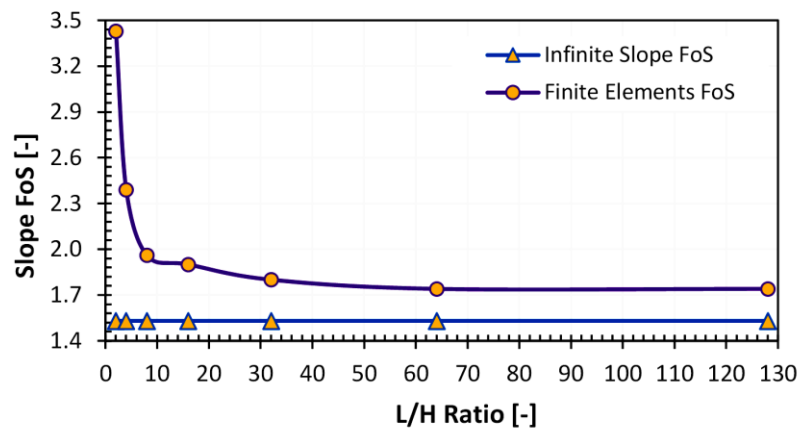


Figure 17.- FoS for different L/H ratios predicted from IS and FE methods, termed here a convergence curve, in this case for a 6-noded mesh with the parameters shown in Table 5.

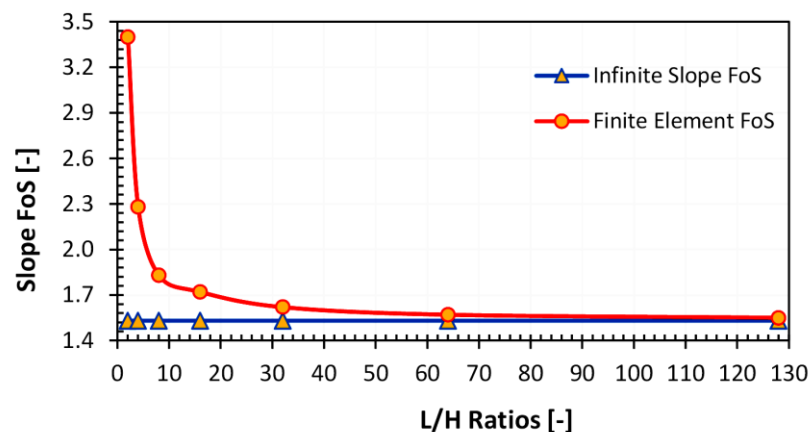


Figure 18.- FoS for different L/H ratios predicted from IS and FE methods, termed here a convergence curve, in this case for a 15-noded mesh with the parameters shown in Table 5.

It is appreciable that the 15-noded mesh is more accurate and capture at higher levels the variation of the FoS along the different L/H ratios, in the other hand, Although the convergence curve mismatch at greater extent the IS FoS curve, it performed better in the calculations, e.g. less computational time required due to the smaller amount of finite elements to be calculated. Looking for the ratio where insensitive occurs, both curves (Figure 17 & Figure 18) are normalized to the ratio L/H = 128, this means, to express the difference of the FE FoS between each of the L/H ratios to the L/H ratio of 128 as a percentage of the ratio L/H = 128, since this section, this ratio will be called NRL and will be defined by the Eq. 11.

$$NRL = \frac{FE\ FoS_{ratio\ i} - FE\ FoS_{ratio\ of\ 128}}{FE\ FoS_{ratio\ of\ 128}} \times 100 \quad Eq. 11$$

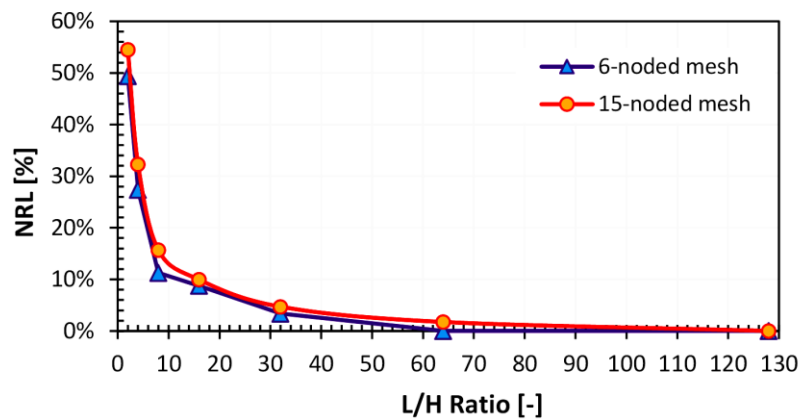


Figure 19.- Normalization of the curves in Figure 17 & Figure 18 to the L/H ratio of 128.

Although both curves match closely most of the time, the 15-noded mesh is able to capture more accurately the slope FoS, this is due to the discretization scale, a 15-noded mesh divides each of the triangular elements in 15 nodes with 12 Gauss quadrature points, in contrast, the 6-noded mesh does it with just 3 Gauss quadrature points and 6 nodes. As the aim of the project is to run many models at different geometrical sizes, it is considered that the 15-noded mesh will capture most of the variations in the Finite Element Slope FoS at different L/H ratios.

4.1.3. Meshing influence in the FoS FE normalization

To discover the influence of the Meshing strategy it will be modelled the extremes of the meshing software availabilities (very coarse & very fine). For the models presented from Figure 20 to Figure 24, the geotechnical parameters from Table 4 have been used.

a) Very coarse mesh

This mesh is the larger available in PLAXIS 2D and PLAXIS 3D, it usually uses a Relative Element Size factor of $r_e = 2$, and the Target Element Dimension goes according to the size of the model and the Relative Element Factor, in PLAXIS commands it is denoted as: **mesh (0.12)** for 2D models, and **mesh (0.075, 256)** for 3D models, in **Figure 20** is presented an example of the mesh elements distribution.

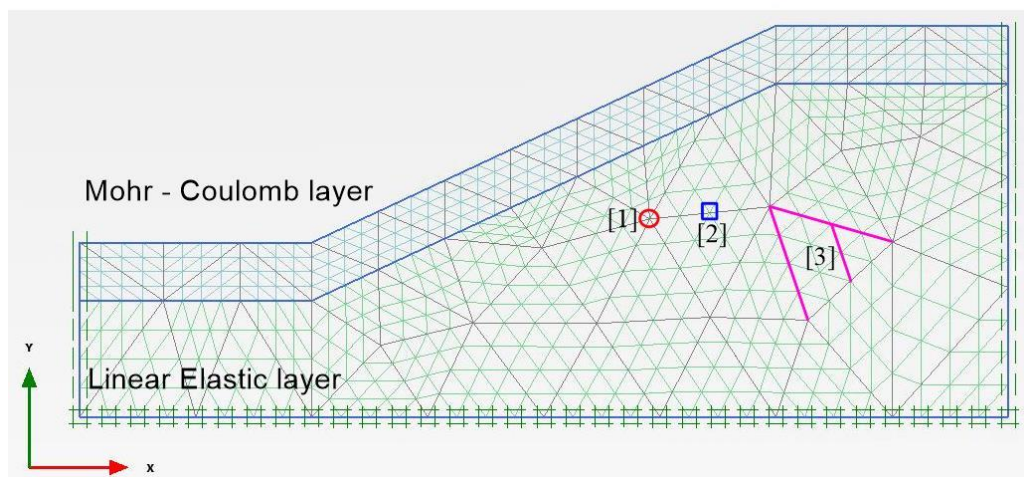


Figure 20.- Very coarse mesh in PLAXIS 2D model, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e).

b) Very fine mesh

This mesh is the finest available in PLAXIS 2D and PLAXIS 3D, it usually uses a Relative Element Size factor of $r_e = 0.5$, and the Target Element Dimension goes according to the size of the model and the Relative Element Factor, in PLAXIS

commands it is denoted as: **mesh (0.03)** for 2D models, and **mesh (0.025, 256)** for 3D models, in **Figure 21** is presented an example of the mesh elements distribution.

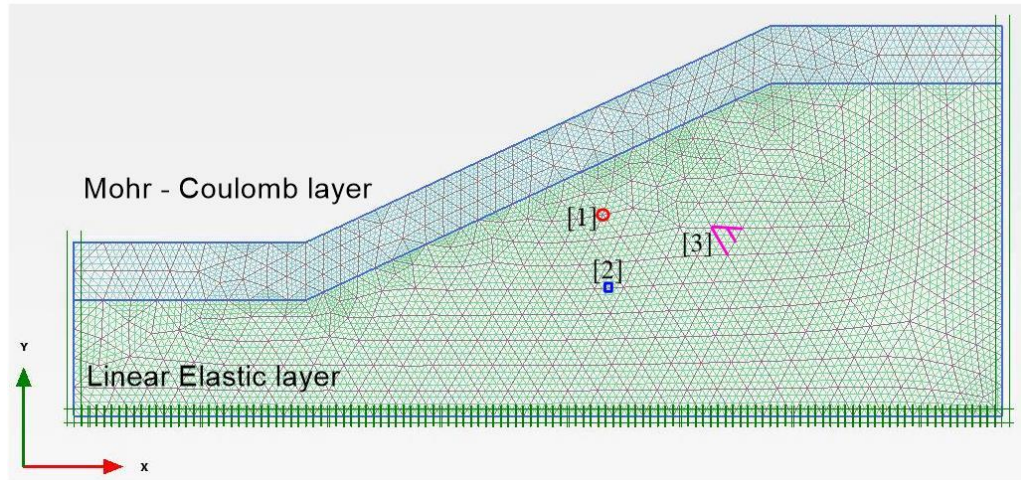


Figure 21.- Very fine mesh in PLAXIS 2D model, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e).

It is clearly that the very fine mesh produces a lot more elements for the analysis than the very coarse mesh, thus the calculations will delay more. However, the influence in the FoS is only observed when calculating the FE FoS at different ratios of L/H as presented in Table 5², furthermore, the analysis is meant to be done in terms of normalized ratios, this is the difference between each of the L/H ratios to the L/H ratio of 128 according to the Eq. 11 and expressed as NRL in Table 5 and Figure 22.

Table 5.- Size elements influence in the FE FoS

L/H	Very Coarse Mesh			Very Fine Mesh		
	FE FoS	IS FoS	NRL	FE FoS	IS FoS	NRL
2	3.51	1.53	55.87%	3.40	1.53	54.41%
4	2.44	1.53	36.53%	2.28	1.53	32.19%
8	1.93	1.53	19.81%	1.83	1.53	15.51%
16	1.73	1.53	10.34%	1.71	1.53	9.67%
32	1.62	1.53	4.73%	1.62	1.53	4.58%
64	1.56	1.53	1.07%	1.57	1.53	1.37%
128	1.55	1.53	0.00%	1.55	1.53	0.00%

² All calculations have done under 9 decimal units defined by default in PLAXIS and Python Software.



Figure 22.- Normalised ratio to the FE FoS at different mesh sizes

Although a very fine mesh captures most of the displacements in the model, it is generating a great deal of elements in the bottom layer (Figure 21), which in fact we are not interested in. To solve this, a combination of mesh sizes is presented in Figure 23, in the top layer (MC) a very fine mesh will capture all the displacements, thus a precise FoS will be calculated, and in the bottom layer a very coarse mesh is used making the performance of the calculations higher without affecting the accuracy of the FE FoS or transmitting displacements to the bottom layer.

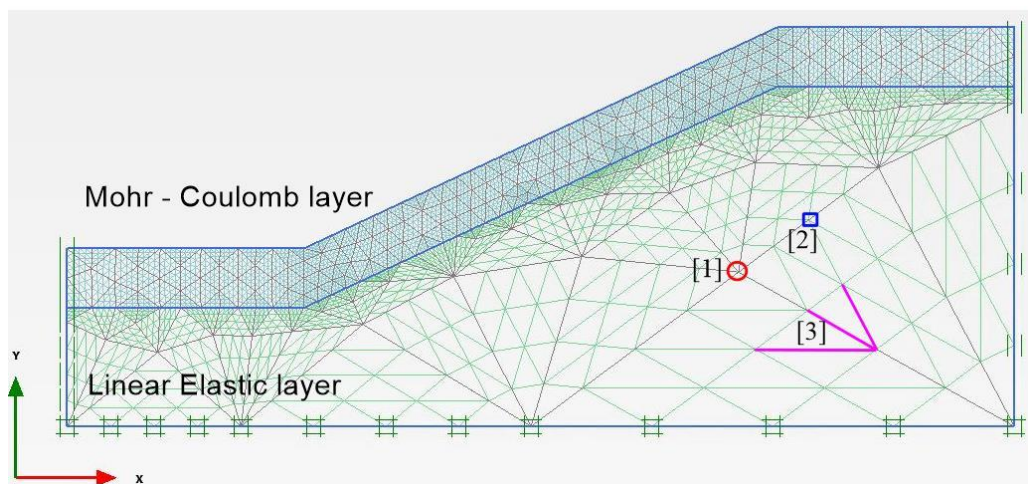


Figure 23.- Combined Mesh in PLAXIS 2D model, very fine for the Mohr – Coulomb layer and very coarse for the Linear Elastic layer, the [3] element size, [2] Gauss Point Quadrature, and [1] nodes are distributed according to the model geometry, refinement and Relative Element Size factor (r_e)

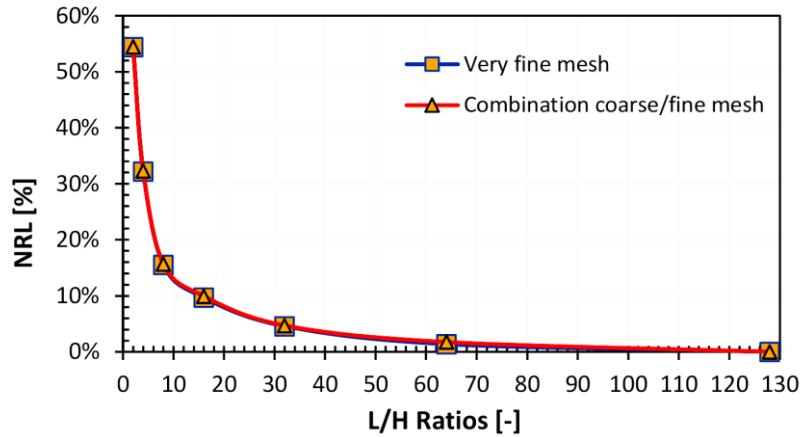


Figure 24.- Combined Mesh in PLAXIS 2D model, a very fine mesh for the top layer, and a very coarse, with coarseness factor of 4, for the bottom layer. In the plot, the very fine mesh curve is not visible because it is always below the combined mesh curve.

4.1.4. Observations in the dilatancy angle

By using the same geotechnical parameters stated in Table 4 and adding a dilatancy angle equal to the frictional angle ($\psi = \phi'$), a new finite element model in PLAXIS 2D is created, then the FE FoS results are normalised through the Eq. 11 to originate a convergence curve with dilatancy angle equal to the frictional angle. Finally, to observe the variability of this convergence curve, it is plotted together with the convergence curve without dilatancy angle ($\psi = 0^\circ$).

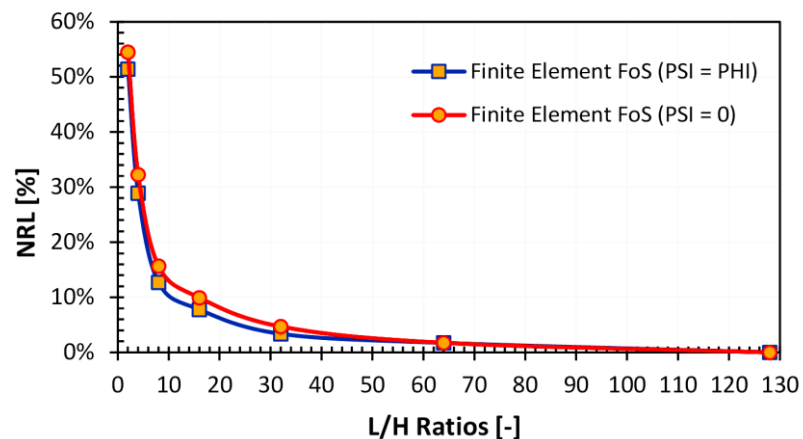


Figure 25.- Dilatancy angle variation in the Slope FE Model, both of the plotted models use the geometrical and geotechnical parameter from Table 4.

Although Dawson, *et al.*, (1999), expressed that the dilatancy angle does not have much influence in the calculations of the Slope Factor of Safety, it can be appreciated that the curves in Figure 25 mismatch for the L/H ratios below 40, this difference is not greater than 5%, even at very small ratios where the convergence curve is very sensible. In the present research project the dilatancy angle will be considered as zero ($\psi = 0^\circ$), along the Phase 2 of the investigation, a dilatancy angle of zero belongs to a non – associative flow rule in soil modelling (Wood, 2007), which guarantee more stable Factor of Safeties through a Phi/c reduction procedure in PLAXIS Software (Brinkgreve, et al., 2018).

4.1.5. 3D settings for the models

As a manner of summarising the selected options for the 2D models were presented in Table 3. In addition, in the same table it was presented in advance the geotechnical, geometrical and software parameters for the 3D models, which are basically the same as the 2D models with the exception of the mesh characteristics, which are not the same but selected its equivalents due to the lack of availability in PLAXIS 3D Software, however, it was show that the models are mesh independent due to the normalization to the largest L/H ratio (Figure 24).

Furthermore, the axis of the models change its direction to the ones stated in Figure 12, thus its fixities, e.g. fully fixed along Y_{\min} , Y_{\max} & Z_{\min} ; normally fixed along X_{\min} & X_{\max} ; and to observe the slope displacements in Z_{\max} it is set as free to slide. The configurations of the model in PLAXIS 3D with the expected displacements can be appreciated in Figure 26, although the mesh is not presented for a better appreciation of the displacements, each of the fixities is set in one of the nodes of the mesh, therefore, the difference between the top (very fine mesh) and the bottom (very coarse mesh) layer of soils in the model, as similar as proposed in the 2D models.

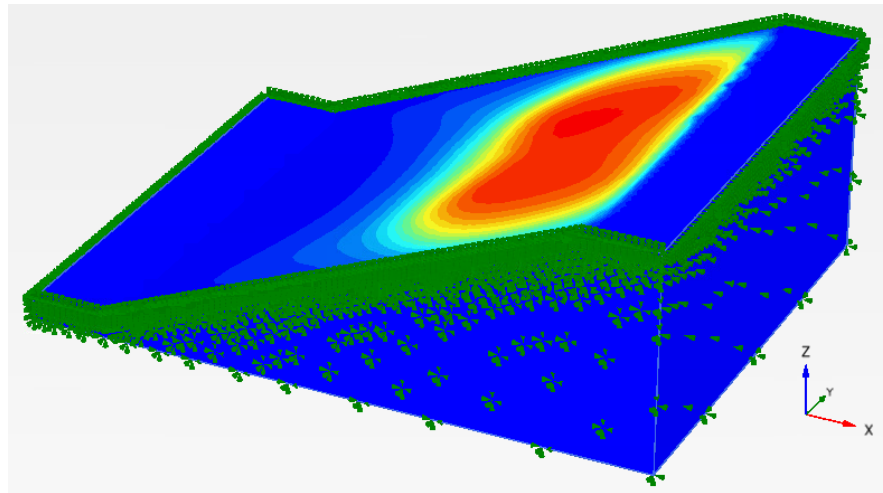


Figure 26.- PLAXIS 3D Slope model settings and expected displacements from the analysis. For the model, it has been used the values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = 64$.

A cross section of the model in Figure 26 will determine if the deformations in the top layer are behaving as expected for a shallow landslide without affecting the bottom layer consider as bedrock in the models (Figure 27).

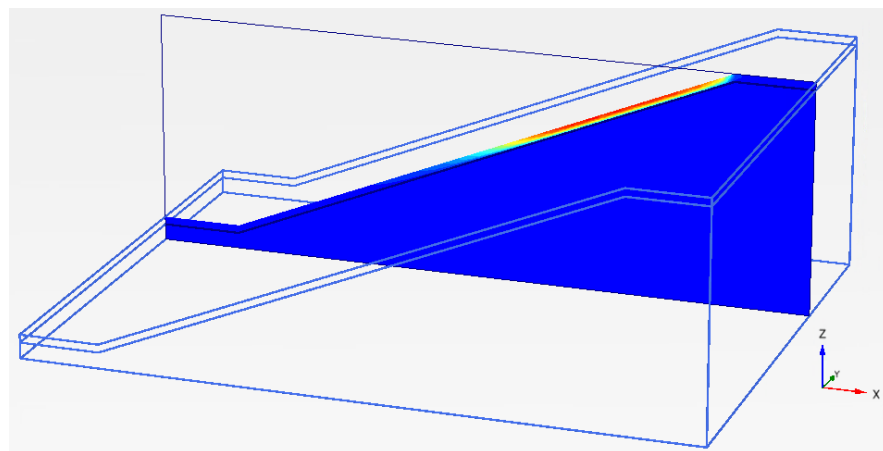


Figure 27.- Cross section of the 3D model in Figure 26, although the displacements are not notorious, it is appreciated that the displacements follow the same behaviour as in Figure 16.

To observe the convergence of the FE FoS to the IS FoS, a sequence of Width/Depth (W/H) ratios have been selected, e.g. $W/H = 1, 2, 4, 8, 16, 32$ & 64 , so as to have most of the calculations at ratios where it is believed to be greater mismatch

between the FE FoS and IS FoS, e.g. $W/H < 20$. The FE FoS and IS FoS curves are presented in Figure 28 for the L/H ratio of 64 only.

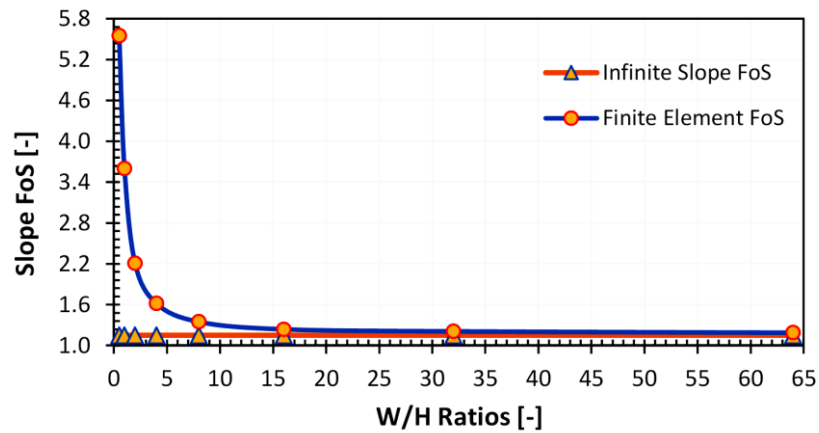


Figure 28.- Slope FE FoS and IS FoS for a 10-noded mesh in PLAXIS 3D, the model uses geometrical and geotechnical values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = 64$. convergence curve for 3D models

To localize the W/H ratio where insensitivity occurs, by looking at Figure 28, it can be determined that since the W/H ratio of 32, the convergence curve becomes almost horizontal, therefore, it will be addressed the W/H ratio of 64 as the ratio where insensitivity occurs along the width. As an example, it was normalized the curve from Figure 28 to W/H ratio of 64.

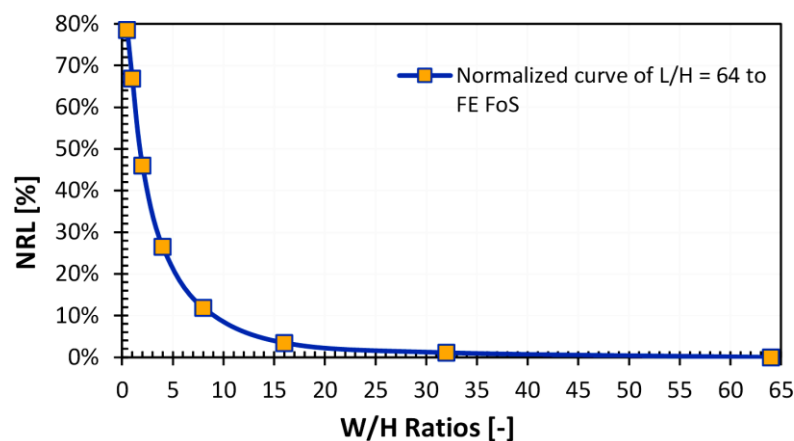


Figure 29.- Normalized curve to the W/H ratios of 64.

If L/H ratio is changed along the available L/H inputs in Figure 15, the convergence of the W/H curve is shifted up and down as shown in Figure 30. The

analysis of this curves will determine the L/H ratio at which the mismatch between the FE FoS and the IS FoS is closer and uniform for the W/H convergence curve.

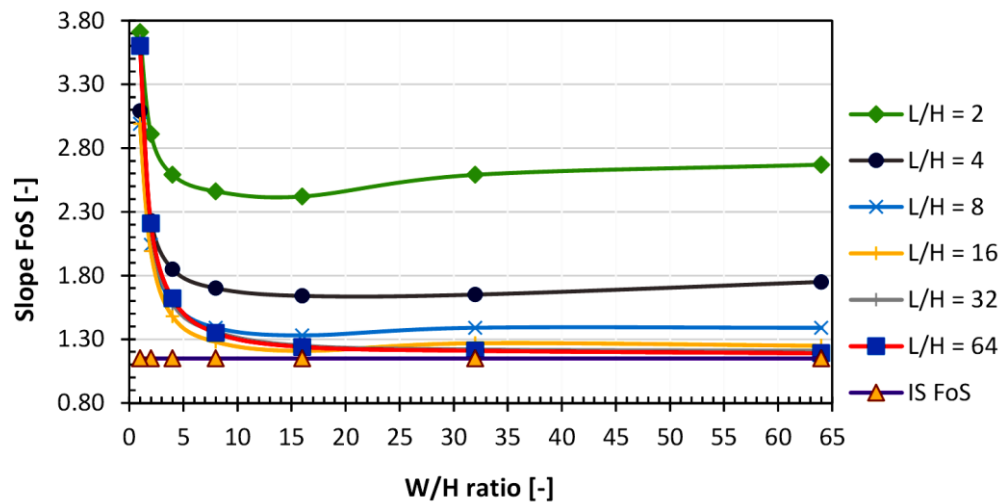


Figure 30.- Variability of W/H convergence curves at different L/H ratios, the model uses geometrical and geotechnical values in Table 4 with the exception of $m = 1$ instead of $m = 0.40$ for $L/H = [2, 4, 8, 16, 32 \text{ \& } 64]$.

In Figure 30, the best representation of the W/H convergence curve will be achieved if L/H ratio is higher than 32, due to the non – fluctuation and accuracy of the convergence curves to the IS FoS. Therefore, as expressed before, a reasonable L/H ratio to model the 3D models (for the W/H analysis), will be L/H equal to 64. Although this ratio is higher than 32, it will be needed to observe how general this can be under the variation of the geotechnical and geometrical parameters in the slope. This will be done in phase 2 by modelling ~1000 models for the length analysis.

4.2. Phase 2, Monte Carlo Simulation

a) Infinite length

As a manner of reproducing the results of Milledge, *et al.* (2012), it was generated randomly 3000 2D models for the analysis of the infinite length. The random generation was led by the flowchart presented in Figure 14, and the model configurations in Table 3,

the results of each models was then normalised to the L/H ratio of 128 and presented in Figure 33.

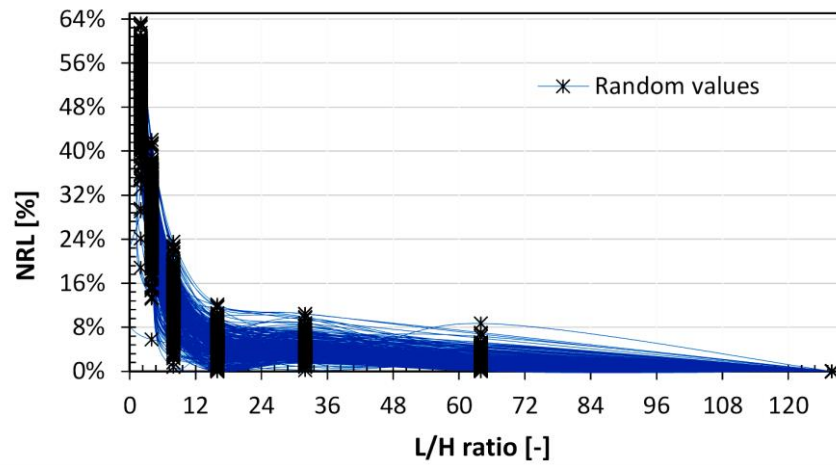


Figure 31.- Random data generated by the Monte Carlo Simulation for the infinite Length Analysis (~3000 2D models), all curves are normalized to the L/H ratio of 128.

Although the data is randomly generated, each of the curves follow a similar pattern. To synthesize and deliver a single representative curve from the data, it has been decided to work with percentiles, e.g. 50th, 90th and 95th, these curves will follow the same pattern as the convergence curves, however, they can give an insight into the accurate of any L/H ratio.

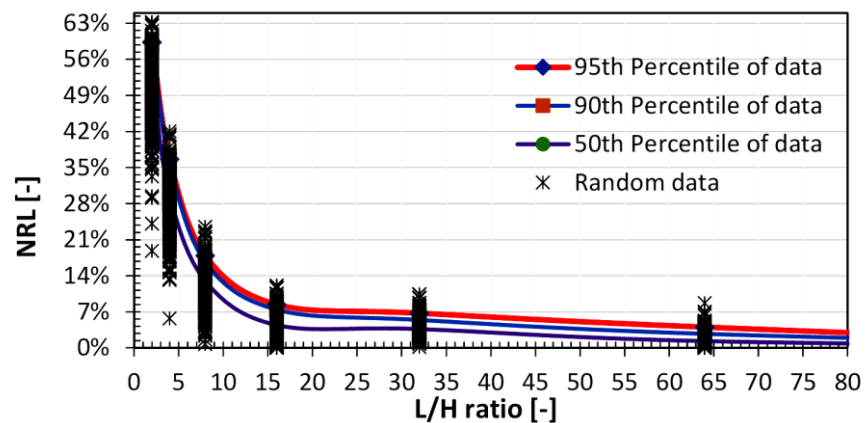


Figure 32.- Normalized curves for the 50th, 90th, 95th percentile of the random data for the infinite length analysis.

The 50th percentile represents the median of the random generated data, although it is a very good statistical representation of the data, we are interested in the convergence of

the majority of normalized curves rather than in the median of them, therefore, to evaluate the accuracy of any L/H ratio it will be better to plot the 95th percentile curve, and the threshold at 10% and 5%.

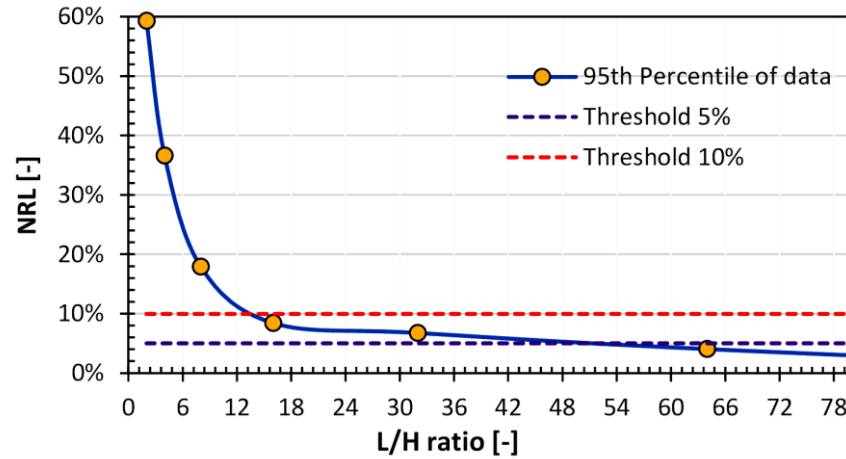


Figure 33.- Normalized curve to the FE FoS for L/H ratios at the 95th Percentile of the random data generated by the Monte Carlo Simulation. This NRL curve is crossing the 10% threshold at L/H ratio of 14 and the 5% threshold at the L/H ratio of 50.

The final results differ to the ones presented by Dr Milledge, *et al.*, (2012), e.g. infinite length dimensions for L/H ratios greater than 25 for a 5% threshold and 16 for a 10% threshold, however, these are showing that PLAXIS software is able to reproduce the convergence curve of the FE FOS to the IS FoS for the analysis of shallow landslides. Therefore, it will be suitable for modelling the infinite width as well (3D models), where, the analysis is basically a duplication of the 2D analysis, but adding the parameters stated in Table 3 and Figure 15.

b) Infinite width

At the end of Phase 1, it was told that the accuracy of the convergence will depend tightly on the L/H ratio selected, and the W/H normalization ratio, however, it requires from a sensibility analysis to generalise the L/H ratio selected. In Figure 33, it was determined that for L/H ratios greater than 50 a slope can be considered as infinite in length under the sensitivity analysis of 3000 2D models, this results are general for the 95th

percentile of the data and the 5% threshold. Therefore, for the analysis of the width, it has been selected a L/H ratio equal to 64 for the Monte Carlo Simulations (~1000 3D models), each model was then normalised to the W/H ratio of 64 and presented in Figure 34.

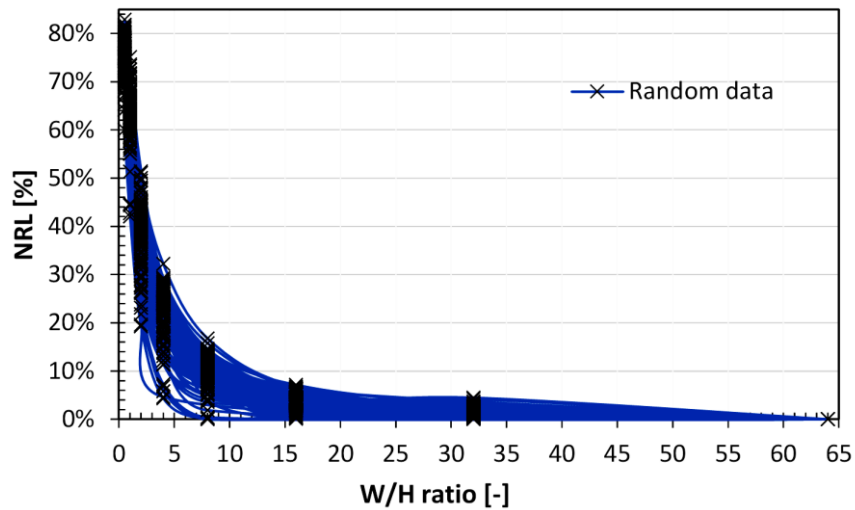


Figure 34.- Random data generated by the Monte Carlo Simulation for the infinite Width Analysis (~1000 3D models), all curves belong to the L/H ratio of 64 and are normalized to the W/H ratio of 64.

As in the 2D analysis, although the data is randomly generated, each of the curves follow a similar pattern. The synthetization of the data is done in terms of percentiles, e.g. 50th, 90th and 95th, these curves will follow the same pattern as the convergence curves, however, they can give an insight into the accurate of any W/H ratio.

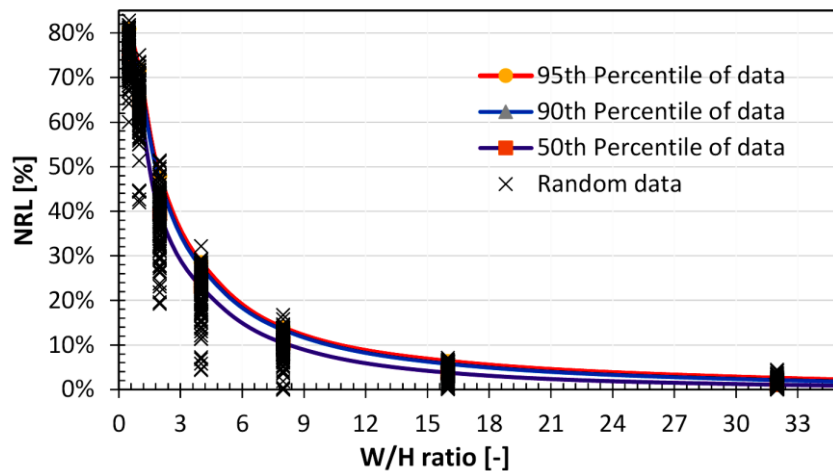


Figure 35.- Normalized curves for the 50th, 90th, 95th percentile of the random data for the infinite width analysis.

As the main interest of the research project is in the convergence of the majority of normalized curves rather than in the median of them, the evaluation of the accuracy of any W/H ratio will be represented in the plot with the 95th percentile curve, and the threshold at 10% and 5%.

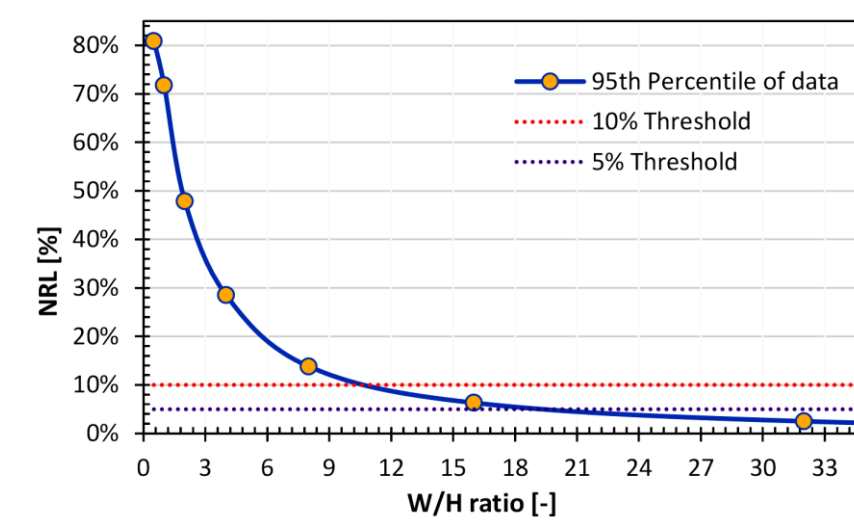


Figure 36.- Normalized curve to the FE FoS for W/H ratios at the 95th Percentile of the random data generated by the Monte Carlo Simulation. This NRL curve is crossing the 10% threshold at W/H ratio of 10 and the 5% threshold at the W/H ratio of 20.

In contrast to the length convergence curve in Figure 33, this time (Figure 36), the convergence of the FE FoS to the IS FoS is quicker, crossing the 10% threshold at ratios W/H of 10 and the 5% threshold at ratios W/H of 20. These findings are remarkably representative and indicate that the sensitivity of the width in a slope is even more sensitive than the length itself, however, these results were expected because normally in a natural slope the length of landslides is many times the size width.

CHAPTER 5.- DISCUSSION

Geometrically speaking an slope can take any value and any shape in length, width or slope angle (β), while it is commonly observed that landslide lengths always exceed widths (Marchesini, et al., 2009), it is uncertain at which measurements and under which geotechnical conditions it is going to fail. Although soil failure on natural slopes, has been proved to occur more frequently at the base of the soil layer and few meters below surface (Montgomery, et al., 2000), e.g. Figure 37, there are others of regional impacts which could reach hundreds of meter in length, width and depths, nevertheless they are rarely seen (this type of slides are out of the scope of this project thus not presented here).



Figure 37.- Slope failure after a suddenly – heavy rain, Cajamarca, Peru/2019.

The frequency of recurrence in size of landslides is an analysis that will be carry out by gathering inventories from [1] The Appalachian Mountains (Montgomery, et al., 2000); [2] Hakoishi, Japan (Paudel, et al., 2003); [3] San Grabiell Mountains, California (Rice, et al., 1969); [4] Santa Barbara County, California (Gabet & Dunne, 2002); [5] Cumbria, England (Warburton, et al., 2008); and [6] Oregon Coast Range (Montgomery, 1991). These inventories were mixed together, dividing the length & width by the depth, then counted in intervals of 5 for the L/H & W/H ratios so as to have comparable ratios as those ones found for the convergence curves (Figure 33 & Figure 36), the size of the gathered inventory is 343 landslides with L/H & W/H ranging as shown in Figure 38.

Although, most of the landslides are located in the ratio of 10 for length and width, it is worth to notice that further than the ratio of 64 the landslides recurrence is quite rare (W/H inventory), which supports and strengthens the assumption made for the Normalization Curve to the FE FoS of W/H ratio of 64 at the moment of the width (3D) analysis. On the contrary, for the L/H inventory it is still appreciable landslides further than the ratio L/H of 64, but not further than the ratio of 80, which also strengthens the assumption made for the Normalization Curve to the FE FoS of L/H ratio of 128.

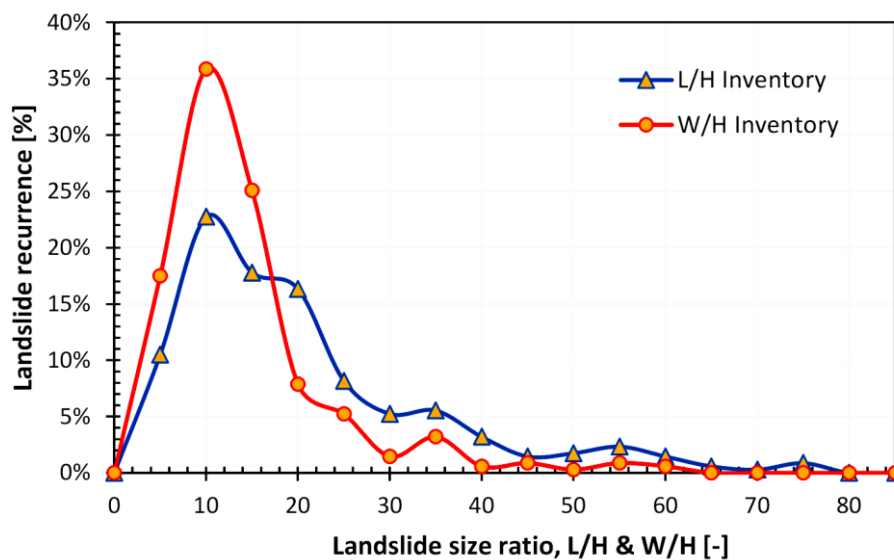


Figure 38.- Landslide size recurrence in terms of L/H & W/H ratios, these data is a collection from the following authors: [1] The Appalachian Mountains (Montgomery, et al., 2000); [2] Hakoishi, Japan (Paudel, et al., 2003); [3] San Grabiel Mountains, California (Rice, et al., 1969); [4] Santa Barbara County, California (Gabet & Dunne, 2002); [5] Cumbria, England (Warburton, et al., 2008); and [6] Oregon Coast Range (Montgomery, 1991).

To be able to tell the inaccuracy that someone can incur by using the Infinite Slope method, for any landslide, it is possible to merge Figure 33 and Figure 36 into one picture, then with the percentage recurrence of Figure 38 we can exactly figure out the breach of the Infinite Slope Method to the Finite Element Method along the length and width independently. For example, it is known that most of the natural landslides are located in the L/H and W/H ratio of 10, therefore, from Figure 39 the inaccuracy of the Calculated Factor of Safety through the

Infinite Slope Method will be about 13.8% along the length and 10% along the width. These percentages are higher in value, while the L/H or W/H is smaller, and smaller in value while the L/H or W/H increases but closer one each other, heading zero for W/H of 64 and L/H of 128. Normally, it would be considered that a landslide has fully reach its infinite dimensions once L/H and W/H result in an NRL [%] equal to zero, in any other case, there will always be a threshold of accuracy to be considered.

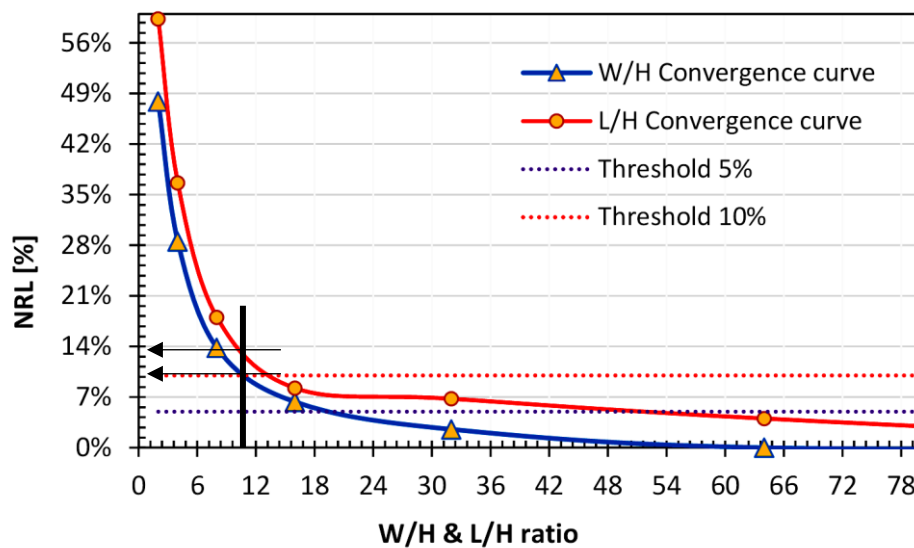


Figure 39.- Normalized curves to the FE FoS, crossing the 5% threshold at W/H ratio of 20, L/H ratio of 50 and 10% threshold at W/H ratio of 10, L/H ratio of 14.

The convergence curves, in this case, have the particularity of warning that calculations of the Factor of Safety at small W/H & L/H ratios, with the Infinite Slope Method, will underestimate the FoS at almost 50% of its FE FoS value, causing inaccuracies in those models which frequently use the IS Method as its central component for the calculations. Therefore, it would not be weird to observe hazard models which present very long landslides rather than wide, this happens due to the need of a very coarse resolution mesh required to be applied the IS Method (e.g. L/H equal 50, is equivalent to a 100x100 resolution mesh for a 2m depth landslide).

The random data generated is presented as a series of scatter plots for L/H_{crit} and W/H_{crit} in Figure 40 & Figure 41, respectively. For the length analysis it has been simulated 3000 2D

samples, where the modelled L/H ratios are 2, 4, 8, 16, 32, 64 and 128; the critical L/H ratio is defined as the ratio where each of the convergence curves crosses the threshold of 5% and 10%, then plotted in Figure 40 for each of the explored geotechnical or geometrical parameter.

The uncertainty of the plots is uniform across the L/H_{crit} of 5% and 10% threshold, but with a clear difference in the convergence critical ratio, while for the threshold of 10% most of the data is located around the L/H_{crit} of 8 and 16, for the threshold of 5% the L/H_{crit} is only 15 but with a greater scattering plots for each of the IS parameters. Although most of the IS parameters seems to be following similar scattering patterns, there is no sign of sequential or build equation that could predict the distribution of the data, neither for the 10% threshold nor for the 5% threshold.

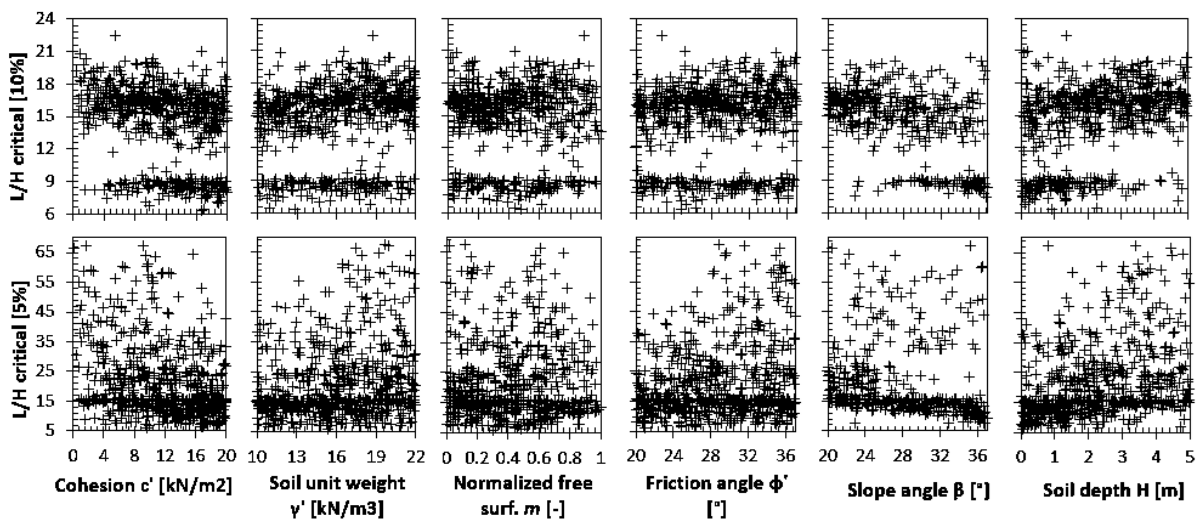


Figure 40.- Scatter plots showing the random selection of the IS parameters at which the IS predictions converge to the FE predictions within 5% and 10% threshold (L/H_{crit}), the involved geotechnical and geometrical parameters are: effective soil cohesion [kN/m^2], effective soil unit weight [kN/m^3], normalized free surface height [-], effective friction angle [$^{\circ}$], slope angle [$^{\circ}$] & soil depth [m].

On the other hand, for the width analysis it has been simulated 1000 3D samples, where the modelled W/H ratios are 0.5, 1, 2, 4, 8, 16, 32 and 64; the critical W/H ratio is defined as the ratio where each of the convergence curves crosses the threshold of 5% and 10%, then

plotted in Figure 41 for each of the explored geotechnical or geometrical parameter. In the scenario of a 3D slope analysis for the width observations, the L/H ratio must be held constant to analyse the width variations alone, therefore, the soil depth, in this case, represents a single line around the selected depth, not presented in the plots.

The uncertainty of the plots is uniform across the W/H_{crit} of 5% and 10% threshold, as alike as the L/H_{crit} plots in Figure 40, nevertheless, the data in this case is less scattered. While for the W/H_{crit} of 10% threshold most of it converges in the ratios of 9 and 16, for the 5% threshold it does in the ratio of 15.

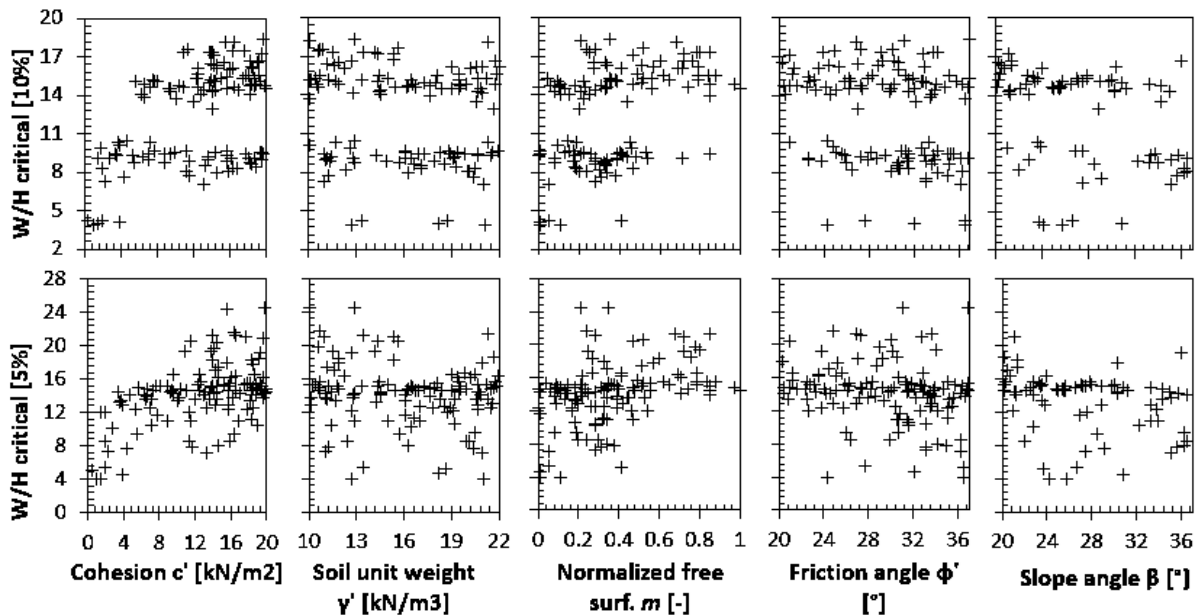


Figure 41.- Scatter plots showing the random selection of the IS parameters at which the IS predictions converge to the FE predictions within 5% and 10% threshold (W/H_{crit}), the involved geotechnical and geometrical parameters are: effective soil cohesion [kN/m²], effective soil unit weight [kN/m³], normalized free surface height [-], effective friction angle [°] & slope angle [°]

Similar to the L/H_{crit} scenario, it was not found any sign of sequential or uniform pattern that could be represented in a mathematical equation, so as to predict the distribution of the scatter geotechnical or geometrical parameters at which the IS predictions converge to the FE predictions within 5 or 10% threshold. Nevertheless, the effective cohesion seems to be and

strong dependency along the W/H_{crit} of 5%, in Figure 42 it is plotted the effective cohesion [kN/m^2] in log scale to appreciate the dependency.

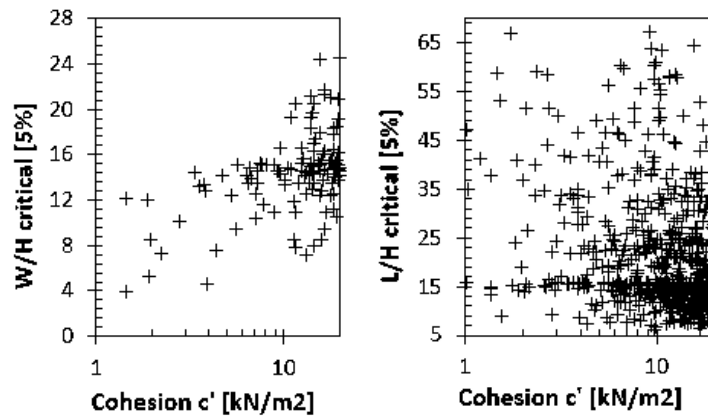


Figure 42.- Effective Cohesion dependency on the W/H_{crit} of 5%. Besides, it is plotted the L/H_{crit} of 5%. Log scale has been used for both charts.

The apparent dependency of the cohesion (c') at the W/H_{crit} of 5% [left], and not to the L/H_{crit} of 5% [right], can be explained through the Eq. 7, [$S = c' + \sigma' \tan \phi'$], where it was defined the available friction strength in the failure mechanism. In general, it can be said that the resistance generated in the failure plane depends highly stronger on the cohesion, than in the normal stresses, which is normally reduced by the frictional angle phi [$\tan \phi'$]. This dependency is not notorious in the right picture due to the setting of the numerical models, where it was defined as normally fixed, and further apart X_{min} and X_{max} , allowing the model to develop resistance only along the failure plane. While in the 3D models, it was defined fully fixed at Y_{min} and Y_{max} ³, ignoring the importance of the resistance that the lateral sides of the slides are offering.

³ Refer to Figure 26 or Figure 12, for a fully description of the axes and fixities.

CHAPTER 6.- CONCLUSIONS AND RECOMMENDATIONS

6.1. Conclusions

The research project has focused on the analysis of the validity on the infinite length and width assumption for modelling shallow landslides, this is basically to determine a limit (L/H & W/H ratio), where the Infinite Slope Method becomes unsuitable in the calculations, the reached conclusions from the analysis are as follows,

- ❖ The Stability Factor of Safety (FoS) for shallow landslides was calculated through the Infinite Slope Method (IS), and benchmarked by the Finite Element Method (FE), giving as a result that along the width of the slope the convergence curve is steeper at the beginning and almost horizontal since the ratio W/H of 32 ($\approx 0\%$ gradient). The normalized curve, from the sensitivity analysis through the Monte Carlo Simulation, led to a curve which crosses the 10% threshold at the ratio W/H of 10 and the 5% threshold at the ratio W/H of 20 for the 95th percentile of the Monte Carlos Simulations, for the width analysis was simulated 1000 3D models in PLAXIS 3D Software. On the other hand, the sensitivity analysis of the length concluded that the 10% threshold is crossed at the ratio L/H of 14 and the 5% threshold at the ratio L/H of 50 for the 95th percentile of the Monte Carlos Simulations, for the length analysis, it was simulated 3000 2D models in PLAXIS 2D Software.
- ❖ The L/H_{crit} and W/H_{crit} ratio conveys in 15 for the 5% threshold, showing similar patterns, within the critical ratio, in the scatter plots for all the geometrical and geotechnical parameters of the IS Method. Nevertheless, it seems that cohesion has a stronger dependency on the W/H_{crit} , rather than in the L/H_{crit} .
- ❖ The infinite width assumption was analysed in a range of 3D finite element models built in PLAXIS 3D/2017, a defined soil layer in the top of the models, performs an

Elastic – Perfectly – Plastic behaviour with Mohr – Coulomb failure criterion (MC), for drained conditions and dilatancy angle equal to zero (0), due to its closer agreement to the Infinite Slope FoS. Beneath the layer allowed to slide was simulated a stiff bedrock layer with a Linear Elastic behaviour for non – porous material. The initial loading calculations were performed by gravity loads generation and phreatic ground water conditions, then a safety phase was set for the calculation of the Factor of Safety through the Phi/c reduction procedure. The mesh was defined as 10-noded/tetrahedral with very fine elements for the MC soil layer and very coarse for the bedrock Linear Elastic soil layer.

- ❖ The infinite length assumption was analysed in a range of 2D finite element models built in PLAXIS 2D/2018, the model properties and soil characteristics are similar to the ones used for the infinite width assumption, the mesh was defined as 15-noded/triangular with very fine elements for the MC soil layer and very coarse for the bedrock Linear Elastic soil layer.
- ❖ A soil layer with a Linear Elastic behaviour for non – porous material, helped to simulate a bedrock, over this soil layer, the infinite slope dimensions recreate a planar failure mechanism (translational), typical in shallow landslides, under any variation of the geotechnical and geometrical parameters along the slope.
- ❖ It was randomly generated 3000 2D and 1000 3D models for the Monte Carlo Simulations to wider the sensitivity of the finding to a range of geometrical and geotechnical conditions. These parameters were automatically generated through a Python script which worked in connection with PLAXIS Software, this for non-human intervention during the random selections of the models.

- ❖ Six [6] inventories of data were collected from different authors⁴, then analysed for the recurrence and ratio preference in dimensions, getting the result that most of the landslides, without the importance of the length or width, are located in the L/H, W/H ratio of 10, and no more than 1% of the landslides is wider than the ratio of 64, which leads to the argument that the Infinite Slope Method applied to most of the landslides in the L/H & W/H ratio of 10, could incur in an inaccurate calculation of the FoS of approximately 14% for the length and 10% for the width, this benchmarked by the Finite Element Methods.

⁴[1] The Appalachian Mountains (Montgomery, et al., 2000); [2] Hakoishi, Japan (Paudel, et al., 2003); [3] San Gabriel Mountains, California (Rice, et al., 1969); [4] Santa Barbara County, California (Gabet & Dunne, 2002); [5] Cumbria, England (Warburton, et al., 2008); and [6] Oregon Coast Range (Montgomery, 1991).

6.2. Recommendations for Future Works

- ❖ A real soil layer is a complex material composed by many particles with different geotechnical properties than its closest neighbours, therefore, the assumption that the soil properties are equally distributed along the width and depth of the slope might lead to inaccurate results. The inclusion of advanced soil modelling with anisotropic distribution of the geotechnical properties will definitely adjust the limitations of the results in the current project.
- ❖ Although PLAXIS 2D & 3D is a very accurate piece of Numerical Software, it recommends, for slope analysis, that we should use advanced soil modelling (e.g. HS Small model, Soft Soil Creep Model, Soft Soil Model & NGI – ADP model), instead of a Mohr – Coulomb Model, which is First order (crude) approximation of a slope reality.
- ❖ The variable time in the present project is undoubtedly notorious, the scatter plots presented, at the end of the discussion (Figure 40 & Figure 41), should lead to the definition of patterns and mathematical equations, which normally would describe the behaviour of each of the analysed soil parameters at the L/H_{crit} o W/H_{crit} . To overcome this, it is attached the script in Python language at the Appendix – A, and instructions to be requested or download from the Author, the code can be easily implemented and left running for some months until very high accuracy occurs.

REFERENCES

- [1] Abramson, L. W., Boyce, G. M., Sharma, S. & Lee, T. S., 2002. Slope Stability and Stabilization Methods. Second ed. New York: John Wiley & Sons Inc.
- [2] Bautista, J. C., 2014. Mathematics and Python Programming. First ed. California, US: Lulu press, Inc.
- [3] Bishop, A. W., 1955. The Use of the Slip Circle in the Stability of Slopes. Amsterdam: Elsevier: Geotechnique 5, 7-17.
- [4] Brinkgreve, R. J., Kumarswamy, S., Swolfs, W. M. & Foria, F., 2017. PLAXIS 3D Manuals. Netherlands: Plaxis bv P.O. Box 572, 2600 AN DELFT.
- [5] Brinkgreve, R. J., Kumarswamy, S., Swolfs, W. M. & Foria, F., 2018. PLAXIS 2D Manuals. Netherlands: Plaxis bv P.O. Box 572, 2600 AN DELFT.
- [6] Dawson, E. M., Rtoth, W. H. & Drescher, A., 1999. 'Slope Stability Analysis by strength Reduction'. Geotechnique, 49(6), pp. 835-840.
- [7] Gabet, E. J. & Dunne, T., 2002. 'Landslides on Coastal Sage-Scrub and Grassland Hillslopes in a Severe El Nino Winter: The Effects of Vegetation Conversion on Sediment Delivery'. Geol. Soc. Am. Bull, 114(8), pp. 983-990.
- [8] Griffiths, D. V., Huang, J. & Giorgia, F. W., 2010. 'Numerical and Analytical Observations on Long and Infinite Slopes'. Journal for Numerical and Analytical Methods in Geomechanics, Issue 35, pp. 569-585.
- [9] Hahn, G. J. & Shapiro, S. S., 1967. Statistical Models in Engineering. First ed. New York: Wiley.
- [10] Hammanh, R., Yacoub, T., Corkum, B. & Curran, J., 2005. A Comparison of Finite Element Slope Stability Analysis with Conventional Limitequilibrium Investigation. 58th Canadian Geomechanical and 6th joint IAH - CNC and CGS Groundwater Specialty Conferences GeoSak 2005, pp. 480-487.
- [11] Hill, R., 1950. The Mathematical Theory of Plasticity. First ed. New York: Oxford University Press.
- [12] Iverson, R. M., 2000. 'Landslide Triggered by Rain Infiltration'. Water Resour, Issue doi: 10.1029/2000WR900090, pp. 1897-1910.

- [13] Ladd, C. C., 1991. 'Stability Evaluation During Staged Construction'. *Journal of the Geotechnical Engineering Division*, Vol. 117(GT4), pp. 540-614.
- [14] Langtangen, H. P., 2016. *A Primer on Scientific Programming with Python*. Fifth ed. London, UK: Texts in Computational Science and Engineering.
- [15] Larsen, I. J., Montgomery, D. R. & Korup, O., 2010. 'Landslide Erosion Controlled by Hillslope Material'. *Nat. Geosci*, 3(4), pp. 247-251.
- [16] Marchesini, I., Corrado, C. & De Rosa, P., 2009. 'A Preliminary Method for the Evaluation of Landslides Volume at Regional Scale'. *Geoinformatica*, 13(3), pp. 277-289.
- [17] Milledge, D. G., 2008. *Modelling Shallow Landslides: The Importance of Hydrological Controls and Lateral Reinforcement*. Durham: Durham University. PhD Thesis, 466 pp.
- [18] Milledge, D. G. et al., 2014. 'A Multidimensional Stability Model for Predicting Shallow Landslide Size and Shape Across Landscapes'. *Journal of Geophysical Research: Earth Surface*, 119(doi:10.1002/2014JF003135), pp. 2481-2504.
- [19] Milledge, D. G., Griffiths, D. V., Lane, S. N. & Warburton, J., 2012. 'Limits on the Validity of Infinite Length Assumptions for Modelling Shallow Landslides'. *Earth Surface Process and Landforms*, 37(esp.3235), pp. 1158-1166 (2012).
- [20] Montgomery, D. R., 1991. 'Channel Initiation and Landscape Evolution'. Berkeley: University of California, PhD Thesis.
- [21] Montgomery, D. R., Schmidt, K. M., Greenberg, H. M. & Dietrich, W. E., 2000. 'Forest Clearing and regional Landslides'. *Geology*, 28(4), pp. 311-314.
- [22] Moore, P. J., 1970. 'The Factor of Safety Against Undrained Failure of a Slope'. *Soils and Foundations*, Vol. 10(3), pp. 81-91.
- [23] Morgan, B. A., Wicczorek, G., Campbell, R. & Gori, P., 1997. 'Debris - Flow Hazards in Areas Affected by the June 27, 1995 Storm in Madison County'. U.S. Geological Survey, Va.
- [24] Paudel, P. P. et al., 2003. 'An Assessment of Shallow Landslides Mechanism Induced by Rainfall in Hakoishi Area'. *Kyush. Forest Res*, Volume 56, pp. 122-128.

- [25] Petley, D. N. et al., 2005. 'Development of Progressive Landslide Failure in Cohesive Materials'. *Geology*, 33(3), pp. 201-204.
- [26] Rice, R. M., Crobett, E. & Bailey, R., 1969. 'Soil Slips Related to Vegetation, Topography, and Soil in South California'. *Water Resour. Res.*, Issue doi:10.1029/WR005i003p00647, pp. 647-659.
- [27] Selvi, M., 1976. 'Slope Erosion due to Extreme Rainfall: A Case Study from New Zeland'. *Geografiska Annaler. Ser A. Phys. Geogr.*, Volume 58, pp. 131-138.
- [28] Summerfield, M., 2007. *Rapid GUI Programming with Python and Qt*. First ed. United States of America: Michigan: Pearson Education, Inc.
- [29] Terzaghi, K., Peak, R. B. & Mersi, G., 1996. *Soil Mechanics in Engineering Practice*. Third ed. New York: John Wiley & Sons, INC..
- [30] Warburton, J., Milledge, D. G. & Johnson, R. M., 2008. 'Assessment of Shallow Landslides Activity Following the January 2005 Storm, Northern Cumbria'. *Proc. Cumberland Geol. Soc.*, Volume 7, pp. 263-283.
- [31] Wood, D. M., 2007. *Soil Behaviour and Critical State Soil Mechanics*. First ed. United States of America: Cambridge University Press.

APPENDIX – A/ MODEL SCRIPTS

a. Code for Plaxis 2D Models/ Includes Monte Carlo Simulation runs

Following to the next page, it is attached the script in Python language and SciTE shell interpreter, which Plaxis has by default installed in the system.

The electronic version of this code can be enquire to the author (carlos_tapia_3@hotmail.com), or directly downloaded from,



© All rights reserved by C.T Cabrera 2020 & NCL University.
May this QR Code includes expiration dates.

b. Code for Plaxis 3D Models/ Includes Monte Carlo Simulation runs

Following to the next page, it is attached the script in Python language and SciTE shell interpreter, which Plaxis has by default installed in the system.

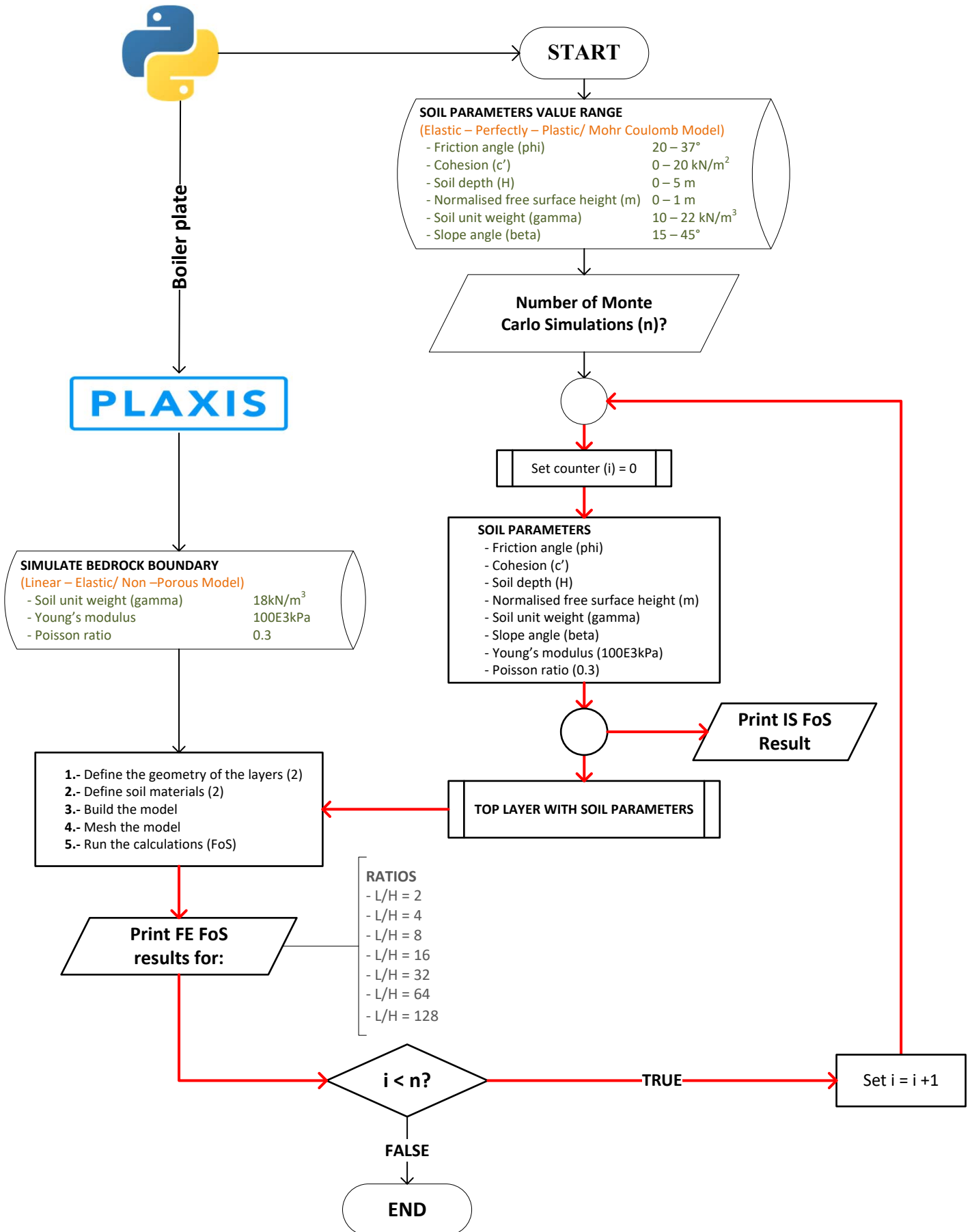
The electronic version of this code can be enquire to the author (carlos_tapia_3@hotmail.com), or directly downloaded from,



© All rights reserved by C.T Cabrera 2020 & NCL University.
May this QR Code includes expiration dates.

Monte Carlo Simulation Flowchart

(Python scripting with Plaxis 2D Interphase Connection)



```

1  #BOILERPLATE ACTIVATION AND PYTHON CONNECTIVITY
2  from plxscripting.easy import *
3  s_i, g_i = new_server('localhost', 10001, password = 'N!@PuYn~Ge645%t')
4
5  def make_geometry(H, slope, toplayer, bottomlayer):
6      #Determine model shape and build the geometry
7      if L < 10*H:
8          w_slope = 4*H
9
10         x_left = 0
11         x_toe = x_left + w_slope
12         x_crest = x_toe + L
13         x_right = x_crest + w_slope
14
15         #Structure mode:
16         g_i.gotostructures()
17         import math
18
19         #Top polygon, start bottom left, counter clockwise
20         top_points = []
21         #Bottom boundary
22         top_points.append([x_left, 0])
23         top_points.append([x_toe, 0])
24         top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
25         top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
26         #Upper boundary
27         top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H])
28         top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H])
29         top_points.append([x_toe, H])
30         top_points.append([x_left, H])
31         toppolygon, topsoil = g_i.polygon(*top_points)
32
33         #set the water table
34         g_i.gotoflow()
35         import math
36         water_flow = []
37         water_flow.append([x_left, H*m])
38         water_flow.append([x_toe, H*m])
39         water_flow.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
40         water_flow.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
41         g_i.waterlevel(*water_flow)
42
43         #Back in structures
44         g_i.gotostructures()
45         import math
46
47         #Bottom polygon, start bottom left, counter clockwise
48         bottom_points = []
49         #Bottom boundary
50         bottom_points.append([x_left, -bottomlayer["Hb"]])
51         bottom_points.append([x_right, -bottomlayer["Hb"]])
52         #Upper boundary
53         bottom_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
54         bottom_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
55         bottom_points.append([x_toe, 0])
56         bottom_points.append([x_left, 0])
57         bottompolygon, bottomsoil = g_i.polygon(*bottom_points)
58
59         #Save the polygons
60         return toppolygon, bottompolygon
61
62     elif 11*H < L < 65*H:
63         w_slope = 8*H
64
65         x_left = 0
66         x_toe = x_left + w_slope
67         x_crest = x_toe + L
68         x_right = x_crest + w_slope
69
70         #Structure mode:

```

```

71     g_i.gotostructures()
72     import math
73
74     #Top polygon, start bottom left, counter clockwise
75     top_points = []
76     #Bottom boundary
77     top_points.append([x_left, 0])
78     top_points.append([x_toe, 0])
79     top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
80     top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
81     #Upper boundary
82     top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H])
83     top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H])
84     top_points.append([x_toe, H])
85     top_points.append([x_left, H])
86     toppolygon, topsoil = g_i.polygon(*top_points)
87
88     #Water table
89     g_i.gotoflow()
90     import math
91     water_flow = []
92     water_flow.append([x_left, H*m])
93     water_flow.append([x_toe, H*m])
94     water_flow.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
95     water_flow.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
96     g_i.waterlevel(*water_flow)
97
98     #Back in structures
99     g_i.gotostructures()
100    import math
101
102    #Bottom polygon, start bottom left, counter clockwise
103    bottom_points = []
104    #Bottom boundary
105    bottom_points.append([x_left, -bottomlayer["Hb"]])
106    bottom_points.append([x_right, -bottomlayer["Hb"]])
107    #Upper boundary
108    bottom_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
109    bottom_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
110    bottom_points.append([x_toe, 0])
111    bottom_points.append([x_left, 0])
112    bottompolygon, bottomsoil = g_i.polygon(*bottom_points)
113
114    #Save the polygons
115    return toppolygon, bottompolygon
116
117 else:
118     w_slope = 16*H
119
120     x_left = 0
121     x_toe = x_left + w_slope
122     x_crest = x_toe + L
123     x_right = x_crest + w_slope
124
125     #Structure mode:
126     g_i.gotostructures()
127     import math
128
129     #Top polygon, start bottom left, counter clockwise
130     top_points = []
131     #Bottom boundary
132     top_points.append([x_left, 0])
133     top_points.append([x_toe, 0])
134     top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
135     top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
136     #Upper boundary
137     top_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H])
138     top_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H])
139     top_points.append([x_toe, H])
140     top_points.append([x_left, H])

```

```

141     toppolygon, topsoil = g_i.polygon(*top_points)
142
143     #Water table
144     g_i.gotoflow()
145     import math
146     water_flow = []
147     water_flow.append([x_left, H*m])
148     water_flow.append([x_toe, H*m])
149     water_flow.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
150     water_flow.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
151     g_i.waterlevel(*water_flow)
152
153     #Back to structures
154     g_i.gotostructures()
155     import math
156
157     #Bottom polygon, start bottom left, counter clockwise
158     bottom_points = []
159     #Bottom boundary
160     bottom_points.append([x_left, -bottomlayer["Hb"]])
161     bottom_points.append([x_right, -bottomlayer["Hb"]])
162     #Upper boundary
163     bottom_points.append([x_right, toplayer["L"]*math.tan(slope*math.pi/180)])
164     bottom_points.append([x_crest, toplayer["L"]*math.tan(slope*math.pi/180)])
165     bottom_points.append([x_toe, 0])
166     bottom_points.append([x_left, 0])
167     bottompolygon, bottomsoil = g_i.polygon(*bottom_points)
168
169     #Save the polygons
170     return toppolygon, bottompolygon
171
172 def make_topsoilmaterial(soilinfo):
173     #Mohr Coulomb soil type conditions
174     nu = 0.3
175     soilmodel = 2 # Mohr - Coulomb model
176     drainagetype = 0 # (0) Drained (1) Undrained (A)
177     Gref = soilinfo["E"]/(2*(1+nu))
178
179     soil_params = [{"MaterialName", soilinfo["name"]},
180                   ("SoilModel", soilmodel),
181                   ("DrainageType", drainagetype),
182                   ("gammaUnsat", soilinfo["gamma"]),
183                   ("gammasad", soilinfo["gamma"]),
184                   ("Gref", Gref),
185                   ("nu", nu),
186                   ("cref", soilinfo["c"]),
187                   ("phi", soilinfo["phi"]),
188                   ("psi", 0*soilinfo["phi"])] #Dilancy angle
189
190     soil_topmaterial = g_i.soilmat(*soil_params)
191     return soil_topmaterial
192
193 def make_bottomsoilmaterial(soilinfo):
194     #Linear elastic non - porous material
195     nu = 0.3
196     soilmodel = 1 # linear elastic
197     drainagetype = 4 # non - porous
198     Gref = soilinfo["E"]/(2*(1+nu))
199
200     soil_params = [{"MaterialName", soilinfo["name"]},
201                   ("SoilModel", soilmodel),
202                   ("DrainageType", drainagetype),
203                   ("gammaUnsat", soilinfo["gamma"]),
204                   ("gammasad", soilinfo["gamma"]),
205                   ("Gref", Gref),
206                   ("nu", nu)]
207     soil_bottommaterial = g_i.soilmat(*soil_params)
208     return soil_bottommaterial
209
210 def make_model(H, slope, toplayer, bottomlayer):

```

```
211 #Unpack geometry
212 top_pg, bottom_pg = make_geometry(H, slope, toplayer, bottomlayer)
213
214 #Assign materials
215 topmaterial = make_topsoilmaterial(toplayer)
216 top_pg.Soil.Material = topmaterial
217
218 bottommaterial = make_bottomsoilmaterial(bottomlayer)
219 bottom_pg.Soil.Material = bottommaterial
220
221 def meshcalculateread():
222     #Mesh
223     g_i.gotomesh()
224     g_i.set(g_i.Polygon_1_1.CoarsenessFactor, 0.2) #Refinement
225     g_i.set(g_i.Polygon_2_1.CoarsenessFactor, 4) #Coarseness
226     g_i.mesh(0.12) #Very Coarse Mesh
227
228 #Calculation Conditions
229 g_i.gotostages()
230
231 #Initial phase
232 g_i.initialPhase.DeformCalcType = "Gravity loading" #K0 procedure"
233 g_i.InitialPhase.PorePresCalcType = "Phreatic"
234
235 g_i.activate(g_i.soils, g_i.initialPhase)
236 g_i.activate(g_i.geometry, g_i.initialPhase)
237
238 #Boundary condition
239 g_i.set(g_i.Deformations.BoundaryXMin, g_i.InitialPhase, "Normally fixed")
240 g_i.set(g_i.Deformations.BoundaryXMax, g_i.InitialPhase, "Normally fixed")
241 g_i.set(g_i.Deformations.BoundaryYMin, g_i.InitialPhase, "Fully fixed")
242 g_i.set(g_i.Deformations.BoundaryYMax, g_i.InitialPhase, "Free")
243
244 #Ground water conditions
245 g_i.set(g_i.GroundwaterFlow.BoundaryXMin, g_i.InitialPhase, "Open")
246 g_i.set(g_i.GroundwaterFlow.BoundaryXMax, g_i.InitialPhase, "Open")
247 g_i.set(g_i.GroundwaterFlow.BoundaryYMin, g_i.InitialPhase, "Closed")
248 g_i.set(g_i.GroundwaterFlow.BoundaryYMax, g_i.InitialPhase, "Open")
249
250 #Safety phase
251 safetyphase = g_i.phase(g_i.initialPhase)
252 safetyphase.DeformCalcType = "Safety"
253 safetyphase.Identification = "Factor of Safety"
254
255 safetyphase.Deform.UseDefaultIterationParams = False
256 safetyphase.Deform.MaxSteps = 50 #number of steps required for the calculations
257
258 #Run numerical calculations
259 g_i.calculate()
260
261 #Read results
262 FoS = safetyphase.Reached.SumMsf.value
263
264 return FoS
265
266 def determine_FactorOfSafety(H, slope, toplayer, bottomlayer):
267     #Create a new model
268     s_i.new()
269
270 #Model Initial conditions
271 initialsets = [{"Title", ""},
272                ("ElementType", "15-Noded"), #Allowed values are: 6noded, 15noded or ordinal value.
273                ("WaterWeight", 9.81)]
274
275 g_i.setproperties(*initialsets)
276
277 #Model construction call
278 make_model(H, slope, toplayer, bottomlayer)
279
280 #Model calculation and retrieve results call
```

```

281     FoS = meshcalculateread()
282
283     return FoS
284
285
286
287 #2D PYTHON SCRIPT FOR ALEATORY SIMULATIONS IN PLAXIS
288 #MONTE CARLO DATA GENERATION
289 import random
290 numberofsimulations = 3000
291
292 for i in range (numberofsimulations):
293     #Random values selection
294     slope_angle = 15 + (45 - 15)*random.random()
295     soil_high = 0 + (5 - 0)*random.random()
296     water = 0 + (1 - 0)*random.random()
297     cohesion = 0 + (20 - 0)*random.random()
298     phi = 20 + (37 - 20)*random.random()
299     gamma = 10 + (22 - 10)*random.random()
300
301     #Fixed values for few trials
302     #slope_angle = 25
303     #soil_high = 5
304     #water = 1
305     #cohesion = 20
306     #phi = 30
307     #gamma = 19
308
309     #Infinite slope Method
310     import math
311     import datetime
312     IS = (cohesion +
soil_high*(math.cos(slope_angle*math.pi/180)*math.cos(slope_angle*math.pi/180))*(gamma -
9.81*water)*math.tan(phi*math.pi/180))/(gamma*soil_high*math.sin(slope_angle*math.pi/180)*math.cos(s
lope_angle*math.pi/180))
313     time = datetime.datetime.now()
314     now = time.strftime("%H.%M.%S %d-%m-%y")
315
316     #For the range of values stated as (L/H = 128)
317     slope = slope_angle
318     H = soil_high
319     L = 128*H
320     m = water
321
322     toplayer = {"name": "Top",
323                 "c": cohesion,
324                 "phi": phi,
325                 "gamma": gamma,
326                 "L": L,
327                 "m": m,
328                 "E": 100e3}
329
330     bottomlayer = {"name": "Bottom",
331                    "gamma": 18,
332                    "Hb": 2*H,
333                    "m": m,
334                    "E": 100e3}
335
336     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
337     FoSL = FoS
338     if FoS > 0:
339         NRL = (FoS - FoSL)*100/FoS
340         n_p = (FoS - IS)*100/FoS
341         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 FoS_(IS): {:.2f} Started at:
{:.2f}").format(cohesion, phi, gamma, IS, now))
342         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 128 m= {:.2f} L=
{:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%").format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
343     else:
344         pass

```

```

345
346 #For the range of values stated as (L/H = 64)
347 if FoS > 0:
348     slope = slope_angle
349     H = soil_high
350     L = 64*H
351     m = water
352
353     toplayer = {"name": "Top",
354                "c": cohesion,
355                "phi": phi,
356                "gamma": gamma,
357                "L": L,
358                "m": m,
359                "E": 100e3}
360
361     bottomlayer = {"name": "Bottom",
362                   "gamma": 18,
363                   "Hb": 2*H,
364                   "m": m,
365                   "E": 100e3}
366
367     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
368     if FoS > 0:
369         NRL = (FoS - FoSL)*100/FoS
370         n_p = (FoS - IS)*100/FoS
371         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 64 m= {:.2f} L=
{: .2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
372     else:
373         pass
374     else:
375         pass
376
377 #For the range of values stated as (L/H = 32)
378 if FoS > 0:
379     slope = slope_angle
380     H = soil_high
381     L = 32*H
382     m = water
383
384     toplayer = {"name": "Top",
385                "c": cohesion,
386                "phi": phi,
387                "gamma": gamma,
388                "L": L,
389                "m": m,
390                "E": 100e3}
391
392     bottomlayer = {"name": "Bottom",
393                   "gamma": 18,
394                   "Hb": 2*H,
395                   "m": m,
396                   "E": 100e3}
397
398     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
399     if FoS > 0:
400         NRL = (FoS - FoSL)*100/FoS
401         n_p = (FoS - IS)*100/FoS
402         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 32 m= {:.2f}
L= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
403     else:
404         pass
405
406 #For the range of values stated as (L/H = 16)
407 if FoS > 0:
408     slope = slope_angle
409     H = soil_high
410     L = 16*H

```



```

411     m = water
412
413     toplayer = {"name": "Top",
414                "c": cohesion,
415                "phi": phi,
416                "gamma": gamma,
417                "L": L,
418                "m": m,
419                "E": 100e3}
420
421     bottomlayer = {"name": "Bottom",
422                   "gamma": 18,
423                   "Hb": 2*H,
424                   "m": m,
425                   "E": 100e3}
426
427     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
428     if FoS > 0:
429         NRL = (FoS - FoSL)*100/FoS
430         n_p = (FoS - IS)*100/FoS
431         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 16 m= {:.2f}
L= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
432     else:
433         pass
434     else:
435         pass
436
437     #For the range of values stated as (L/H = 8)
438     if FoS > 0:
439         slope = slope_angle
440         H = soil_high
441         L = 8*H
442         m = water
443
444         toplayer = {"name": "Top",
445                    "c": cohesion,
446                    "phi": phi,
447                    "gamma": gamma,
448                    "L": L,
449                    "m": m,
450                    "E": 100e3}
451
452         bottomlayer = {"name": "Bottom",
453                        "gamma": 18,
454                        "Hb": 2*H,
455                        "m": m,
456                        "E": 100e3}
457
458         FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
459         if FoS > 0:
460             NRL = (FoS - FoSL)*100/FoS
461             n_p = (FoS - IS)*100/FoS
462             print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 8 m= {:.2f} L=
{:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
463         else:
464             pass
465     else:
466         pass
467
468     #For the range of values stated as (L/H = 4)
469     if FoS > 0:
470         slope = slope_angle
471         H = soil_high
472         L = 4*H
473         m = water
474
475         toplayer = {"name": "Top",
476                    "c": cohesion,

```

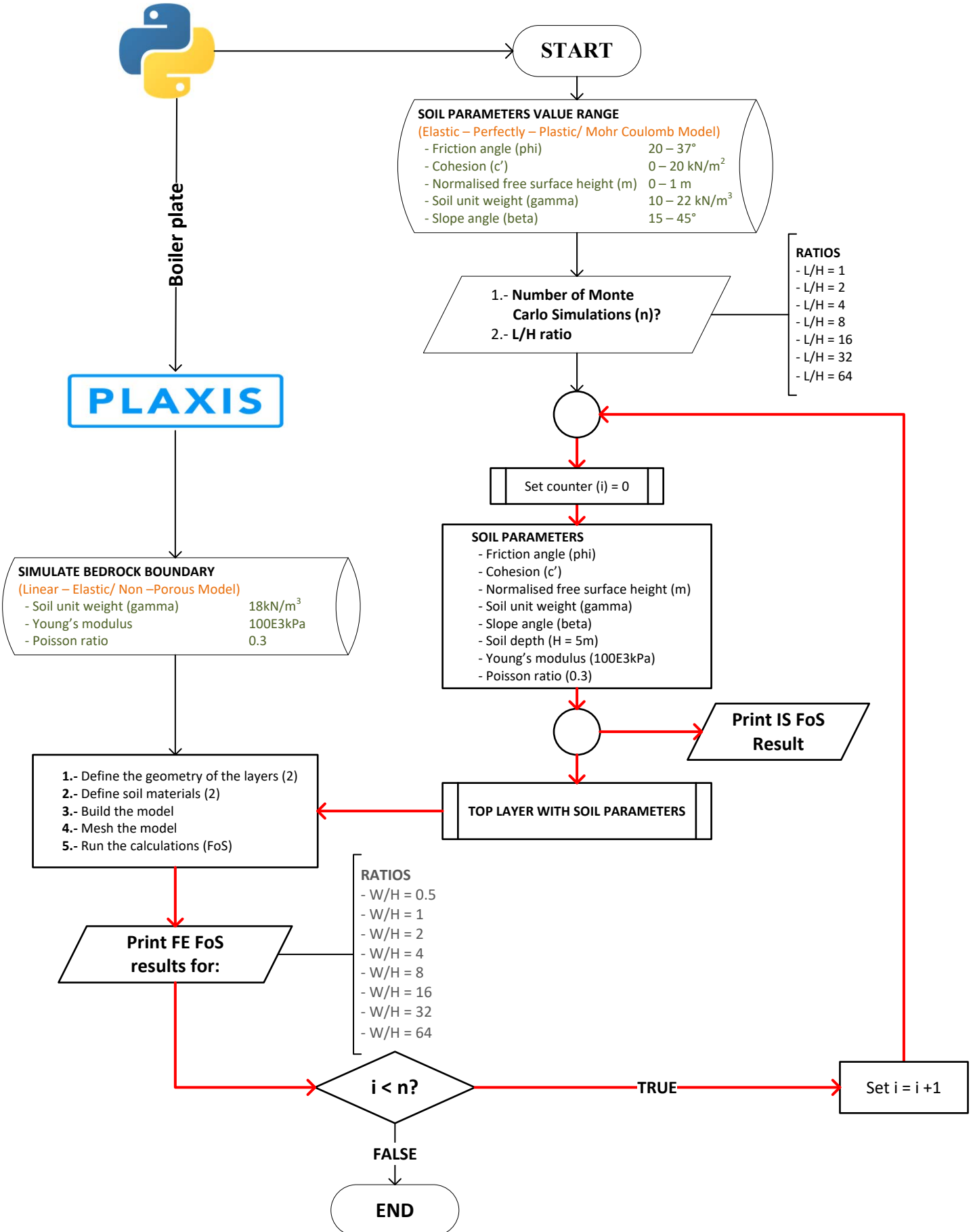
```

477         "phi": phi,
478         "gamma": gamma,
479         "L": L,
480         "m": m,
481         "E": 100e3}
482
483     bottomlayer = {"name": "Bottom",
484                   "gamma": 18,
485                   "Hb": 2*H,
486                   "m": m,
487                   "E": 100e3}
488
489     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
490     if FoS > 0:
491         NRL = (FoS - FoSL)*100/FoS
492         n_p = (FoS - IS)*100/FoS
493         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 4 m= {:.2f} L=
{:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
494     else:
495         pass
496     else:
497         pass
498
499     #For the range of values stated as (L/H = 2)
500     if FoS > 0:
501         slope = slope_angle
502         H = soil_high
503         L = 2*H
504         m = water
505
506     toplayer = {"name": "Top",
507                "c": cohesion,
508                "phi": phi,
509                "gamma": gamma,
510                "L": L,
511                "m": m,
512                "E": 100e3}
513
514     bottomlayer = {"name": "Bottom",
515                   "gamma": 18,
516                   "Hb": 2*H,
517                   "m": m,
518                   "E": 100e3}
519
520     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
521
522     if FoS > 0:
523         NRL = (FoS - FoSL)*100/FoS
524         n_p = (FoS - IS)*100/FoS
525         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= 2 m= {:.2f} L=
{:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H, m, L, FoS, n_p,
NRL))
526     else:
527         pass
528     else:
529         pass

```

Monte Carlo Simulation Flowchart

(Python scripting with Plaxis 3D Interphase Connection)



```

1 #BOILERPLATE ACTIVATION AND PLAXIS CONNECTIVITY WITH PYTHON
2 from plxscripting.easy import *
3 s_i, g_i = new_server('localhost', 10000, password = 'x>ZCTn5fB8ED7%BE')
4
5 def make_geometry(H, slope, toplayer, bottomlayer):
6     #Determine model shape and build the geometry
7     if L < 6*H:
8         w_slope = 4*H
9
10        x_left = 0
11        x_toe = x_left + w_slope
12        x_crest = x_toe + L
13        x_right = x_crest + w_slope
14
15        #Structure mode:
16        g_i.gotostructures()
17        import math
18
19        #Top polygon, start bottom left, counter clockwise
20        top_points = []
21        #Bottom boundary
22        top_points.append([x_left, 0, 0])
23        top_points.append([x_toe, 0, 0])
24        top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
25        top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
26        #Upper boundary
27        top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
28        top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
29        top_points.append([x_toe, 0, H])
30        top_points.append([x_left, 0, H])
31        toppol = g_i.surface(*top_points)
32
33        #Bottom polygon, start bottom left, counter clockwise
34        bottom_points = []
35        #Bottom boundary
36        bottom_points.append([x_left, 0, -bottomlayer["Hb"]])
37        bottom_points.append([x_right, 0, -bottomlayer["Hb"]])
38        #Upper boundary
39        bottom_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
40        bottom_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
41        bottom_points.append([x_toe, 0, 0])
42        bottom_points.append([x_left, 0, 0])
43        buttropol = g_i.surface(*bottom_points)
44
45        #Water table
46        g_i.gotowater()
47        import math
48        water_flow = []
49        water_flow.append([x_left,0, H*m])
50        water_flow.append([x_toe,0, H*m])
51        water_flow.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
52        water_flow.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
53        water_flow.append([x_right, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
54        water_flow.append([x_crest, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
55        water_flow.append([x_toe, toplayer["W"], H*m])
56        water_flow.append([x_left, toplayer["W"], H*m])
57        g_i.waterlevel(*water_flow)
58
59        #Back in structures
60        g_i.gotostructures()
61        #Create the volumes
62        toppolygon, topsoil = g_i.extrude((g_i.Polygon_1), 0, toplayer["W"], 0,)
63        bottompolygon, bottomsoil = g_i.extrude((g_i.Polygon_2), 0, toplayer["W"], 0,)
64        g_i.delete (toppol, buttropol)
65
66        return toppolygon, bottompolygon
67
68    elif 7*H < L < 17*H:
69        w_slope = 6*H
70

```

```

71     x_left = 0
72     x_toe = x_left + w_slope
73     x_crest = x_toe + L
74     x_right = x_crest + w_slope
75
76     #Structure mode:
77     g_i.gotostructures()
78     import math
79
80     #Top polygon, start bottom left, counter clockwise
81     top_points = []
82     #Bottom boundary
83     top_points.append([x_left, 0, 0])
84     top_points.append([x_toe, 0, 0])
85     top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
86     top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
87     #Upper boundary
88     top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
89     top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
90     top_points.append([x_toe, 0, H])
91     top_points.append([x_left, 0, H])
92     toppol = g_i.surface(*top_points)
93
94     #Bottom polygon, start bottom left, counter clockwise
95     bottom_points = []
96     #Bottom boundary
97     bottom_points.append([x_left, 0, -bottomlayer["Hb"]])
98     bottom_points.append([x_right, 0, -bottomlayer["Hb"]])
99     #Upper boundary
100    bottom_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
101    bottom_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
102    bottom_points.append([x_toe, 0, 0])
103    bottom_points.append([x_left, 0, 0])
104    buttropol = g_i.surface(*bottom_points)
105
106    #Water table
107    g_i.gotowater()
108    import math
109    water_flow = []
110    water_flow.append([x_left,0, H*m])
111    water_flow.append([x_toe,0, H*m])
112    water_flow.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
113    water_flow.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
114    water_flow.append([x_right, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
115    water_flow.append([x_crest, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
116    water_flow.append([x_toe, toplayer["W"], H*m])
117    water_flow.append([x_left,toplayer["W"], H*m])
118    g_i.waterlevel(*water_flow)
119
120    #Back in structures
121    g_i.gotostructures()
122    #Create the volumes
123    toppolygon, topsoil = g_i.extrude((g_i.Polygon_1), 0, toplayer["W"], 0,)
124    bottompolygon, bottomsoil = g_i.extrude((g_i.Polygon_2), 0, toplayer["W"], 0,)
125    g_i.delete (toppol, buttropol)
126
127    return toppolygon, bottompolygon
128
129 else:
130     w_slope = 10*H
131
132     x_left = 0
133     x_toe = x_left + w_slope
134     x_crest = x_toe + L
135     x_right = x_crest + w_slope
136
137     #Structure mode:
138     g_i.gotostructures()
139     import math
140

```

```

141     #Top polygon, start bottom left, counter clockwise
142     top_points = []
143     #Bottom boundary
144     top_points.append([x_left, 0, 0])
145     top_points.append([x_toe, 0, 0])
146     top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
147     top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
148     #Upper boundary
149     top_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
150     top_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H])
151     top_points.append([x_toe, 0, H])
152     top_points.append([x_left, 0, H])
153     toppol = g_i.surface(*top_points)
154
155     #Bottom polygons, start bottom left, counter clockwise
156     bottom_points = []
157     #Bottom boundary
158     bottom_points.append([x_left, 0, -bottomlayer["Hb"]])
159     bottom_points.append([x_right, 0, -bottomlayer["Hb"]])
160     #Upper boundary
161     bottom_points.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
162     bottom_points.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)])
163     bottom_points.append([x_toe, 0, 0])
164     bottom_points.append([x_left, 0, 0])
165     buttropol = g_i.surface(*bottom_points)
166
167     #Water table
168     g_i.gotowater()
169     import math
170     water_flow = []
171     water_flow.append([x_left,0, H*m])
172     water_flow.append([x_toe,0, H*m])
173     water_flow.append([x_crest, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
174     water_flow.append([x_right, 0, toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
175     water_flow.append([x_right, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
176     water_flow.append([x_crest, toplayer["W"], toplayer["L"]*math.tan(slope*math.pi/180)+H*m])
177     water_flow.append([x_toe, toplayer["W"], H*m])
178     water_flow.append([x_left, toplayer["W"], H*m])
179     g_i.waterlevel(*water_flow)
180
181     #Back in structures
182     g_i.gotostructures()
183     #Create the volumes
184     toppolygon, topsoil = g_i.extrude((g_i.Polygon_1), 0, toplayer["W"], 0,)
185     bottompolygon, bottomsoil = g_i.extrude((g_i.Polygon_2), 0, toplayer["W"], 0,)
186     g_i.delete (toppol, buttropol)
187
188     return toppolygon, bottompolygon
189
190 def make_topsoilmaterial(soilinfo):
191     #Mohr Coulomb soil type conditions
192     nu = 0.3
193     soilmodel = 2 # Mohr - Coulomb model
194     drainagetype = 0 # (0)Drained (1)Undrained(A)
195     Gref = soilinfo["E"]/(2*(1+nu))
196
197     soil_params = [{"MaterialName", soilinfo["name"]},
198                   ("SoilModel", soilmodel),
199                   ("DrainageType", drainagetype),
200                   ("gammaUnsat", soilinfo["gamma"]),
201                   ("gammasat", soilinfo["gamma"]),
202                   ("Gref", Gref),
203                   ("nu", nu),
204                   ("cref", soilinfo["c"]),
205                   ("phi",soilinfo["phi"]),
206                   ("psi",0*soilinfo["phi"])] #Dilancy angle
207
208     soil_topmaterial = g_i.soilmat(*soil_params)
209     return soil_topmaterial
210

```

```

211 def make_bottomsoilmaterial(soilinfo):
212     #Linear elastic non - porous material
213     nu = 0.3
214     soilmodel = 1     # Linear elstic
215     drainagetype = 4 # non - porous
216     Gref = soilinfo["E"]/(2*(1+nu))
217
218     soil_params = [("MaterialName", soilinfo["name"]),
219                   ("SoilModel", soilmodel),
220                   ("DrainageType", drainagetype),
221                   ("gammaUnsat", soilinfo["gamma"]),
222                   ("gammaset", soilinfo["gamma"]),
223                   ("Gref", Gref),
224                   ("nu", nu)]
225     soil_bottommaterial = g_i.soilmat(*soil_params)
226     return soil_bottommaterial
227
228 def make_model(H, slope, toplayer, bottomlayer):
229     #Unpack geometry
230     top_pg, bottom_pg = make_geometry(H, slope, toplayer, bottomlayer)
231
232     #Assign materials
233     topmaterial = make_topsoilmaterial(toplayer)
234     top_pg.Soil.Material = topmaterial
235
236     bottommaterial = make_bottomsoilmaterial(bottomlayer)
237     bottom_pg.Soil.Material = bottommaterial
238
239 def meshcalculateread():
240     #Mesh
241     g_i.gotomesh()
242     g_i.set(g_i.Volume_1_1.CoarsenessFactor, 0.2) #Refinement
243     g_i.set(g_i.Volume_2_1.CoarsenessFactor, 4) #Coarseness
244     g_i.mesh(0.075, 256) #Coarse Mesh
245
246     #Calculation conditions
247     g_i.gotostages()
248
249     #Initial phase
250     g_i.initialPhase.DeformCalcType = "Gravity loading"
251     g_i.InitialPhase.PorePresCalcType = "Phreatic"
252
253     g_i.activate(g_i.soils, g_i.initialPhase)
254     g_i.activate(g_i.geometry, g_i.initialPhase)
255
256     #Boundary condition
257     g_i.set(g_i.Deformations.BoundaryXMin, g_i.InitialPhase, "Normally fixed")
258     g_i.set(g_i.Deformations.BoundaryXMax, g_i.InitialPhase, "Normally fixed")
259     g_i.set(g_i.Deformations.BoundaryYMin, g_i.InitialPhase, "Fully fixed")
260     g_i.set(g_i.Deformations.BoundaryYMax, g_i.InitialPhase, "Fully fixed")
261     g_i.set(g_i.Deformations.BoundaryZMin, g_i.InitialPhase, "Fully fixed")
262     g_i.set(g_i.Deformations.BoundaryZMax, g_i.InitialPhase, "Free")
263
264     #Ground water conditions
265     g_i.set(g_i.GroundwaterFlow.BoundaryXMin, g_i.InitialPhase, "Open")
266     g_i.set(g_i.GroundwaterFlow.BoundaryXMax, g_i.InitialPhase, "Open")
267     g_i.set(g_i.GroundwaterFlow.BoundaryYMin, g_i.InitialPhase, "Open")
268     g_i.set(g_i.GroundwaterFlow.BoundaryYMax, g_i.InitialPhase, "Open")
269     g_i.set(g_i.GroundwaterFlow.BoundaryZMin, g_i.InitialPhase, "Closed")
270     g_i.set(g_i.GroundwaterFlow.BoundaryZMax, g_i.InitialPhase, "Open")
271
272     #Safety phase
273     safetyphase = g_i.phase(g_i.initialPhase)
274     safetyphase.DeformCalcType = "Safety"
275     safetyphase.Identification = "Factor of Safety"
276
277     if W/H < 15:
278         safetyphase.Deform.UseDefaultIterationParams = False
279         safetyphase.Deform.MaxSteps = 50 #Number of steps required for the calculations
280     elif W/H > 15:

```

```

281     safetyphase.Deform.UseDefaultIterationParams = False
282     safetyphase.Deform.MaxSteps = 25
283     else:
284         pass
285
286     #Run numerical calculations
287     g_i.calculate()
288
289     #Read results
290     FoS = safetyphase.Reached.SumMsf.value
291
292     return FoS
293
294 def determine_FactorOfSafety(H, slope, toplayer, bottomlayer):
295     #Create a new model
296     s_i.new()
297
298     #Increasing of L distance according with the geometry and model size
299     if L < 6*H:
300         ini_limits = [0, 0, 8*H + L, toplayer["w"]]
301         g_i.SoilContour.initializerectangular(*ini_limits)
302     elif 7*H < L < 17*H:
303         ini_limits = [0, 0, 12*H + L, toplayer["w"]]
304         g_i.SoilContour.initializerectangular(*ini_limits)
305     else:
306         ini_limits = [0, 0, 20*H + L, toplayer["w"]]
307         g_i.SoilContour.initializerectangular(*ini_limits)
308
309     #Model Initial conditions
310     initialsets = [("Title", ""),
311                  ("ElementType", "10-Noded"), #Allowed values are: 10noded or ordinal values
312                  ("WaterWeight", 9.81)]
313
314     g_i.setproperties(*initialsets)
315
316     #Model construction call
317     make_model(H, slope, toplayer, bottomlayer)
318
319     #Model calculation and retrieve results call
320     FoS = meshcalculateread()
321
322     return FoS
323
324
325 #3D PYTHON SCRIPT FOR ALEATORY SIMULATIONS IN PLAXIS
326 #MONTE CARLO DATA GENERATION
327 import random
328 numberofsimulations = 1000
329
330 for i in range (numberofsimulations):
331     #Random values selection
332     slope_angle = 15 + (45 - 15)*random.random()
333     water = 0 + (1 - 0)*random.random()
334     cohesion = 0 + (20 - 0)*random.random()
335     phi = 20 + (37 - 20)*random.random()
336     gamma = 10 + (22 - 10)*random.random()
337
338     #Fixed values according to each L/H required
339     soil_high = 5 #keep constant this time for all calculations
340     length = 10 #Change at constant rate each time for every L/H ratio
341
342     #Fixed values for few trials
343     #slope_angle = 25
344     #water = 0
345     #cohesion = 20
346     #phi = 30
347     #gamma = 19
348
349     #Infinite slope Method
350     import math

```



```

351 import datetime
352 IS = (cohesion +
soil_high*(math.cos(slope_angle*math.pi/180)*math.cos(slope_angle*math.pi/180))*(gamma -
9.81*water)*math.tan(phi*math.pi/180))/(gamma*soil_high*math.sin(slope_angle*math.pi/180)*math.cos(s
lope_angle*math.pi/180))
353 time = datetime.datetime.now()
354 now = time.strftime("%H.%M.%S %d-%m-%y")
355
356 #For the range of values stated as (W/H = 64)
357 slope = slope_angle
358 H = soil_high
359 L = length
360 m = water
361 W = 64*H
362
363 toplayer = {"name": "Top",
364            "c": cohesion,
365            "phi": phi,
366            "gamma": gamma,
367            "L": L,
368            "m": m,
369            "W": W,
370            "E": 100e3}
371
372 bottomlayer = {"name": "Bottom",
373               "gamma": 18,
374               "Hb": 2*H,
375               "m": m,
376               "W": W,
377               "E": 100e3}
378
379 FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
380 FoSL = FoS
381 if FoS > 0:
382     NRL = (FoS - FoSL)*100/FoS
383     n_p = (FoS - IS)*100/FoS
384     LH = L/H
385     print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 FoS_(IS): {:.2f} Started at:
{:.2f}").format(cohesion, phi, gamma, IS, now))
386     print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 64 m=
{:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%").format(cohesion, phi, gamma, slope, H, LH, m,
W, FoS, n_p, NRL))
387 else:
388     pass
389
390 #For the range of values stated as (W/H = 32)
391 if FoS > 0:
392     slope = slope_angle
393     H = soil_high
394     L = length
395     m = water
396     W = 32*H
397
398 toplayer = {"name": "Top",
399            "c": cohesion,
400            "phi": phi,
401            "gamma": gamma,
402            "L": L,
403            "m": m,
404            "W": W,
405            "E": 100e3}
406
407 bottomlayer = {"name": "Bottom",
408               "gamma": 18,
409               "Hb": 2*H,
410               "m": m,
411               "W": W,
412               "E": 100e3}
413
414 FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)

```

```

415
416     if FoS > 0:
417         NRL = (FoS - FoSL)*100/FoS
418         n_p = (FoS - IS)*100/FoS
419         LH = L/H
420         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 32
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
421     else:
422         pass
423     else:
424         pass
425
426     #For the range of values stated as (W/H = 16)
427     if FoS > 0:
428         slope = slope_angle
429         H = soil_high
430         L = length
431         m = water
432         W = 16*H
433
434         toplayer = {"name": "Top",
435                   "c": cohesion,
436                   "phi": phi,
437                   "gamma": gamma,
438                   "L": L,
439                   "m": m,
440                   "W": W,
441                   "E": 100e3}
442
443         bottomlayer = {"name": "Bottom",
444                       "gamma": 18,
445                       "Hb": 2*H,
446                       "m": m,
447                       "W": W,
448                       "E": 100e3}
449
450         FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
451
452     if FoS > 0:
453         NRL = (FoS - FoSL)*100/FoS
454         n_p = (FoS - IS)*100/FoS
455         LH = L/H
456         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 16
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
457     else:
458         pass
459     else:
460         pass
461
462     #For the range of values stated as (W/H = 8)
463     if FoS > 0:
464         slope = slope_angle
465         H = soil_high
466         L = length
467         m = water
468         W = 8*H
469
470         toplayer = {"name": "Top",
471                   "c": cohesion,
472                   "phi": phi,
473                   "gamma": gamma,
474                   "L": L,
475                   "m": m,
476                   "W": W,
477                   "E": 100e3}
478
479         bottomlayer = {"name": "Bottom",
480                       "gamma": 18,

```

```

481         "Hb": 2*H,
482         "m": m,
483         "W": W,
484         "E": 100e3}
485
486     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
487
488     if FoS > 0:
489         NRL = (FoS - FoSL)*100/FoS
490         n_p = (FoS - IS)*100/FoS
491         LH = L/H
492         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 8
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
493     else:
494         pass
495     else:
496         pass
497
498     #For the range of values stated as (W/H = 4)
499     if FoS > 0:
500         slope = slope_angle
501         H = soil_high
502         L = length
503         m = water
504         W = 4*H
505
506         toplayer = {"name": "Top",
507                    "c": cohesion,
508                    "phi": phi,
509                    "gamma": gamma,
510                    "L": L,
511                    "m": m,
512                    "W": W,
513                    "E": 100e3}
514
515         bottomlayer = {"name": "Bottom",
516                       "gamma": 18,
517                       "Hb": 2*H,
518                       "m": m,
519                       "W": W,
520                       "E": 100e3}
521
522         FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
523
524         if FoS > 0:
525             NRL = (FoS - FoSL)*100/FoS
526             n_p = (FoS - IS)*100/FoS
527             LH = L/H
528             print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 4
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
529         else:
530             pass
531     else:
532         pass
533
534     #For the range of values stated as (W/H = 2)
535     if FoS > 0:
536         slope = slope_angle
537         H = soil_high
538         L = length
539         m = water
540         W = 2*H
541
542         toplayer = {"name": "Top",
543                    "c": cohesion,
544                    "phi": phi,
545                    "gamma": gamma,
546                    "L": L,

```

```

547         "m": m,
548         "W": W,
549         "E": 100e3}
550
551     bottomlayer = {"name": "Bottom",
552                   "gamma": 18,
553                   "Hb": 2*H,
554                   "m": m,
555                   "W": W,
556                   "E": 100e3}
557
558     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
559
560     if FoS > 0:
561         NRL = (FoS - FoSL)*100/FoS
562         n_p = (FoS - IS)*100/FoS
563         LH = L/H
564         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 2
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
565     else:
566         pass
567     else:
568         pass
569
570     #For the range of values stated as (W/H = 1)
571     if FoS > 0:
572         slope = slope_angle
573         H = soil_high
574         L = length
575         m = water
576         W = 1*H
577
578     toplayer = {"name": "Top",
579                "c": cohesion,
580                "phi": phi,
581                "gamma": gamma,
582                "L": L,
583                "m": m,
584                "W": W,
585                "E": 100e3}
586
587     bottomlayer = {"name": "Bottom",
588                   "gamma": 18,
589                   "Hb": 2*H,
590                   "m": m,
591                   "W": W,
592                   "E": 100e3}
593
594     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
595
596     if FoS > 0:
597         NRL = (FoS - FoSL)*100/FoS
598         n_p = (FoS - IS)*100/FoS
599         LH = L/H
600         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H= 1
m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%" .format(cohesion, phi, gamma, slope, H, LH,
m, W, FoS, n_p, NRL))
601     else:
602         pass
603     else:
604         pass
605
606     #For the range of values stated as (W/H = 0.5)
607     if FoS > 0:
608         slope = slope_angle
609         H = soil_high
610         L = length
611         m = water
612         W = 0.5*H

```

```
613
614     toplayer = {"name": "Top",
615                "c": cohesion,
616                "phi": phi,
617                "gamma": gamma,
618                "L": L,
619                "m": m,
620                "W": W,
621                "E": 100e3}
622
623     bottomlayer = {"name": "Bottom",
624                   "gamma": 18,
625                   "Hb": 2*H,
626                   "m": m,
627                   "W": W,
628                   "E": 100e3}
629
630     FoS = determine_FactorOfSafety(H, slope, toplayer, bottomlayer)
631
632     if FoS > 0:
633         NRL = (FoS - FoSL)*100/FoS
634         n_p = (FoS - IS)*100/FoS
635         LH = L/H
636         print("C': {:.2f} kN/m2 Phi= {:.2f} Gamma= {:.2f} kN/m3 Beta: {:.2f} H= {:.2f} L/H= {:.0f} W/H=
0.5 m= {:.2f} W= {:.2f} : FoS_(FE): {:.2f} NR= {:.2f}% NRL= {:.2f}%".format(cohesion, phi, gamma, slope, H,
LH, m, W, FoS, n_p, NRL))
637     else:
638         pass
639     else:
640         pass
641
```