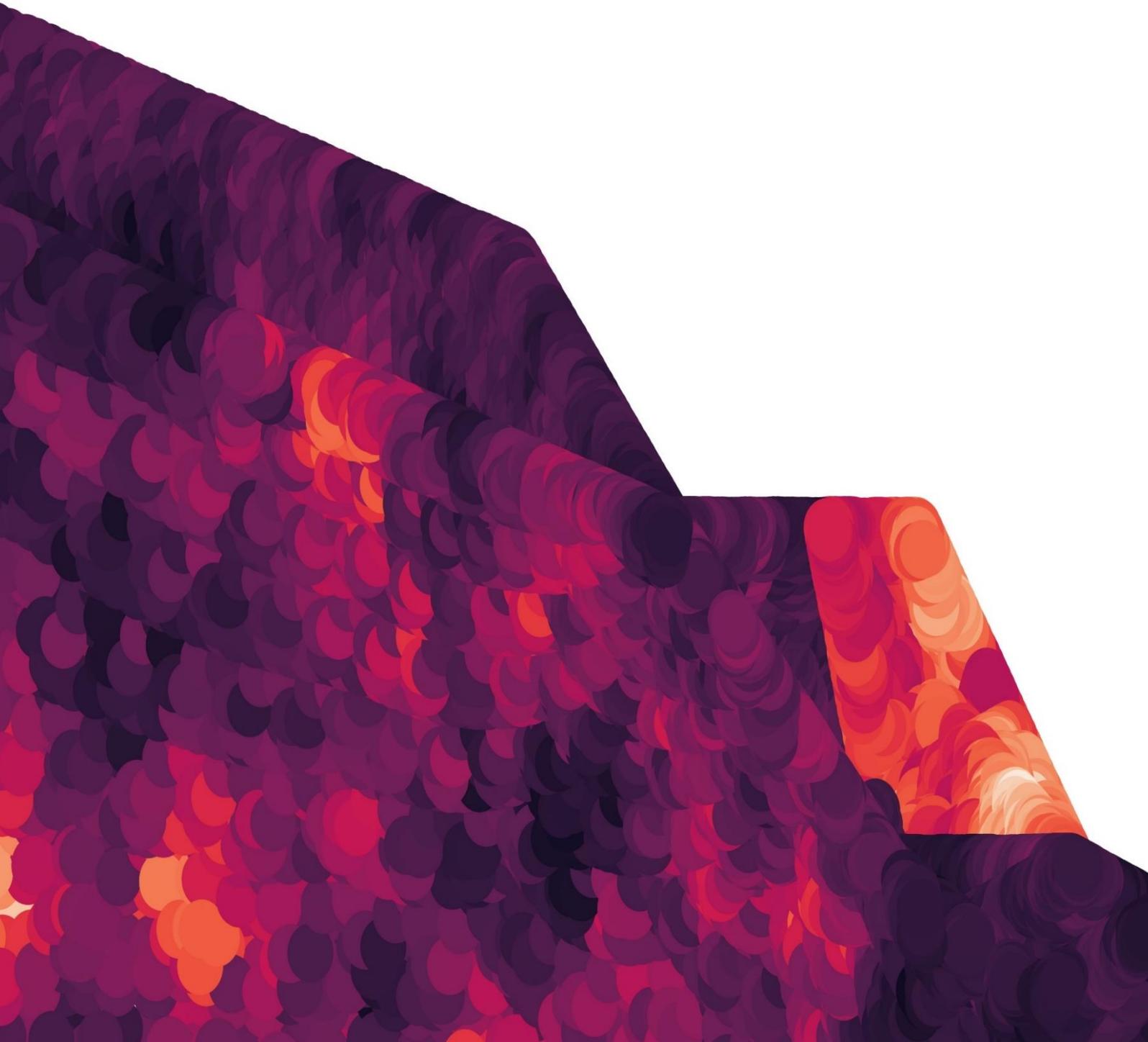


Implementation of spatial variability in PLAXIS

P.T. Pana



IMPLEMENTATION OF SPATIAL VARIABILITY IN PLAXIS

A thesis submitted to the Delft University of Technology in partial fulfillment
of the requirements for the degree of

Master of Science in Applied Earth Sciences
Geo-Engineering Track

Paul Pana

December 2022

Supervisor (Chair): Dr. ir. A.P. van den Eijnden
Supervisor: Dr.ir. R.B.J. Brinkgreve
Supervisor: Dr. M. Soleymani Shishvan

SUMMARY

Soils are some of the most irregular materials engineers handle, with the spatial variability of their properties being defined as heterogeneity. Full quantification is impossible, due to the hidden nature of the subsurface. Common design practice involves a deterministic analysis often performed using the finite element method (FEM), for which PLAXIS represents one of the most popular packages. In parallel, a scientifically explored method for dealing with medium-scale heterogeneity is the random finite element method (RFEM), implying a stochastic simulation of a problem with variable random fields from a site-specific distribution of values.

The objective of this research project is the comprehensive implementation of the concept underlying RFEM within the PLAXIS software, in order to allow it to explicitly consider spatial variability in geotechnical problems. This is done by programming the method in PLAXIS' user-defined soil modelling (UDSM) facility.

Results using the new framework compare well with literature case studies performed with existing RFEM codes. Slope reliability results as a function of spatial variability matched the trends of outcomes obtained using existing research codes. Lengthier correlation distances led to wider distributions of solutions, most decisively due to less spatial averaging allowing failure to propagate through semi-continuous weak zones. Nonetheless, larger probabilities of failure were observed with PLAXIS, possibly due to differences in meshing and algorithms employed.

The framework was also tested on a novel example, revealing the potential unlocked. Reliability and serviceability of a foundation next to a slope were simultaneously analyzed, showcasing the influence of soil properties' spatial variability on the probability to exceed performance criteria (Probability of failure - P_f). Results showed that when slope failure is deemed unlikely by deterministic analyses ($SF = 2.5$), meaningful probabilities of failure ($P_f \geq 0.25$) are identified when considering the spatial variability of influential soil properties for the ultimate limit state. The same observation holds for the serviceability analysis when accounting for heterogeneity. In both cases, for a constant vertical correlation distance, the larger the correlation distance horizontally, the larger the probability of failure. This is assumed to occur as a result of the development of weak zones underneath the foundation.

The added value provided by this study consists in unlocking the RFEM concept for general practice in the industry. By including this feature in widely utilized commercial software, professionals could bring their models closer to reality by taking into account inherent soil spatial variability.

Although this first step is a crucial nudge towards more realistic modelling in geotechnical engineering, further developments are needed. This implementation is based on the Mohr-Coulomb constitutive model. Research and development should continue towards employment with more complex constitutive models, allowing modeling with more complex features of soil behaviour.

Calculation time can also become a concern for more complex models which involve a larger number of stress points, random variables and/or phases. Further research is also required towards improving the framework's efficiency.

Keywords: Heterogeneity, RFEM, PLAXIS, user-defined soil model, scale of fluctuation (θ), Mohr-Coulomb, reliability, serviceability, probability of failure (P_f)

ACKNOWLEDGEMENTS

In the most essential form, I feel the need to express my gratitude to randomness. Besides providing me with the very conceptual basis of this thesis, it has in fact also led me on the convoluted path towards tackling this subject. Its omnipresence guided the creation of these lines just as it offered many great thinkers the basis to deliver remarkable abstract literature for us all to dwell upon. Last but surely not least, volatility governs the very stressors that allow human nature to thrive from adversity and I thank it for making me more antifragile every day.

Just as importantly, I am deeply grateful to my daily supervisors – Bram van den Eijnden and Ronald Brinkgreve – who introduced me to this fascinating topic, allowing me to acquire considerable fundamental knowledge through repeated exposure to beneficial failure. Their level of transparency and kindness in supporting me cannot and will not be understated – it will transcend in my memory through time. Moreover, I would like to thank Masoud Soleymani-Shishvan for the valuable multidisciplinary perspective he has brought to this work. To all of you, many thanks for guiding me to distill the work and its presentation to its finest version possible.

Finally, I would like to extend my sincere gratitude to my parents for supporting me from all possible perspectives throughout the various ups-and-downs my academic and life journey brought upon my path. The same goes to all the people that were there for some ideological table tennis or even a “random” chat during this emotionally charged period. Your input was crucial and oftentimes nudged my intuition towards the right direction, fact for which I cannot thank you enough.

Paul Pană

Constanța, December 2022

CONTENTS

1	Introduction	1
1.1	Background and motivation	1
1.2	Objective and research questions	4
2	Literature review	5
2.1	The Finite Element Method structure	5
2.2	Random fields and RFEM	6
3	Methodology	9
3.1	Implementation - RFEM in UDSM	9
3.1.1	RFEM and random field generation	9
3.1.2	PLAXIS UDSM Modifications	12
3.1.3	Connection RFEM-UDSM	14
3.1.4	Full code generalization	15
3.2	Spatial correlation influence on finite element mesh choices	16
4	Verification	19
4.1	Simple verification	19
4.2	Verification 1 - 2:1 $c' - \phi'$ slope	24
4.3	Verification 2 - 1:1 $c' - \phi'$ slope	29
5	Validation case study	33
5.1	Problem description	33
5.2	Results	37
6	Demonstrative case study	41
6.1	Problem description	41
6.2	Results	44
7	Good practices and remarks on functionality	47
7.1	RFEM in PLAXIS recipe	47
7.2	Operational considerations	50
8	Discussion	51
8.1	Implementation	51
8.2	Validation	53
8.3	Expansion	54
9	Conclusions and recommendations	55
9.1	Conclusions	55
9.2	Recommendations	58
	References	59
A	Random field generation	65
B	FORTTRAN Subroutine modification (UDSM)	69
C	Monte Carlo in PLAXIS	71
D	Automation script - full Starnmeer case code	73
E	Scale of fluctuation, ρ and mesh choices	83
F	CDF of SF_S - verification case 2	85
G	Random fields as a function of Θ - verification case 2	87
H	Stress points spacing - validation case study	89
I	Random fields as a function of Θ - Validation Case Study	91
J	Separate comparisons - CDF_S of SF_S - Starnmeer Validation case	93
K	Demonstrative case study: Deterministic analysis	95
L	Demonstrative Case Study: Random fields as a function of θ	97
M	Computation times overview	99

LIST OF FIGURES

Figure 1.1	Example of spatial variability of hydraulic conductivity in 3D (Yetbarek, 2020)	1
Figure 1.2	Example Finite Element Mesh from PLAXIS 2D (Bentley Systems, 2022)	2
Figure 2.1	FEM assembly structure (Zienkiewicz et al., 2013)	5
Figure 2.2	Vertical scale of fluctuation and point statistics of a random field (Samy, 2003)	7
Figure 2.3	Influence of scale of fluctuation on random field appearance - the larger the value, the smoother the transitions between values (Samy, 2003)	7
Figure 3.1	Original RFEM procedural steps	9
Figure 3.2	Flowchart for generation of random fields	11
Figure 3.3	Comparison between the formulation of the UDSM in this study for two random variables and the original MC example UDSM (Bentley Systems, 2022)	12
Figure 3.4	Comparison between the formulation of the UDSM in this study for three random variables and the original MC example UDSM (Bentley Systems, 2022)	13
Figure 3.5	Flowchart highlighting the connection of RFEM to PLAXIS	14
Figure 3.6	Procedural steps setting up the Python code automating RFEM in PLAXIS	15
Figure 3.7	3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 10 m, \theta_y = 2 m$. The closer points are to the origin, the larger the correlation, the warmer the color.	16
Figure 3.8	Parameter Markovian correlation (ρ) as a function of an anisotropic spatial correlation structure, $\theta_y = 2 m \theta_x = 10 m$	17
Figure 3.9	Influence of mesh size on spatial correlation representations using random fields of cohesion [kPa] and a combination of $\theta_y = 0.5 m \theta_x = 3 m$	18
Figure 4.1	X-Y Gradient field visualization for cohesion [kPa] in PLAXIS	20
Figure 4.2	X-Y Gradient field visualization for cohesion [kPa] in Python	20
Figure 4.3	Random field distribution for cohesion [kPa] in Python - $\theta_v = 1 m, \theta_h = 15 m$	21
Figure 4.4	Random field distribution for cohesion [kPa] in PLAXIS - $\theta_v = 1 m, \theta_h = 15 m$	21
Figure 4.5	Random field distribution for cohesion [kPa] in Python - $\theta_v = 1, \theta_h = 15 m$	22
Figure 4.6	Random field distribution for cohesion [kPa] in PLAXIS - $\theta_v = 1, \theta_h = 15 m$	22
Figure 4.7	Random field distribution for friction angle [°] in Python - $\theta_v = 1, \theta_h = 15 m$	23
Figure 4.8	Random field distribution for friction angle [°] in PLAXIS - $\theta_v = 1, \theta_h = 15 m$	23
Figure 4.9	Slope geometry of the case study used for the first verification (Allahverdizadeh et al., 2015): H = general problem dimension [m], β = slope angle [°]	24

Figure 4.10	Probabilities of failure as a function of scale of fluctuation and coefficient of variation of cohesion for the first verification (Allahverdizadeh et al., 2015)	25
Figure 4.11	Comparison of probabilities of failure between this study and the reference study (Allahverdizadeh et al., 2015)	26
Figure 4.12	Probability of failure evolution as a function of number of realizations for the investigated scales of fluctuation	27
Figure 4.13	Cumulative distribution function of safety factors for the current study along with deterministic solutions for mean, median and the deterministic solution of the original study (Allahverdizadeh et al., 2015)	27
Figure 4.14	Geometry of the slope from the original study tackled for the first verification (Chok et al., 2015). H = general problem dimension [m]	29
Figure 4.15	Probabilities of failure as a function of scales of fluctuation and coefficient of variation for properties in the original study (Chok, 2015)	29
Figure 4.16	Probability of failure evolution as a function of number of realizations in the original study (Chok, 2015)	30
Figure 4.17	Comparison of probabilities of failure between this study and the reference study (Chok, 2015)	30
Figure 4.18	Probability of failure evolution as a function of number of realizations for the investigated scales of fluctuation	31
Figure 4.19	Cumulative distribution function of safety factors for the current study along with deterministic solutions for mean, median and the deterministic solution of the original study (Chok, 2015)	32
Figure 5.1	Geometry of the validation case study - the Starnmeer dyke (Hicks et al., 2019)	33
Figure 5.2	Results of the case study in the form of cumulative distribution functions for the safety factor as a function of spatial variability (Hicks et al., 2019)	34
Figure 5.3	Finite element mesh generated for the analysis of the validation case study	35
Figure 5.4	Illustration of the random field for c' [kPa] for $\theta_v = \theta_h = 0.5 m$ in PLAXIS	36
Figure 5.5	Illustration of the random field for ϕ [°] for $\theta_v = \theta_h = 0.5 m$ in PLAXIS	36
Figure 5.6	Cumulative distribution function of safety factors for the current study compared to the original study (Hicks et al., 2019)	37
Figure 6.1	Views of the site of the foundation analyzed in this problem (Google, 2022)	41
Figure 6.2	Mesh with output extraction points for the demonstrative case study	42
Figure 6.3	Zoom in view on stress point (purple) spacing at a remote area in the mesh	43
Figure 6.4	Slope failure probability as a function of number of realizations	44
Figure 6.5	CDF of tilt [°] comparison for $\theta_h = 2m$ and $\theta_h = 10m$ (both with $\theta_v = 2 m$	45
Figure 6.6	Tilt P_f as a function of number of realizations for the serviceability analysis	46
Figure 7.1	Recipe for systematic use of RFEM in PLAXIS	47
Figure E.1	3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 2 m, \theta_y = 2 m$	83
Figure E.2	3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 10 m, \theta_y = 10 m$	83

Figure E.3	Parameter Markovian correlation (ρ) as a function of an isotropic spatial correlation structure, $\theta_y = 2\text{ m}$ $\theta_x = 2\text{ m}$	84
Figure E.4	Parameter Markovian correlation (ρ) as a function of an isotropic spatial correlation structure, $\theta_y = 10\text{ m}$ $\theta_x = 10\text{ m}$	84
Figure F.1	Cumulative distribution functions and deterministic solutions for safety factors in the 2 nd verification case study	85
Figure G.1	Random field for c' [kPa] - Verification case 2 - $\Theta = 0.1$ ($\theta_v = \theta_h = 1\text{ m}$)	87
Figure G.2	Random field for c' [kPa] - Verification case 2 - $\Theta = 0.5$ ($\theta_v = \theta_h = 5\text{ m}$)	87
Figure G.3	Random field for c' [kPa] - Verification case 2 - $\Theta = 1$, ($\theta_v = \theta_h = 10\text{ m}$)	88
Figure G.4	Random field for c' [kPa] - Verification case 2 - $\Theta = 5$, ($\theta_v = \theta_h = 50\text{ m}$)	88
Figure G.5	Random field for c' [kPa] - Verification case 2 - $\Theta \rightarrow \infty$	88
Figure H.1	Spacing between integration points in the sand layer - Starnmeer validation case (Red points = nodes; Purple points = stress points)	89
Figure I.1	Starnmeer validation case - random field for c' [kPa], $\theta_v = 0.5\text{ m}$, $\theta_h = 6\text{ m}$	91
Figure I.2	Starnmeer validation case - random field for ϕ' [°], $\theta_v = 0.5\text{ m}$, $\theta_h = 6\text{ m}$	91
Figure I.3	Starnmeer validation case - random field for c' [kPa], $\theta_v = \theta_h \rightarrow \infty$	92
Figure I.4	Starnmeer validation case - random field for ϕ' [°], $\theta_v = \theta_h \rightarrow \infty$	92
Figure J.1	Comparison of CDFs of safety factors for the Starnmeer validation case: $\theta_v = \theta_h = 0.5\text{ m}$	93
Figure J.2	Comparison of CDFs of safety factors for the Starnmeer validation case: $\theta_v = 0.5\text{ m}$ $\theta_h = 6\text{ m}$	93
Figure J.3	Comparison of CDFs of safety factors for the Starnmeer validation case: $\theta_v = \theta_h \rightarrow \infty$	94
Figure K.1	Demonstrative case study - Illustration of incremental displacements $ \Delta u $ from the Safety analysis in PLAXIS	95
Figure L.1	Demonstrative case study - random field for G [kPa], $\theta_v = \theta_h = 2\text{ m}$	97
Figure L.2	Demonstrative case study - random field for G [kPa], $\theta_v = 2\text{ m}$, $\theta_h = 10\text{ m}$	97
Figure L.3	Demonstrative case study - random field for G [kPa], $\theta_v = \theta_h \rightarrow \infty$	98
Figure M.1	Illustration of average calculation times in case studies treated in this thesis	99

LIST OF TABLES

Table 3.1	Inputs for a cohesion random field for the mesh size influence illustration	17
Table 4.1	Inputs for gradient field simple verification	19
Table 4.2	Inputs for random field simple verification - uncorrelated fields	20
Table 4.3	Inputs for random field simple verification - cross-correlated fields	22
Table 4.4	Inputs for the first verification (Allahverdizadeh et al., 2015) .	24
Table 4.5	Inputs for the second verification (Chok, 2015)	29
Table 5.1	Original study parameter inputs (Hicks et al., 2019)	33
Table 5.2	Adapted parameter inputs for this study	34
Table 5.3	Comparison of probabilities of failure between the current and the original study	37
Table 6.1	Stochastic inputs for soil properties in the analysis of foundation serviceability	42
Table 6.2	Inputs for plate properties in the analysis of foundation serviceability	42
Table 6.3	Scenarios of analysis as a function of θ	42
Table 7.1	Computation times as a function of random parameters in the final case study	50
Table M.1	Average computation times for all the cases studied in the thesis	99

SYMBOLS AND NOTATIONS

β = Reliability index [-]

c' = Cohesion [kPa]

D_f = Failure region

$E[\mathbf{Z}]$ = Expected (mean) value of a random field

ϕ = Friction angle [°]

Φ = Standard normal cumulative distribution function

γ = Volumetric weight [kN/m^3]

g = Performance function

G = Shear modulus [Pa]

L = Cholesky factorization matrix

μ = Mean value of a data set

N = Normal force [kN]

Q = Shear force [kN]

P_f - Probability of failure [-]

$P_{f,Tilt}$ - Probability of failure in exceeding the tilt serviceability criterion [-]

ρ = Spatial correlation function

R = Cross-correlation matrix

$R(X)$ = Resistant actors in a system

$S(X)$ = Soliciting actors on a system

σ = Standard deviation of a data set

τ = Lag between two points in correlation theory [m]

θ = Scale of fluctuation [m]

Θ = Scale of fluctuation normalized to a problem dimension [-]

V = Coefficient of variation [-]

X = Random variable

ξ = Cross-correlated random field

Z = Random field

ABBREVIATIONS

CDF – Cumulative Distribution Function

CEN – European Committee for Standardization

COV - Coefficient of Variation

CPT – Cone Penetration Test

DLL - Dynamic Link Library

EC7 - Eurocode 7: Geotechnical Design

FEM – Finite Element Method

FOS (or SF) – Factor of Safety

JPD – Joint Probability Density function

LAS - Local Average Subdivision

LEPP – MC – Linear Elastic Perfectly Plastic – Mohr Coulomb

MC – Mohr Coulomb

MCS – Monte Carlo Simulation

PDF – Probability Density Function

RFEM – Random Finite Element Method

SLS - Serviceability Limit State

UDSM – User Defined Soil Model

ULS - Ultimate Limit State

1

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The ground represents one of the most highly variable materials which engineers have to deal with. As opposed to steel, wood or concrete, whose characteristics are widely known and vary much less within better known ranges, designers in geotechnical engineering need to face much greater uncertainties from site to site, or even within the same site location (Fenton et al. 2016b).

Heterogeneity represents the spatial variability of properties of soil at various scales. This starts at the level of particles – grains or the fibrous nature of organic soils. Zooming out to the centimeter to meter scale this illustrates the variability of soil properties within what is denominated as a “uniform” soil layer. At an even larger scale, this could be seen from a geological perspective as the layering of soils themselves, as defined by sedimentological processes at very large time scales.

The variability this study focuses on is the intermediate one (centimeter to meter scale). This heterogeneity can decisively influence the behaviour in the soil-structure interaction in geotechnical projects. It also results in uncertainty in ground conditions, which in turn determines uncertainty in design (Arnold, 2012; Hicks, 2007). An example of how this can influence operational parameters in the ground can be observed in figure 1.1 (Yetbarek et al. 2020) in which the variability of hydraulic conductivity in a 3D soil block is portrayed.

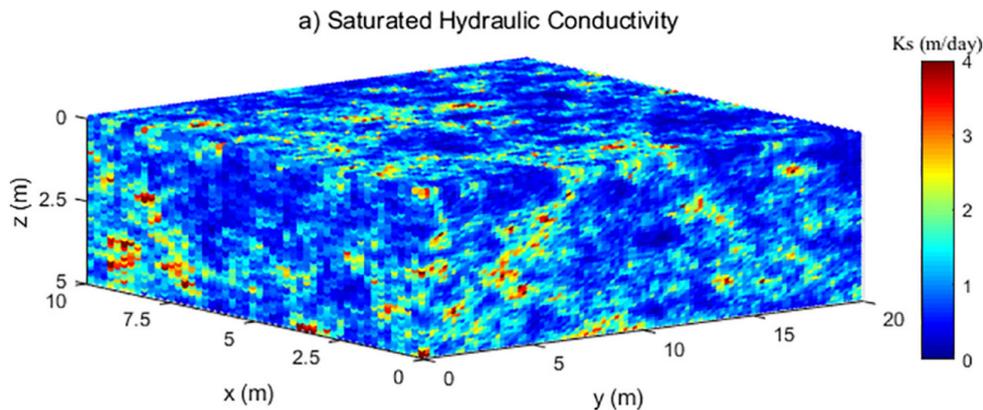


Figure 1.1: Example of spatial variability of hydraulic conductivity in 3D (Yetbarek, 2020)

Full quantification of spatial variability through direct observation is impossible due to the inaccessible nature of the underground and even partial determination of this requires extensive field and laboratory testing (Jaksa et al. 1999; de Gast et al. 2017), with great implications on the costs of preparation of a project. As a result, problems are still conventionally solved analytically, in a deterministic manner, based on a unique combination of soil parameters for each layer, ignoring soil heterogeneity. The usual result of a deterministic calculation in the ultimate limit state is a factor of safety (SF), which does not reveal the risks associated with the project in a comprehensive way, providing only one data point in the risk picture.

A widely used numerical method developed for calculations in engineering is the finite element method (FEM). This can be defined as an approach using the approximation of continuum problems in which the continuum is divided in a finite number of elements, with associated nodes and integration points, which are fully characterized by a set of readily understandable properties. Eventually, the set of equations is reassembled into a matrix system and solved (Zienkiewicz et al., 2013). In geotechnical engineering, this is mostly employed in deterministic calculations.

PLAXIS is a finite element package, developed for the analysis of deformation, stability and groundwater flow in geotechnical engineering. This was created at the TU Delft in the late 1980s as an initiative from the Dutch Ministry of Public Works and Water Management. It was intended to provide a tool for analysis to be used by geotechnical engineers in practice, when manual computation would be too cumbersome and time-consuming. An example of a finite element mesh created for a geotechnical project is presented in figure 1.2 (Bentley Systems, 2022).

Example of a finite element mesh in PLAXIS 2D

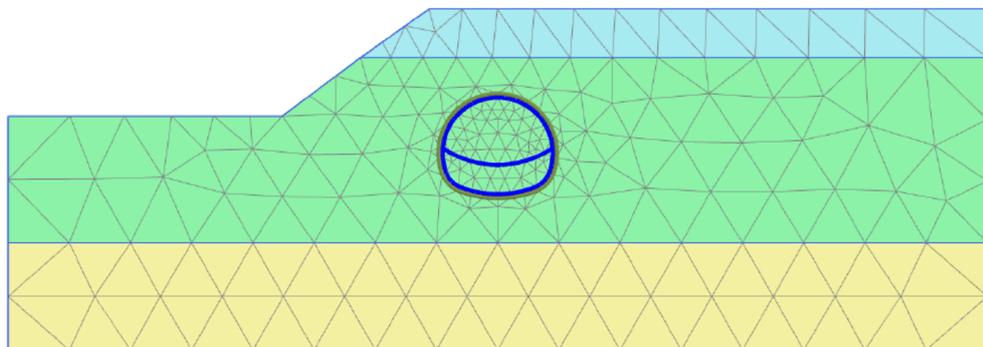


Figure 1.2: Example Finite Element Mesh from PLAXIS 2D (Bentley Systems, 2022)

When using the finite element method for soil calculations, constitutive models describe the relationship between cause and effect. More specifically, these are non-linear relationships between stress rates and strain rates. Besides the constitutive models that have already been programmed within the software and that come packaged with it, PLAXIS allows capable users to create their own material models and implement them. This is facilitated through the User Defined Soil Model (UDSM) interface of the software, which allows users to program the constitutive relationships separately and import them into the software to replicate the behaviour of their material of interest as accurately as possible.

Since PLAXIS' calculation kernel is not accessible, an alternative needs to be found in order to allow for the assignment of different properties at each stress point in the problem. The UDSM facility can be a solution, in which the existing variables may be utilized creatively to read and store property fields at stress point locations.

Currently, to achieve acceptable levels of safety, codes of engineering conduct like the Eurocode 7 (CEN, 2004) require the application of partial or global safety factors to apply a margin of error accounting for the risk of property variability. That being said, the evolution of geotechnical design codes has been lagging behind the structural codes concerning the transition away from working stress design (use of factors of safety). Because of the inherent calculation shortcoming imposed by undetectable variability in the ground, a real desire in the geotechnical engineering community exists for holistic site understanding of material ambiguity. This is likely to lead to safer and more economical designs (Fenton et al. 2016a).

To address this shortcoming, extensive research has been performed to quantify spatial variability from limited data (e.g., Vanmarcke, 1977; Phoon, Kulhawy, 1999; Lloret-Cabot et al., 2014; Fenton et al., 2018; de Gast et al., 2019).

Researchers have developed a significant number of semi-analytical and computational methods. However, for the purpose of this thesis, the computational approach focused upon is the Random Finite Element Method (RFEM) – a comprehensive procedure to capture spatial variability in geotechnical projects. This implies the modelling of every point in the ground (the stress points in a finite element model) as random variables. This collection of distributions of properties (joint distribution) represents the random field (Fenton, Griffiths, 2008). Simulating a number of combinations of soil parameters at stress points within realistic ranges through Monte Carlo Simulations (MCS), the performance of a system can be expressed in probabilistic terms, rather than as a single factor of safety (Hicks, 2014).

Various scholars demonstrated that accounting for this spatial variability using RFEM can lead to safe, more economical designs, all while simultaneously identifying the potential failure mechanisms more precisely (van den Eijnden et al. 2018, Dyson et al. 2019, Hicks et al. 2014).

Even though fundamental research is present in scientific literature, implementation in practice is still very low. This is especially true for RFEM, which for all its merits, could prove computationally expensive, in part due to not making any preliminary assumptions with respect to where failure may occur.

A major upside this research carries consists in the bevy of possibilities it unlocks. In implementing the spatial variability of properties in a thoroughly coded commercial software, the types of problems that could be analyzed using the concept of RFEM increase significantly. Up to this point, mostly ultimate limit state reliability studies were performed on relatively simple geometries and boundary conditions. A successful undertaking in this research unlocks the use of RFEM with features already existent in PLAXIS. Subsequent analyses could be done accounting for spatial variability in staged construction, performing serviceability analyses or even flow problems upon further expansions of the implementation of the concept.

While scientific innovation is desirable in and of itself, its ultimate goal is to improve society, which could only be done through large-scale application of its findings. Observing the tools available in the academic medium, the motivation behind this research becomes obvious: bridging the gap between science and practice. This is done through the implementation of the random finite element method in PLAXIS, leading to the consideration of soil heterogeneity in models as common practice.

1.2 OBJECTIVE AND RESEARCH QUESTIONS

Having introduced the relevant paradigm encircling the study, the explicit objective of the research is subsequently described in this sub-chapter. This is formulated more specifically as a set of research question and sub-questions:

1. **How can spatial variability be implemented in PLAXIS?**
 - a) How can a random field be created, stored and made available to PLAXIS' calculation kernel for the stress point properties?
 - b) How can results (probabilities of failure, safety factors, deformations etc.) for multiple different random fields be calculated automatically?
2. **How do reliability analysis results with the implemented concept compare to results obtained via the existing RFEM codes?**
3. **How can the created tool be utilized to perform more complex geotechnical analyses, while taking into account the spatial variability of soil properties?**

The first research question is formulated to encapsulate the core objective of the research project – the creation of a tool which allows for the accounting of spatial variability of soil properties in the PLAXIS FEM package. The sub-questions focus on the procedural steps which require formulation in this endeavour.

The second research question concentrates on the verification and validation of the implementation of the random finite element method in PLAXIS on cases which were previously calculated with existing RFEM research codes.

The third and last research question concerns the expansion of the concept to analyse a complex geotechnical problem with the developed tool, which could not be solved with the RFEM codes previously developed due to an elevated degree of complexity. This however could be tackled with the combination of the RFEM concept and PLAXIS' existing capabilities to solve more intricate problems involving for instance soil-structure interaction or hydro-mechanical coupling.

To attain the stated goals, the document is structured as follows: First, the relevant background literature treated is presented in chapter 2. Then, the research and elaboration methodology is explained in chapter 3. Next, a verification of the concept implementation is presented in chapter 4. This is followed by a validation case study (chapter 5) in which the results obtained with the current tool are compared with results of a complex problem treated in literature.

After this, another case study which was not previously performed with this method is presented in chapter 6, in order to illustrate the new capabilities this implementation unlocks. Chapter 7, comprising insights on good practices and the operational impact of influential factors of the model summarizes the practical experience obtained by the author in performing the research.

Finally, the document is concluded with a discussion (chapter 8) on the nuances of this research and a set of conclusions and recommendations (chapter 9) for effective utilization of the framework and further possible research lines.

2 | LITERATURE REVIEW

This chapter highlights the relevant literature treated, which provides a basis for attaining the objectives of the research. The section is divided in two sub-chapters, corresponding to the two main theoretical building blocks of this study - the finite element method structure and the theory of random fields.

2.1 THE FINITE ELEMENT METHOD STRUCTURE

A widely used numerical method developed to solve boundary value problems, mostly used deterministically in practice is the finite element method (FEM). The process can be defined as a method of approximation of continuum problems in which the continuum is divided in a finite number of elements characterized by a set of properties readily understandable. Eventually, the set of equations is reassembled into a system and solved. The core steps involved in the finite element method are summarized below (Zienkiewicz et al., 2013):

Illustration of the assembly of elements in the Finite Element Method

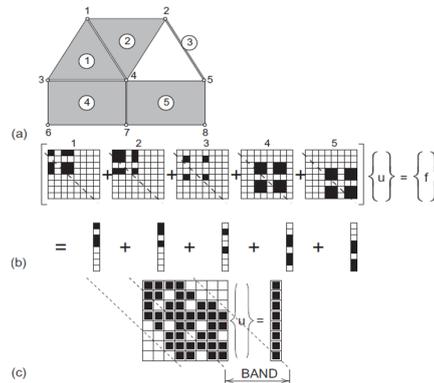


Figure 2.1: FEM assembly structure (Zienkiewicz et al., 2013)

- Step 1. Determination of element properties from the material and loading data, along with the element stiffness matrices creation.
- Step 2. Assembly of the final equations by addition of element matrices in the global matrix. The first two steps are illustrated in figure 2.1.
- Step 3. Insertion of prescribed boundary conditions in the assembled matrix
- Step 4. Solving the equation system to obtain desired variables such as stresses, currents, fluxes, displacements etc..

Computations for geotechnical problems involving soils are done using (geomaterial) constitutive models. In the most basic description, these represent (potentially non-linear) relationships between stress rates and strain rates (Brinkgreve, 1994). For this initial implementation, the Linear Elastic Perfectly Plastic Mohr-Coulomb (LEPP – MC) model is modified to account for spatial variability of soil properties.

2.2 RANDOM FIELDS AND RFEM

In the most essential way, a random variable is used as a means to identify events and their probabilities in numerical terms. The primary motivation for using them is such that rules of conventional mathematics can be applied to quantify probabilities that are harder to grasp intuitively (Fenton & Griffiths, 2008).

To model spatially variable properties, it is necessary to allow every point in the ground to be characterized by one or more random variables. For the purposes of the study, it is more useful to consider a random field, which contains a property that can vary randomly over a given spatial or temporal domain. *"Random fields are collections of random variables, which are completely specified by the joint distribution between all their component random variables"* (Fenton & Griffiths, 2008).

In reliability analyses, regions are defined as a function of these random variables. A failure region is defined as $D_f = [\mathbf{X} | g(\mathbf{X}) \leq 0]$, where $g(\mathbf{X})$ represents the performance function, using the random variable \mathbf{X} encapsulating the soil parameters' distributions. This function is often expressed as a difference between the resisting forces "R" and the solicitation forces on the system "S" – $g(\mathbf{X}) = R(\mathbf{X}) - S(\mathbf{X})$. Using this definition and the joint probability density (JPD) function of the random variables " f_x " for x realisations of the random variable, the probability of failure can be quantified in a stochastic manner using equation (2.1) (Hicks, 2014).

$$P_f = P[g(\mathbf{X}) \leq 0] = \int_{g(\mathbf{X}) \leq 0} f_x(\mathbf{x}) dx_1 \dots dx_n = \text{Probability of failure} \quad (2.1)$$

Keeping in mind the definition of the probability of failure previously presented, another performance assessment variable can be introduced, in the form of the reliability index as per equation (2.2) (Fenton, Griffiths, 2008):

$$\beta = -\Phi^{-1}(P_f), \text{ where } \Phi \text{ is the standard normal cumulative distribution} \quad (2.2)$$

The Random Finite Element Method (RFEM) links random field theory to the finite element method (FEM). Its steps are described in what follows, ranging from the pre-processing all the way to post-processing.

In the pre-processing stage, a statistical characterization needs to be made, on the basis of laboratory and in-situ testing. This involves determining the point statistics – mean (μ) and standard deviation (σ) – of properties of interest on site. After this step, the spatial correlation distances need to be obtained. For the vertical direction, a preliminary assessment is needed to detect depth trends and remove these if they exist. On a de-trended profile, the vertical scale of fluctuation (θ_v) can be quantified with a certain degree of accuracy. Obtaining the horizontal scale of fluctuation is more difficult and the accuracy of its determination is directly proportional with the number and spacing of the data points available (de Gast, 2020).

Loosely speaking, this represents the distance in space after which two points can be considered essentially uncorrelated. Rigorously speaking, if (ρ) is the correlation function between points, the scale of fluctuation is defined in equation (2.3) as:

$$\theta = \int_{-\infty}^{\infty} \rho(\tau) d\tau, \text{ where } \tau \text{ is the distance between the points considered} \quad (2.3)$$

Numerically quantifiable, this represents a measure of distance within which two points have a significant enough correlation, i.e. $\rho \geq e^{-2} \approx 0.135$ as described by Fenton & Griffiths (2008). An illustration of the properties described up to this point can be observed in figure 2.2 (Samy, 2003). If the site investigation allows for it, the horizontal scale of fluctuation (θ_h) may also be inferred comparing closely-spaced in-situ tests, like cone penetration tests (CPT) (Hicks, 2014). A small scale of fluctuation would create a highly scattered environment over a domain, while a larger scale of fluctuation results in a smoother distribution as points are correlated between themselves over larger distances (e.g., Figure 2.3 – Samy, 2003).

Illustration of scale of fluctuation (θ) and point statistics' (μ, σ) meaning

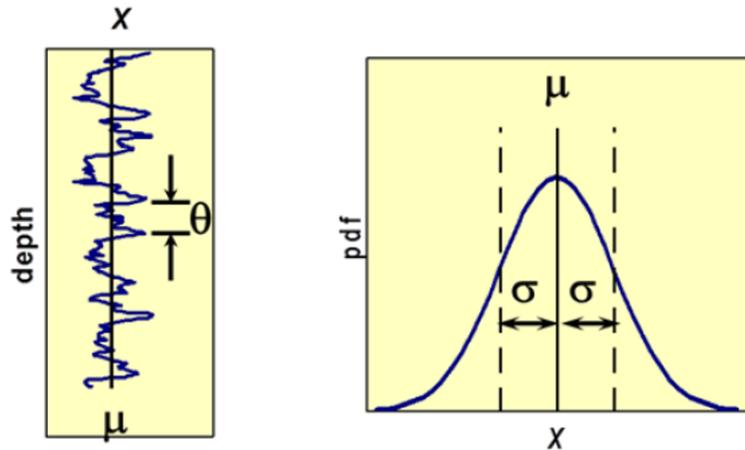


Figure 2.2: Vertical scale of fluctuation and point statistics of a random field (Samy, 2003)

Influence of θ on the appearance of a random field

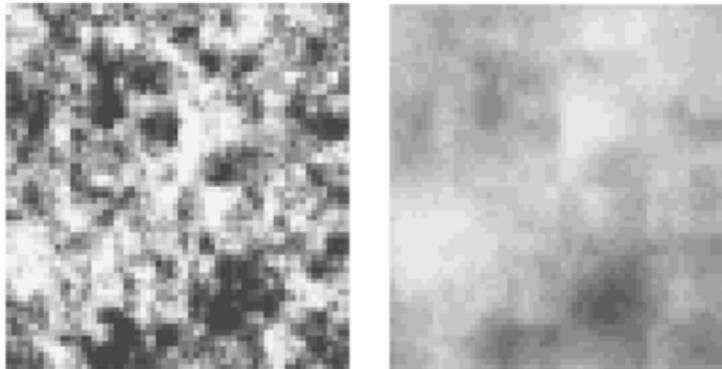


Figure 2.3: Influence of scale of fluctuation on random field appearance - the larger the value, the smoother the transitions between values (Samy, 2003)

In the analysis stage, for the given set of soil property statistics, a series of random fields is generated, for which the problem is analysed with FEM. One solution to the problem is denoted a realization, while the totality of realizations is called a Monte Carlo Simulation (MCS). For all realizations, the field will look similar due to having the same basis statistical properties, but each realization is unique in the combination of values utilized and in the results it yields.

Finally, after the Monte Carlo simulation, the stochastic results of the analysis are presented either as a probability density function (PDF) or cumulative distribution function (CDF) of output results.

A simple but robust parameter inferred is the probability of failure, as a conclusion of a binary quantification of failure versus non-failure (2.4):

$$P_f \approx \frac{1}{N} \sum_1^N I_i, \text{ where } I_i = 1 \text{ if failure criterion is exceeded and } I_i = 0 \text{ if not (2.4)}$$

It should be stressed that failure does not necessarily imply ultimate limit state attainment, or full collapse of a structure. This is a more complex concept dependent on the user's strictness in setting a performance criterion. Failure can be defined for serviceability as well, as the point at which set serviceability limits are exceeded. For instance, if it is deemed crucial for a structure to settle less than 1 cm, "failure" with respect to this criterion may occur without any obvious visual effect. That being said, a performance function can be defined in terms of any output parameter, be it factor of safety, displacement, stress, force etc.

In this context, the number of realizations needed for the simulation to reach a desired accuracy can be inferred using the coefficient of variation as a function of a (defined expectation of) probability of failure (2.5).

$$COV(P_f) \approx \sqrt{\frac{1 - P_f}{P_f(N - 1)}} \quad (2.5)$$

In this context, coefficient of variation (COV) associated to the probability of failure is the accuracy metric denoting the confidence interval of the solution obtained. For instance, a $COV = 0.05$ corresponds to a 95 % confidence in the solution, a target accuracy frequently used in practice due to its link to Eurocode statistical significance requirements (CEN, 2004). This is a function of the target P_f and the number of realizations "N". This formula is frequently used to infer an expectation regarding the number of realizations required to obtain a target probability of failure.

Knowing the methodology behind RFEM, it is now worth noting that this calculation procedure was proven to be effective in practice in many applications. Although theoretically applicable to any type of geotechnical project, often case studies focus on slope stability problems, which are more accessible from a geometrical and analytical complexity standpoint.

For instance, RFEM was used by Hicks and Onisiphorou (2005) to investigate the influence of heterogeneity of the state parameter (Been and Jefferies, 1985) on static liquefaction potential in a predominantly dilative sand fill. Hicks and Boughrarou (1998) researched the influence of the same parameter on the liquefaction potential for the underwater Nerlerk berm, demonstrating the possibility of a predominantly dilative fill to liquefy through failure along semi-continuous weak zones.

3 | METHODOLOGY

In this chapter, the methodology of implementation of the random finite element method in PLAXIS via UDSMs is described in detail. The chapter is divided in two sub-chapters, corresponding to the implementation methodology on one hand, and considerations on the influence of spatial correlation distances on finite element mesh choices on the other hand.

3.1 IMPLEMENTATION - RFEM IN UDSM

3.1.1 RFEM and random field generation

The procedural challenge of this research is represented by the coding and linking of the concept behind RFEM within the PLAXIS interface. To achieve this, it is worth exploring further the methodology behind RFEM, so that it becomes clear at which point this could be encompassed within PLAXIS. A flowchart summarizing the procedure behind RFEM is presented in figure 3.1 below.

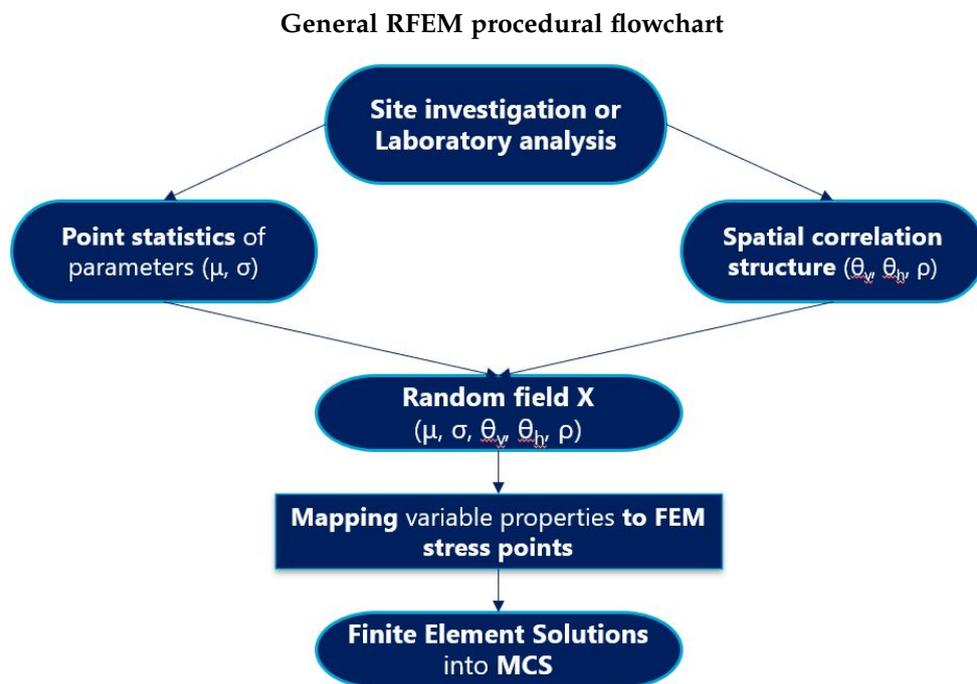


Figure 3.1: Original RFEM procedural steps

A phase that supposes a denser theoretical background is the creation of random fields. In what follows, the theory behind this step is explained in more detail.

The random field generation process starts at the definition of the point statistics (mean - μ , standard deviation - σ) for the parameters of interest – for instance the undrained shear strength s_u (or c' and ϕ' for 2 parameters etc.) – ideally from site

investigations. These should be accompanied by spatial variability property values i.e., scales of fluctuation, in the 2D case vertical and horizontal (θ_v, θ_h).

To ensure positive determinate values, log-normal distributions are often utilized for the parameters in question. To obtain these, the following transformations allow the creation of the fields using the correct mean and standard deviation (3.1):

$$\sigma_{LN} = \sqrt{\log 1 + \frac{\sigma^2}{\mu^2}} \quad \mu_{LN} = \log \mu - 0.5\sigma_{LN}^2 \quad (3.1)$$

There are many correlation functions defined in literature that could express the relationship between distance and correlation, such as the Gaussian, triangular, spherical or Markovian auto-correlation models (Fenton, 1999). The last one is commonly used to express spatial correlation for engineering purposes both due to its simplicity, as well as for an assumption it entails. This property affirms that the conditional probability of a future state of a parameter depends only on the most recently known state, which is usually valid for models, insofar as the historical evolution is not key to present behaviour.

For two points in positions t_1 and t_2 , let it be assumed that the distance between them is given by “ τ ”, where $\tau = t_1 - t_2$. Moreover, let it be assumed that at each of the two positions, a random variable of possible property values exists – $X(t_1), X(t_2)$. These are both characterized by separate probability distribution functions, as well as by a joint probability density function.

Considering these, the Markovian spatial correlation that exists between the two points, can be expressed and computed for “ m ” dimensions as in equation (3.2).

$$\rho(\tau) = e^{-|\tau|}, \quad |\tau| = \sqrt{\left(\frac{2\tau_1}{\theta_1}\right)^2 + \dots + \left(\frac{2\tau_m}{\theta_m}\right)^2} \quad (3.2)$$

After this step, if any known cross-correlation between parameters is known, it can be defined in matrix form. For “ n ” parameters, the cross-correlation matrix is expressed as in equations (3.3),(3.4):

$$\mathbf{R} = \begin{bmatrix} 1 & \rho_{Z_1 Z_2} & \dots & \rho_{Z_1 Z_n} \\ \rho_{Z_2 Z_1} & 1 & \dots & \rho_{Z_2 Z_n} \\ \dots & \dots & \dots & \dots \\ \rho_{Z_n Z_1} & \rho_{Z_n Z_2} & \dots & 1 \end{bmatrix} \quad (3.3)$$

Where the cross-correlations are computed knowing the expected (mean) values as:

$$\rho_{Z_i Z_j} = \frac{E[Z_i, Z_j] - E[Z_i]E[Z_j]}{\sigma_{Z_i}\sigma_{Z_j}} \quad (3.4)$$

After computing the Cholesky factorization \mathbf{L} of the \mathbf{R} matrix, having generated random fields from a standard normal distribution \mathbf{Z} , cross-correlated fields (ξ) can be obtained through the matrix multiplication operation: $\xi = \mathbf{LZ}$

Finally, to obtain the random fields for the underlying log-normal distribution, the transformation formulas in 3.5 and 3.6 (for cross-correlated fields) are used.

$$Z_{LN} = \exp[\sigma_{LN}Z + \mu_{LN}] \quad (3.5)$$

$$\xi_{LN} = \exp[\sigma_{LN}\xi + \mu_{LN}] \quad (3.6)$$

A summary of the procedure behind the generation of random fields is illustrated by the flowchart in figure 3.2 below.

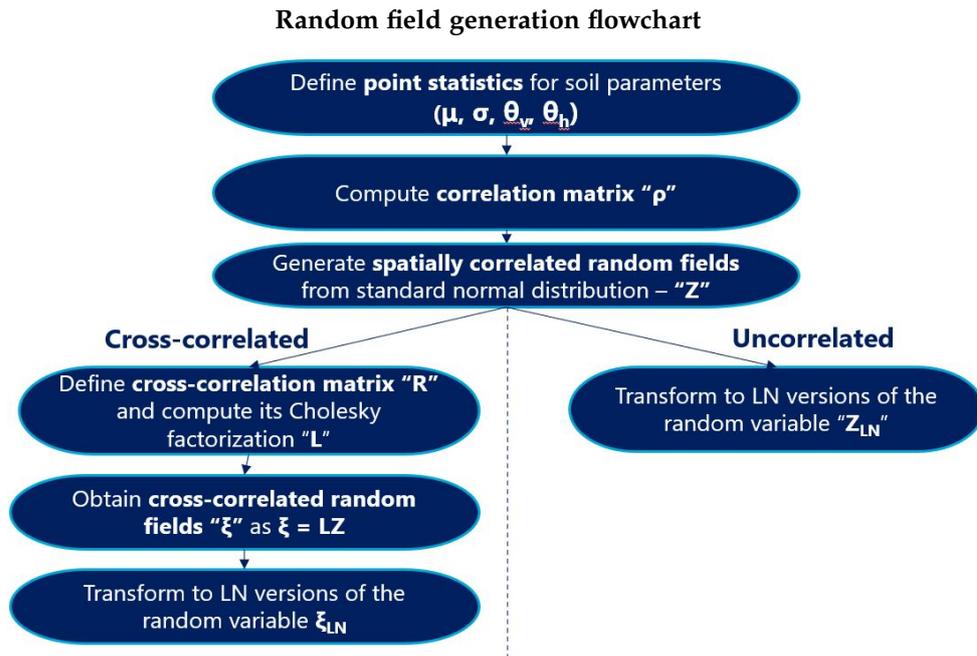


Figure 3.2: Flowchart for generation of random fields

The Python code used for the generation of the random fields in the script that dictates the Monte Carlo simulation around PLAXIS can be found in appendix A.

This implies the saving of random fields to a file and the total number of integration points to a separate file, which are to be read in the UDSM PLAXIS subroutine. This is explained in the following two sub-chapters which treat the adaptation of the example Mohr-Coulomb UDSM and the connection between this and RFEM.

The code in appendix A can be clipped to the one in appendix C to provide a full Python script that generates random fields and commands the Monte Carlo simulation for a simple slope geometry in PLAXIS. The two were originally one unit, but have been split so they can be referred to separately, as they correspond to different theoretical concepts of the elaboration presented.

3.1.2 PLAXIS UDSM Modifications

PLAXIS includes a facility for user-defined soil modelling (UDSM). This allows users to implement constitutive models in PLAXIS by programming them in FORTRAN and compiling them as Dynamic Link Libraries (DLLs), to be added in PLAXIS' repertoire. Six tasks are required for the creation of a UDSM, which are described in detail in the reference manual of PLAXIS (Bentley Systems, 2022).

For the scope of this study, only steps 1, 2 and 4 of the example UDSM for MC need to be modified. The tasks treat three different conceptual steps in development:

- Task 1: Initialization of "state" parameters
- Task 2: Assignment of stress point model parameters
- Task 4: Defining number of "state" parameters

A side-by-side comparison of the original coding of the reading Mohr-Coulomb model steps versus the modified UDSM meant to replicate the RFEM principle can be observed in the flowchart in figure 3.3 below.

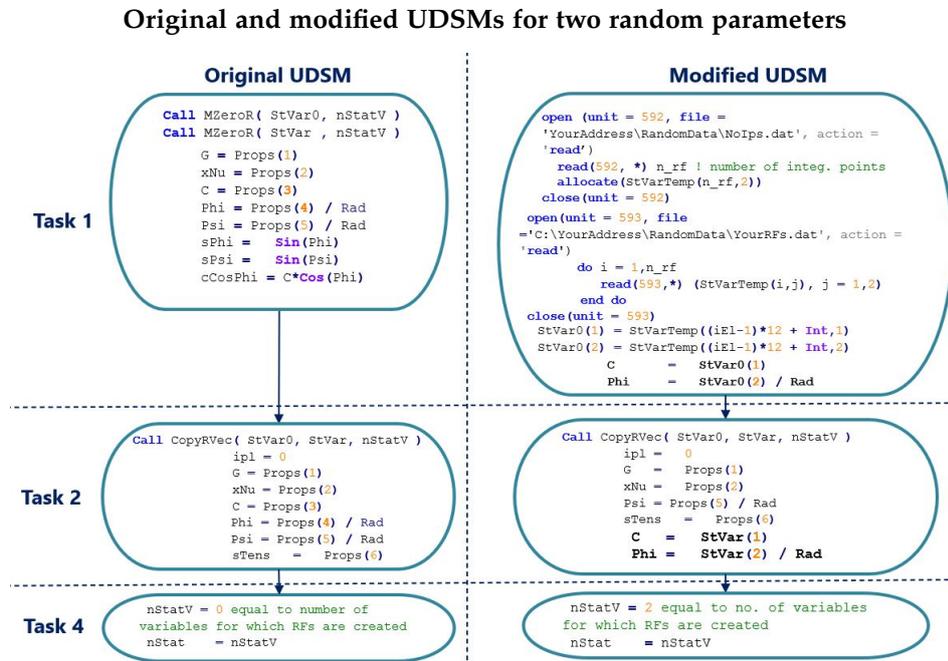


Figure 3.3: Comparison between the formulation of the UDSM in this study for two random variables and the original MC example UDSM (Bentley Systems, 2022)

The initialization phase of state variables in the UDSM interface is utilized to add the random fields in PLAXIS. In the variable set aside for this purpose ("StVar"), an unlimited number of parameters can be added. This makes it attractive to use for the storage of the random field values at integration points.

Originally, the deterministic values of the properties are stored in a "Props" variable, which accesses the input from PLAXIS' user interface. In this case, the user does not need to input any values in the interface, as these will be read from the random field generated in Python and saved in the path coded into the UDSM.

Besides the reading, one can use the UDSM variables for the local integration point "Int" and the element number "iEl" to code the path to the correct global integration point. The corresponding integration point utilized in the calculation step can be expressed as a function of these two variables as in equation (3.7):

$$global_{ip} = (i_{element} - 1) * 12 + local_{ip} \quad (3.7)$$

This expression is valid for 15-noded triangular elements, containing 12 integration points. For 6-noded triangular elements, this number equivalent to the number of integration points would change from "12" to "3".

The concept could be expanded even further to treat more of the properties carried by the "Props" vector as random variables. This implies the expansion of the temporary matrix from which state variables are attributed at each stress point. In theory, this could be done for all the variables involved in this code. However, with each parameter added, the computation times increase non-linearly. As a result, this should be done only if it is deemed that the variability of a particular property may have significant impact on the analysis.

In the final stage of this research, the initial implementation is expanded to take into account a stiffness parameter. For this, the shear modulus (G) used in the example Mohr-Coulomb UDSM is also treated as a random variable. This is useful for serviceability analysis, in which strength is not fully mobilized and stiffness dictates the behaviour of displacements in the problem. The elaboration, similar to the one in figure 3.3, is showcased in figure 3.4 below.

Original and modified UDSMs for three random parameters

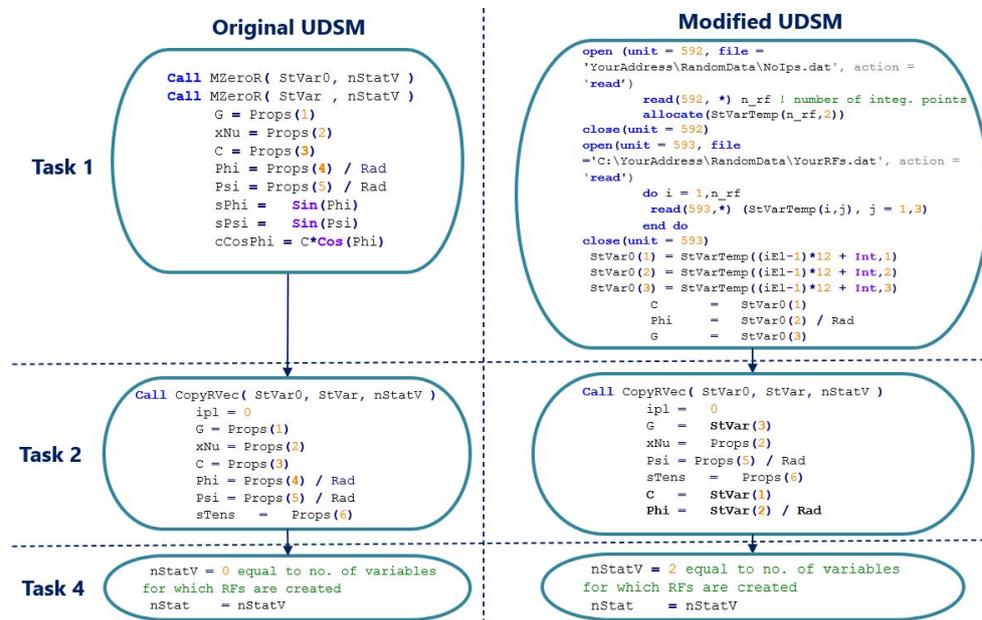


Figure 3.4: Comparison between the formulation of the UDSM in this study for three random variables and the original MC example UDSM (Bentley Systems, 2022)

The code with the key steps in the modified subroutine is shown in appendix B.

3.1.3 Connection RFEM-UDSM

To perform the link between the two frameworks, a common denominator needs to exist. This is encountered, as mentioned, in the stress points to which properties can be attributed and where stress-strain calculations are performed. Random fields are first created in a scripting program (Python) on the basis of stress points' geometry extracted from PLAXIS, soil parameters' point statistics and the spatial correlation structure as explained in sub-chapter 3.1.1. These are then stored in text files in PLAXIS' project folder, as explained in the previous section 3.1.2.

Multiple solutions to the problem are packaged in a Monte-Carlo Simulation by looping over the execution of the project with newly generated properties at every loop iteration. In this way, a stochastic solution is obtained.

Before running the full Monte-Carlo simulation, it is advisable to test the model with one realization. In doing this, a user can check whether or not the random field generated was read and represented accurately by PLAXIS with the selected mesh. Furthermore, by quantifying the computation time of this one realization, an expectation can be created concerning the duration of the full Monte-Carlo loop. The number of realizations defined for one Monte-Carlo run would likely be different if a realization takes 20 seconds, as opposed to 2 minutes.

A summary of the procedure described in this chapter, along with the process of evaluation of a problem is highlighted in figure 3.5 below.

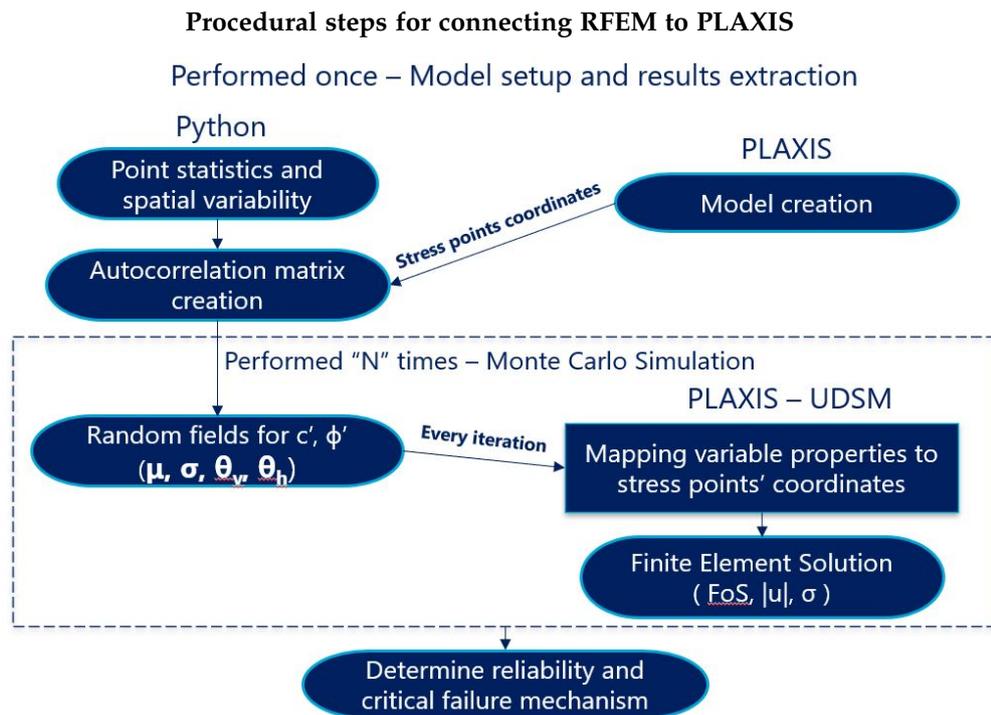


Figure 3.5: Flowchart highlighting the connection of RFEM to PLAXIS

The syntax utilized in the Python scripting interface to command this whole process around PLAXIS can be consulted in appendix C for the case of a simple slope with one soil layer. For more complex cases in which soil and random fields' divisions need to be performed, the procedure is elaborated in the next sub-chapter.

3.1.4 Full code generalization

With increasing complexity of problems, code optimization shifts from a desire to a necessity. As the number of soil layers expands, it becomes apparent that further automation of the code is needed. As a result, this sub-chapter describes a further enhancement of the tool such that users can fully command the creation and execution of a project directly from the Python script, provided a project's geometrical specifications are well-known and available to the user. The procedure is highlighted in the flowchart in figure 3.6 below.

Procedural steps for the automation of model evaluation with RFEM in PLAXIS

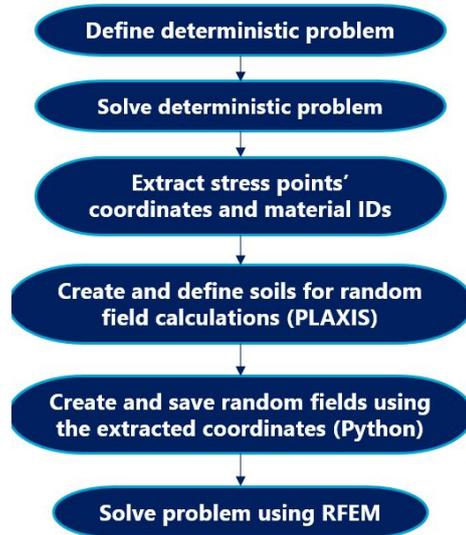


Figure 3.6: Procedural steps setting up the Python code automating RFEM in PLAXIS

The first step is the definition of the problem for a deterministic analysis. This implies the definition of soil layers using soil models already programmed in PLAXIS. Obtaining a deterministic solution is the next step, meaningful from more than one perspective. First, it provides a result that can be used as a benchmark for comparison with the RFEM solution. Even more importantly, after solving the problem, the script gets access to the output panel, allowing for the automatic extraction of coordinates through the scripting interface.

Therefore, after opening the output panel, one can extract the coordinates of the stress points in the soil layers, as well as their material IDs. The material IDs are numbers which correspond to the soil materials created previously, in the order of their creation. They can be used later for a separation of stress points coordinates on the basis of their belonging to a particular soil layer. The next step implies going back to the soil tab and automatically creating new soil materials selecting “UDSM” as a constitutive model and inputting the relevant non-random parameters to make the material valid. For instance, even though c' , ϕ' and G can and will be read from the text file as random fields, there may be other parameters that are relevant for the calculation, or without which the materials are not recognized as valid by the software. For the former category, these may involve Poisson's ratio (ν), or interface properties like c_{inter} for the latter.

At this point, the problem returns to the state previously elaborated in 3.1.3, in which users create random fields on the basis of coordinates of stress points belonging to different soils. These are then saved to a text file which will be accessed by the UDSM when the calculation begins. The last step is triggering the Monte-Carlo simulation as explained in the methodology and showcased in appendix C. The full code for this expanded elaboration can be found in appendix D.

3.2 SPATIAL CORRELATION INFLUENCE ON FINITE ELEMENT MESH CHOICES

In the final model, a close link exists between the spatial correlation distances inferred from the site investigation and the finite element mesh created by the user. The latter should be fine enough to properly capture the former, as indicated by the scale of fluctuation defined in equation (2.3), such that the inter-dependencies between values at different points are expressed appropriately.

For this, it is worth exploring how different scales of fluctuation influence the correlation between property values for a particular spatial correlation structure. There are many correlation functions that can be used to express the relationship between two properties. In this study, the multi-dimensional Markovian correlation function is used (3.2). This is chosen both due to its simplicity, as well as for an assumption it entails. The Markov property affirms that the conditional probability of a future state of a parameter depends only on the most recently known state. This is usually valid for engineering models, insofar as the historical evolution does not play a key role, as it could be the case with consolidation problems (Fenton & Griffiths, 2008).

In figure 3.7 below, an illustration of (Markovian) correlation as a function of distance in two dimensions is presented, for a case in which an anisotropic scale of fluctuation is considered. This means that the significant correlation distance between two points is not the same in the x and y directions. More specifically, in this case values of $\theta_x = 10 \text{ m}$ and $\theta_y = 2 \text{ m}$ were chosen for illustration purposes.

Markovian correlation function - $\Theta_x = 10 \text{ m}$, $\Theta_y = 2 \text{ m}$

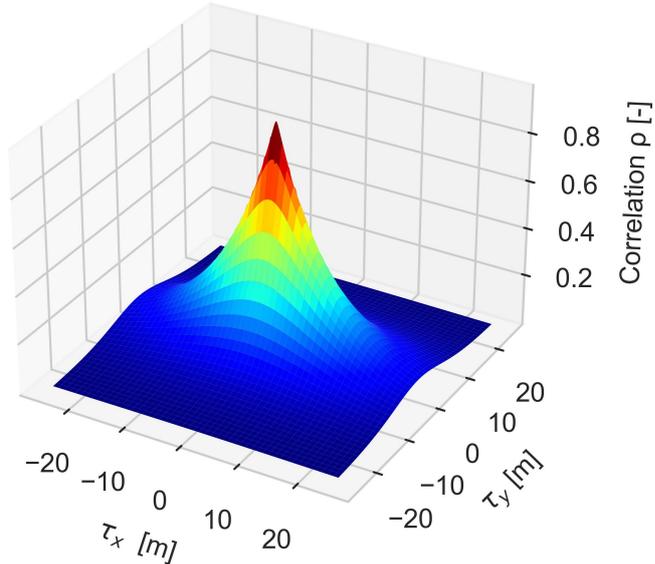


Figure 3.7: 3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 10 \text{ m}$, $\theta_y = 2 \text{ m}$. The closer points are to the origin, the larger the correlation, the warmer the color.

This figure highlights the influence that the scale of fluctuation has on the correlation between the property values at two different points. Inspecting the shape of the correlation structure, it can be observed that the same correlation (ρ) value is reached for a larger distance from the origin in the x-direction than in the y-direction due to the larger scale of fluctuation in the former.

To nuance this illustration regarding how the scale of fluctuation influences the correlation between properties, readers are referred to appendix E, showing comparative plots of isotropic combinations of θ for gradually increasing values.

This is meaningful for the choice of mesh in the model towards expressing the spatial correlation between points properly. Moving to 2D projections of the previous illustrations shown in figure 3.8 aids in fixing the concept previously highlighted.

In order to express a relevant correlation between two points, the value of ρ for that particular distance under the scales of fluctuation considered needs to be large enough. This relationship can be exemplified using figure 3.8a. For instance, if a user creates a model that only ensures a spacing between stress points of 10 m in the y-direction, the correlation between the two points considered is approaching zero (i.e, no correlation). This means that by choosing a mesh that is not fine enough, the information on spatial correlation could be lost, as this would not be expressed by the parameter distribution over the geometry in the model properly.

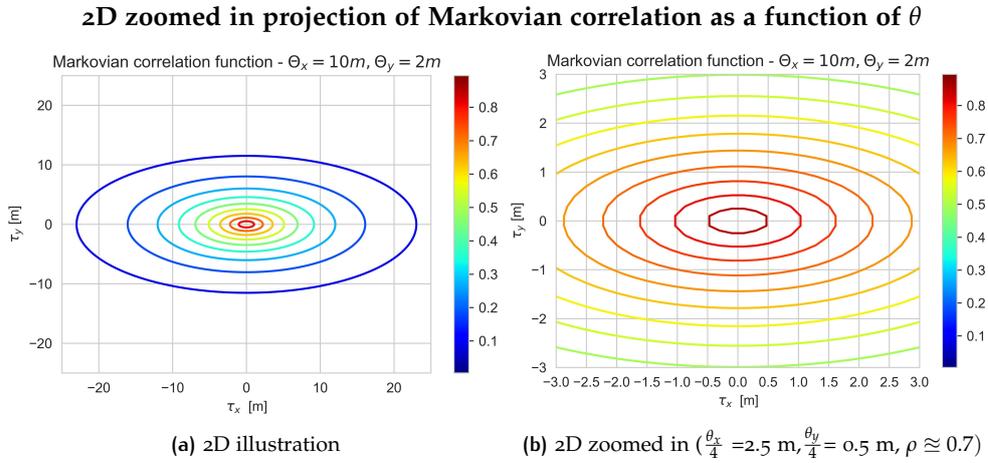


Figure 3.8: Parameter Markovian correlation (ρ) as a function of an anisotropic spatial correlation structure, $\theta_y = 2 \text{ m}$ $\theta_x = 10 \text{ m}$

As a result, to ensure a large enough correlation between points to capture the scales of fluctuation considered, at the very least the maximum distance between two stress points should not be larger than the scale of fluctuation considered. This is needed such that the spatial distribution inferred by practitioners from site investigation results in a model capturing the real distribution as closely as possible.

A useful thumb rule in the creation of the model is to aim for a spacing between stress points of approximately $\frac{\theta}{4} \text{ m}$, which means four stress points are distributed along a distance equal to the scale of fluctuation defined (Spencer, 2007). As it can be seen in figure 3.8b, if this thumb rule is employed, a correlation of $\rho \approx 0.6$ is achieved between the origin and the next closest point, which is significant enough to properly capture the potential creation of semi-continuous weak zones in the sub-surface. This is meaningful as the development of weak planes have been shown to have a significant impact on the performance of a system, allowing failure to propagate through slope geometry (Hicks, 2014).

To illustrate the impact that a poorly chosen mesh size could have on the representation of spatial variability in a model, readers are referred to figure 3.9. Generating the two cohesion random fields is based on parameters in table 3.1.

Table 3.1: Inputs for a cohesion random field for the mesh size influence illustration

Parameter	$\mu(c')$ [kPa]	$V(c')$ [-]	θ_v [m]	θ_h [m]
Value	10	0.5	0.5	3

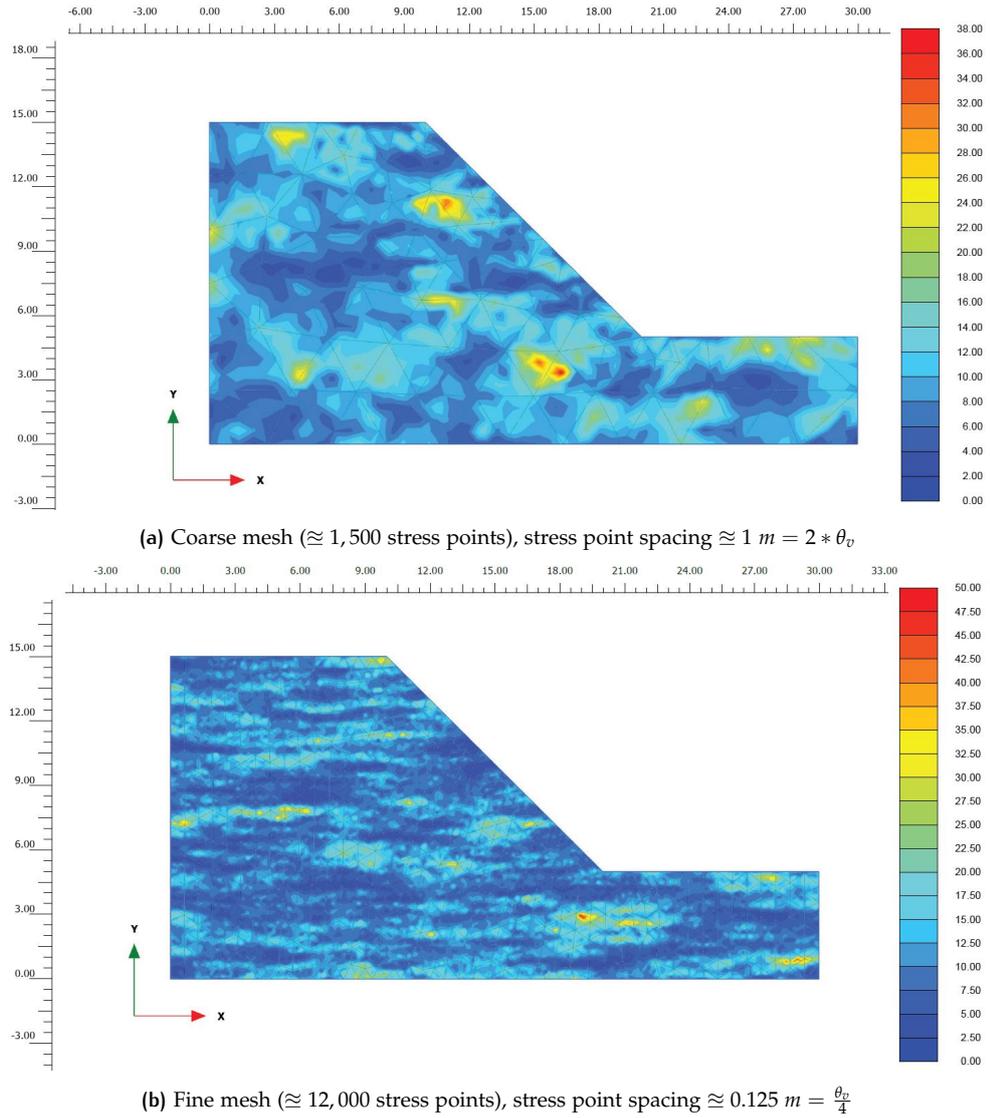
Random fields of cohesion - coarse mesh vs. fine mesh ($\theta_v = 0.5 \text{ m}$, $\theta_h = 3 \text{ m}$)


Figure 3.9: Influence of mesh size on spatial correlation representations using random fields of cohesion [kPa] and a combination of $\theta_y = 0.5 \text{ m}$ $\theta_x = 3 \text{ m}$

Using figure 3.9a as reference, it can be seen that for a stress point spacing larger than the chosen scale of fluctuation, the latter is not captured appropriately by the finite element model. For instance, in the vertical direction, the length of the continuous zones is larger than the scale of fluctuation input. For any point in the domain, there is no other point at a close enough distance which may express a spatially uncorrelated random value found at this second point.

Conversely, as can be seen in figure 3.9b, when choosing a mesh that ensures a fine enough distribution of points, the accurate representation of the random field is not problematic anymore. For instance, the vertical length of continuous zones resembles much closer the predefined value of scale of fluctuation of 0.5 m. That being said, in problems with extensive geometries, operational problems could occur if a very large number of stress points needs to be generated to fully capture the smallest scales of fluctuation all over the domain. As a result, in this respect users need to strike a balance between the accuracy of representation of the random field and the operational performance evaluated in computation times.

4 | VERIFICATION

This chapter illustrates the results of the verification of the tool. To establish whether or not the concept is properly implemented, two types of checks are performed. One concerns the verification of reading and storage of the random fields in PLAXIS. The second, more complex one, checks results against performance functions in two simple $c' - \phi$ slope stability problems on which RFEM was performed with existing codes. For these two latter cases, the differentiating factors are the geometry of the slope as well as the statistical distributions of strength properties (c' , ϕ').

4.1 SIMPLE VERIFICATION

The first and simplest verification is to perform a single realization of a problem with a well-defined, regular field and check whether or not PLAXIS reads and stores the random field correctly in the state variables matrix.

In this case, a gradient field (diagonal, in both the X and Y directions) of cohesion is created and plotted initially in Python. Then, commanding PLAXIS to use the UDSM for a single calculation, one can check with a plot of the corresponding state variable whether or not the fields have been read and stored in correctly. The equation and inputs used are highlighted in table 4.1 below.

Table 4.1: Inputs for gradient field simple verification

Parameter	Equation	A	B ₁	B ₂
Cohesion (c')	$c' = A + B_1 * X + B_2 * Y$	15	0.2	-1

The gradient field of cohesion is generated in Python and a simple gravity calculation on a one-layered slope was performed such that the random field input can be checked. In the output panel of PLAXIS, the State Parameters – “User Defined” parameter is verified, yielding the plot in figure 4.1. This is then compared with a similar plot generated independently in Python, that can be observed in figure 4.2. As one can observe in the two figures on the next page, the reading and storing of a random field for one parameter, in this case c' , and one soil is performed correctly.

Gradient fields of cohesion - verification of value reading in PLAXIS

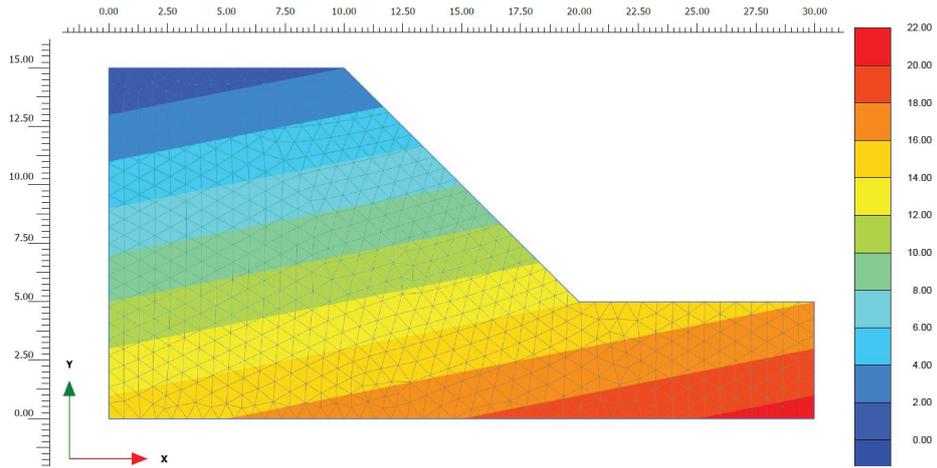


Figure 4.1: X-Y Gradient field visualization for cohesion [kPa] in PLAXIS

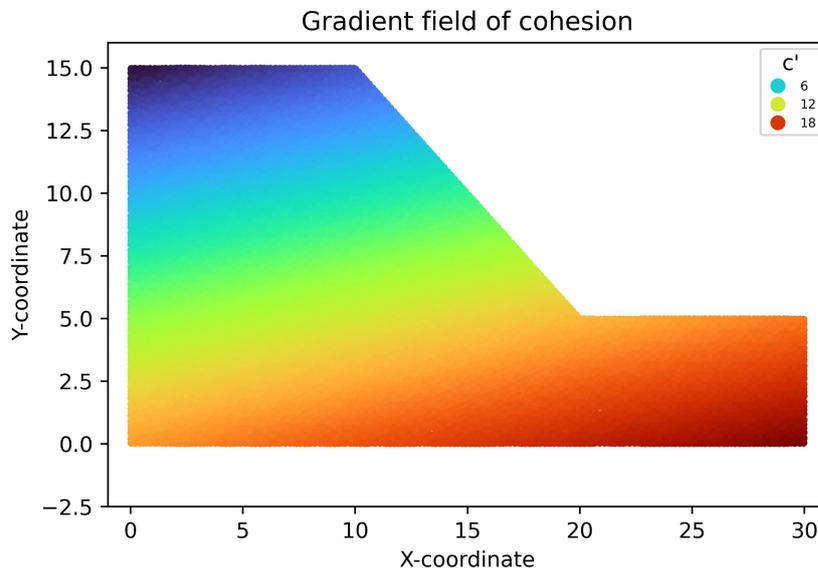


Figure 4.2: X-Y Gradient field visualization for cohesion [kPa] in Python

Then, to verify whether or not the spatial variability parameters are accounted for appropriately, another check is performed. This implies running the model with different scales of fluctuation for the vertical and horizontal directions. The input parameters used for this verification are highlighted in table 4.2 below.

Table 4.2: Inputs for random field simple verification - uncorrelated fields

Parameter	$\mu(c')$ [kPa]	$V(c')$ [-]	θ_v [m]	θ_h [m]
Value	10	0.5	1	15

The illustrations for the obtained distributions in Python and PLAXIS respectively are shown in figures 4.3 and 4.4 on the next page.

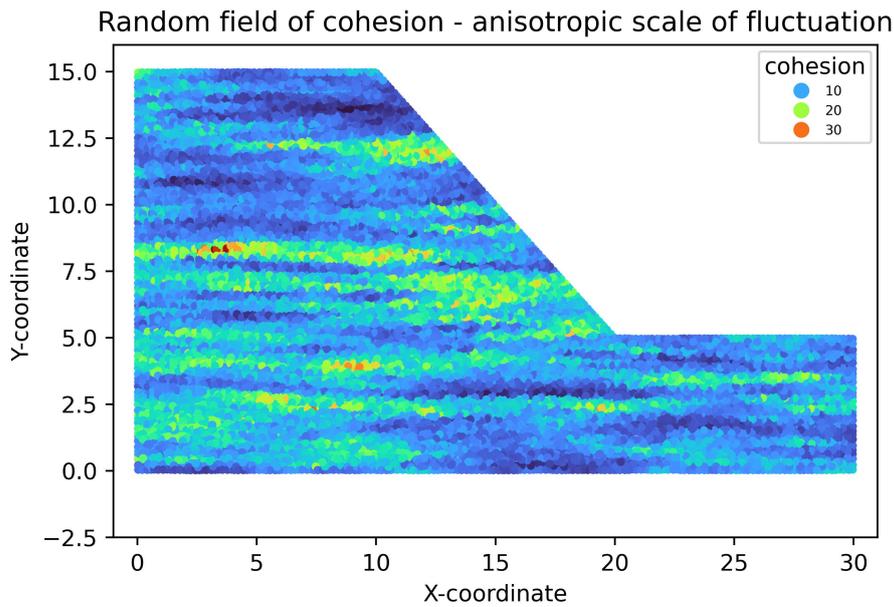


Figure 4.3: Random field distribution for cohesion [kPa] in Python - $\theta_v = 1 m$, $\theta_h = 15 m$

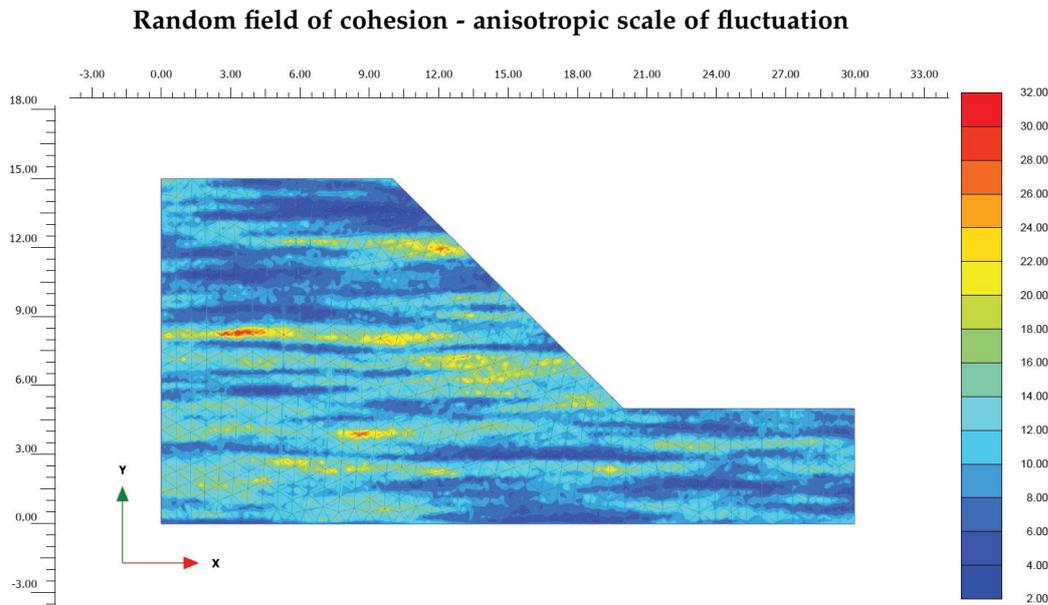


Figure 4.4: Random field distribution for cohesion [kPa] in PLAXIS - $\theta_v = 1 m$, $\theta_h = 15 m$

As it can be seen in the two figures above, the expectation that for a larger correlation horizontally, bands would start creating, starting to resemble a conventional layered distribution is confirmed. This serves as a good demonstration that the spatial variability is taken correctly into account.

Finally, for an increasing degree of complexity, with the same procedure one can create random fields, with cross correlations between the two parameters and verify that the variables created indeed showcase the trend dictated in the field generation. For instance, the $c' - \phi'$ fields can be generated with a negative correlation implying that the areas showcasing a high value of one variable, will be the exact ones where the other variable showcases its lowest values.

For this, two negatively correlated random fields are created for two parameters (c' and ϕ) with a correlation coefficient $\rho_{cc} = -0.99$. This way, the implementation of cross-correlated random fields along with their storing and reading in PLAXIS' calculation kernel are verified. The inputs for c' and ϕ' are listed in table 4.3.

Table 4.3: Inputs for random field simple verification - cross-correlated fields

Parameter	$\mu(c')$ [kPa]	$\mu(\phi')$ [°]	$V(c')$ [-]	$V(\phi)$ [-]	θ_v [m]	θ_h [m]
Value	10	30	0.5	0.25	1	15

In figures 4.5 to 4.8, readers can observe plots of negatively correlated c' and ϕ' random fields obtained separately in Python and PLAXIS' output panel respectively, for scales of fluctuation of one meter vertically and 15 meters horizontally.

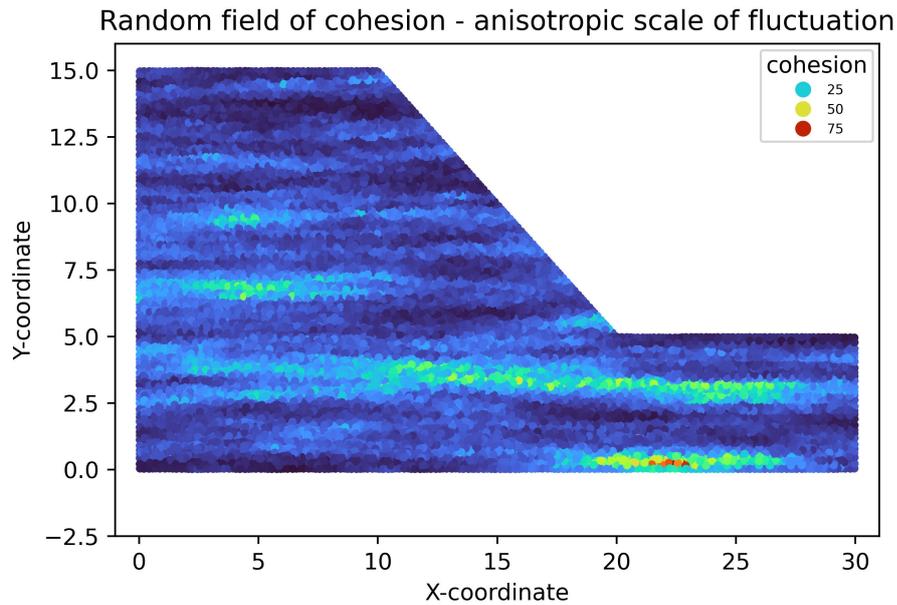


Figure 4.5: Random field distribution for cohesion [kPa] in Python - $\theta_v = 1$, $\theta_h = 15$ m

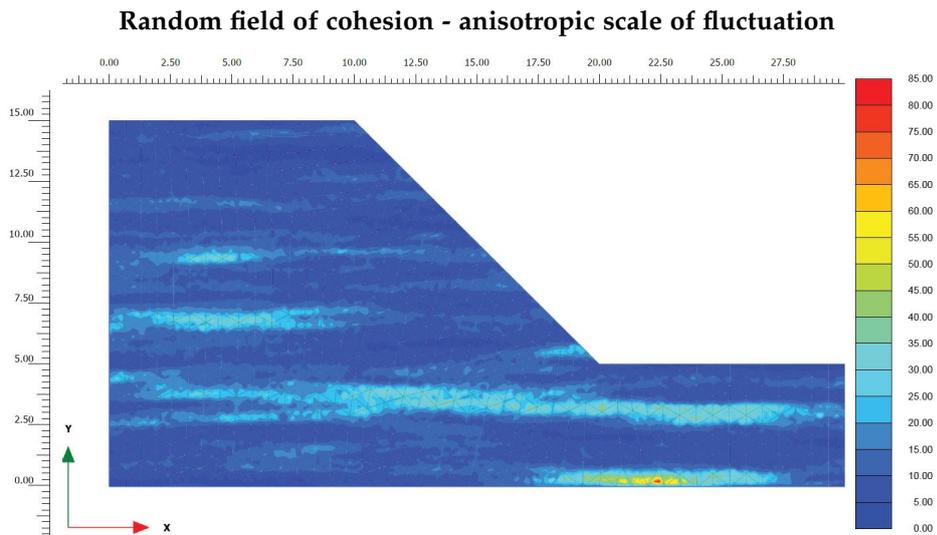


Figure 4.6: Random field distribution for cohesion [kPa] in PLAXIS - $\theta_v = 1$, $\theta_h = 15$ m

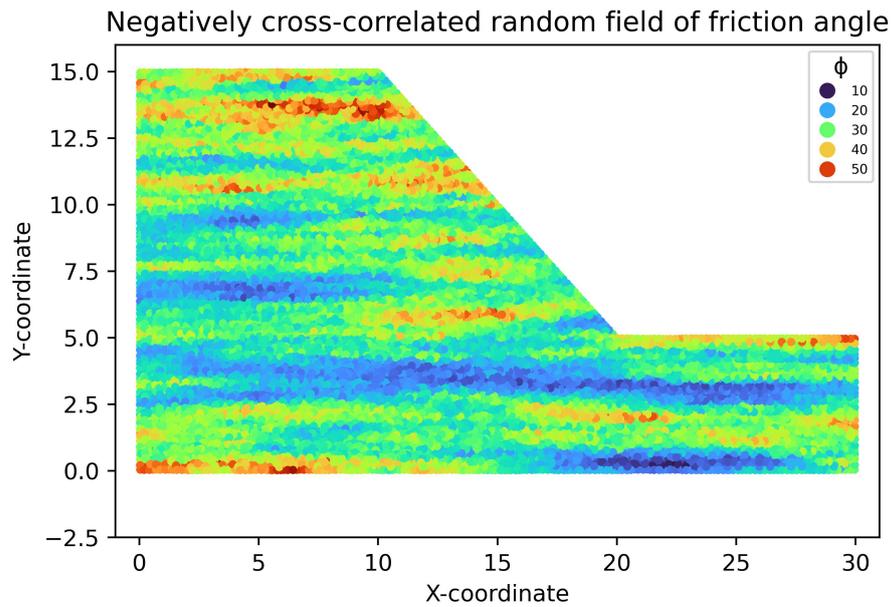


Figure 4.7: Random field distribution for friction angle [°] in Python - $\theta_v = 1$, $\theta_h = 15\text{ m}$

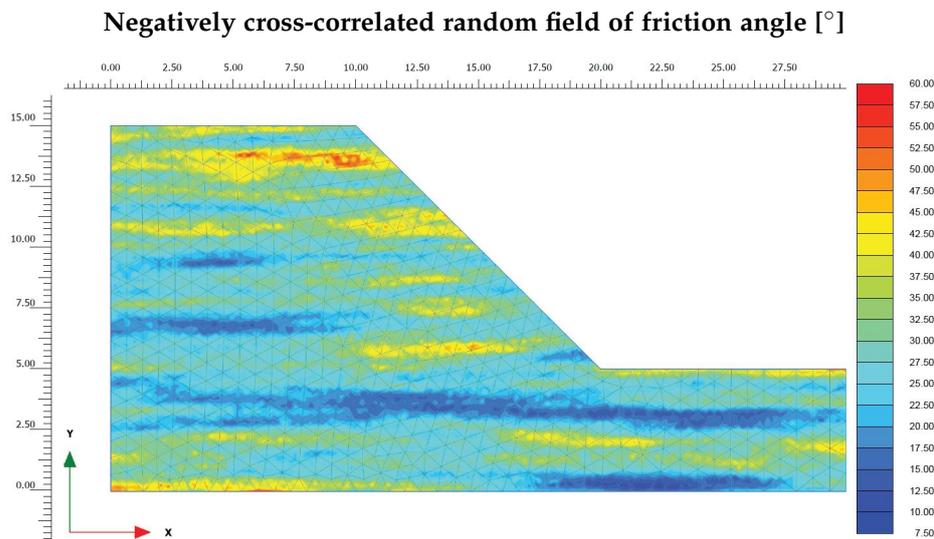


Figure 4.8: Random field distribution for friction angle [°] in PLAXIS - $\theta_v = 1$, $\theta_h = 15\text{ m}$

The past four figures confirm that the random fields are being read and stored properly into PLAXIS' calculation kernel, taking into account the negative cross correlation between c' and ϕ' . It could be observed that for the two random fields, when one showcases areas with high values, the other one shows lower values, as expected due to the negative cross-correlation defined between the two.

This sub-chapter was constructed to help in answering the procedural part of the first research question of this study: *"How can a random field be created, stored and made available to PLAXIS' calculation kernel for the stress point properties?"*

The results presented in this section show that the methodology of implementation described in chapter 3 provides a solution regarding the storage and utilization of random fields in PLAXIS making use of User Defined Soil Models (UDSMs).

4.2 VERIFICATION 1 - 2:1 c' - ϕ' SLOPE

In this section, a more complex verification ensues. To complete the answer to the first research question and (partly) answer the second research question, two examples of simple c' - ϕ' slopes with one soil layer are investigated. These have previously been calculated with existing RFEM codes to assess the influence of spatial correlation and property variability on the probability of failure of a slope. The first example investigates a 2:1 slope, whereas the second example explores a 1:1 slope geometry. In this sub-chapter, the implementation of RFEM in PLAXIS in this thesis is used to recalculate a simple example from literature. In the study of Allahverdizadeh et al. (2015), a research RFEM code was utilized to calculate a 2:1 slope stability problem for a statistical distribution of c' and ϕ' .

The performance parameter chosen is the probability of slope failure. This is computed as mentioned in equation (2.4) as a sum of binary results indicating whether or not the slope fails. In this case, failure is considered to occur when the safety factor computed is lower than 1. As a result, the calculation stages in PLAXIS involve a "Gravity loading" phase, a "Plastic" calculation phase (also called a "Nil" phase) ensuring a better equilibrium of the system and a "Safety" phase in which the safety factor is calculated for the particular realization. This last stage introduces a new feature compared to the original study, which does not compute factors of safety outside of the deterministic case. However, this is deemed useful here for the comparison of the stochastic solution with the deterministic one.

If failure occurs in any of the first two phases, the safety factor recorded is equal to the proportion to which the particular stage was completed to ($\Sigma MStage$ in PLAXIS). The safety factors are packaged into a cumulative distribution function for the case in which the scale of fluctuation tends to infinity. When this is the case, the values of the parameters tend to uniformity throughout the slope and the Monte Carlo analysis results in a set of homogeneous solutions. Given the underlying parameter distribution, it is expected that the mean of this set of solutions would compare well with the deterministic solution obtained with mean parameters.

The geometry of the slope can be seen in figure 4.9, while the parameters utilized are listed in table 4.4 (Allahverdizadeh et al., 2015). For this study, the authors used a unique coefficient of variation for both cohesion and friction angle, $V = 0.4$.

Table 4.4: Inputs for the first verification (Allahverdizadeh et al., 2015)

Parameter	$\mu(c') [kPa]$	$\mu(\tan(\phi')) [-]$	$V(c') = V(\tan(\phi')) [-]$	$\beta [^\circ]$	H [m]	E [Pa]	$\nu [-]$
Value	5	0.364	0.4	26.6	10	10^5	0.3

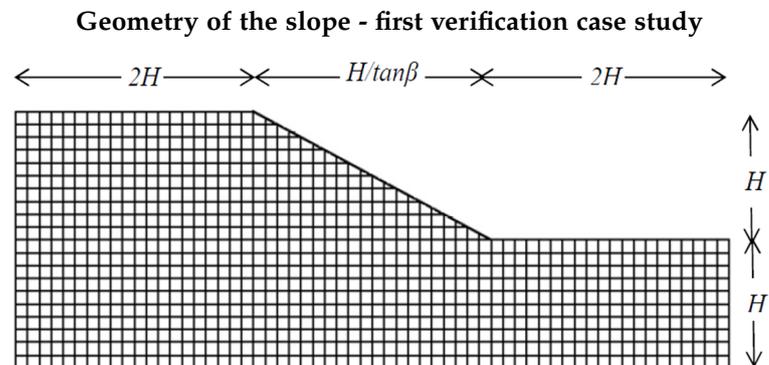


Figure 4.9: Slope geometry of the case study used for the first verification (Allahverdizadeh et al., 2015): H = general problem dimension [m], β = slope angle $[\circ]$

The results obtained by the authors of the original study can be seen in figure 4.10 (Allahverdizadeh, 2015). The scale of fluctuation used is isotropic (equal in both x and y directions) and normalized to the height of the slope ($\Theta = \theta_h/H = \theta_v/H$).

Results of the original case study in the form P_f vs. Θ

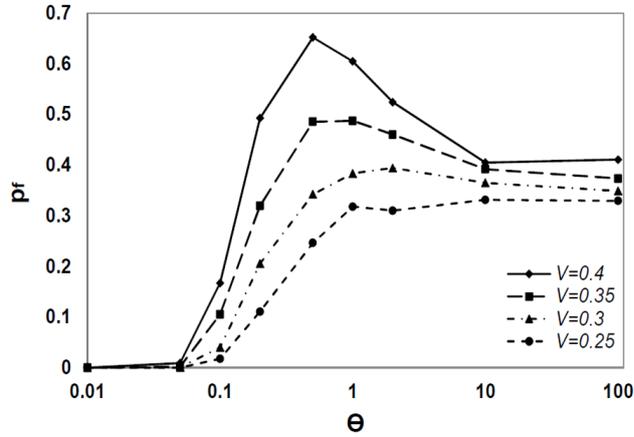


Figure 4.10: Probabilities of failure as a function of scale of fluctuation and coefficient of variation of cohesion for the first verification (Allahverdizadeh et al., 2015)

At this point, some considerations regarding the mesh generation need to be made. In this study, the scale of fluctuation $\Theta = 0.01$ (normalized to the height of the slope) corresponding to $\theta = 0.1$ m is not investigated. This is because in order to capture this scale of fluctuation, a very fine mesh would be required. Looking back on the elaboration in chapter 3.2, capturing this scale of fluctuation would entail a bare minimum of spacing between two integration points smaller or equal to this distance. While in the original code this may not pose significant operational problems, for the analysis in PLAXIS this causes a significant increase in computation time. As a result of this shortcoming, only results for $\Theta > 0.1$ are considered in this verification, for which a fine enough mesh could be created to capture θ while keeping computation times reasonable.

The authors originally used 4000 Monte Carlo realizations to obtain each point on the curve. This was done to ensure repeatability and a good enough confidence in the results for the very low probabilities of failure considered.

In this study, only the results of the uppermost curve in figure 4.10 are used for the verification, for normalized scales of fluctuation starting from $\Theta = 0.1$ and a coefficient of variation of 0.4. As a result, the probabilities of failure investigated do not require that large a number of realizations to achieve the same confidence interval. Using equation (2.5), it was calculated that a number of 1000 realizations is sufficient to achieve a 95 % confidence interval for the target probabilities of failure.

In figure 4.11, a comparison between the probabilities of failure obtained by the literature example and by the current study can be observed. The confidence intervals corresponding to each set of results are also plotted around the curves.

It can be noticed that the RFEM framework implemented in this study results in a good match for probabilities of failure for lower scales of fluctuation. For higher scales of fluctuation, the solutions are not as close, however the same trend is observed. A very large Θ has also been simulated in addition to the solution from literature to check whether the trend of stabilizing probability of failure after a certain value of Θ is respected. This is indeed the case, with the scenario in which the scale of fluctuation is very large ($\Theta \rightarrow \infty$) relative to the problem dimensions yielding the same probability of failure as the case where $\Theta = 10^2$ (i.e. $\theta = 10^3$ m).

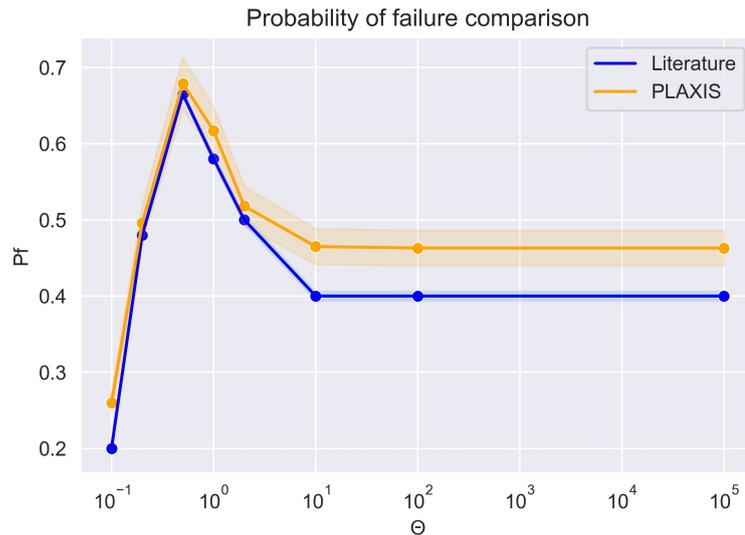


Figure 4.11: Comparison of probabilities of failure between this study and the reference study (Allahverdizadeh et al., 2015)

As mentioned before, the solution in which the scale of fluctuation tends to infinity simulates a distribution of solutions in which the values of the parameters are equal throughout the slope (a set of deterministic analyses). As a result, from a certain value of spatial correlation on, the results of P_f are expected to remain the same for a sufficiently large number of Monte Carlo realizations.

An important observation that can be made is that the tool in this study consistently identifies slightly larger probabilities of failure. A possible explanation for this would be that for a comparable number of elements and stress points, PLAXIS is likely better at identifying failure due to the more complex meshing capabilities. More specifically, in the original study no selective mesh refinement is done.

In PLAXIS, this is not necessarily the case. When creating the mesh automatically, the software identifies the areas where mesh refinement may be necessary. These include zones near structures, under loads and at sharp corners in the geometry. At the same time, this implies a less dense distribution in the rest of the domain, making the computation more efficient in terms of time elapsed to reach a solution. For this analysis, mesh refinement is done near the slope, where sharp corners are encountered, resulting in a much finer mesh in areas of interest for failure development. As a result, PLAXIS may be identifying potential failure planes more frequently, increasing the observed value of P_f in the process. This results in a more conservative solution obtained when employing the current framework.

As mentioned previously, a number of 1000 Monte Carlo realizations was used for all the scenarios, which ensures that all probabilities of failure considered in this analysis are obtained with 95 % confidence as per the formula in equation 2.5.

To demonstrate that this is sufficient, the evolution of probabilities of failure for the cases considered has been plotted against the number of realizations. These can be seen in figure 4.12. It can be noticed that all the curves tend to be horizontal after a certain number of realizations indicating that the solution has stabilized and a larger number of realizations would not yield any significant changes in the result.

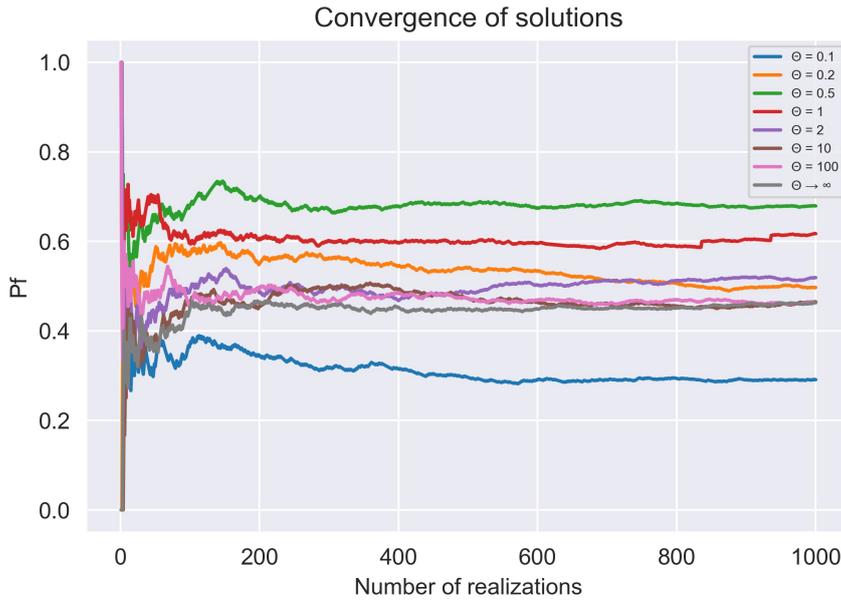


Figure 4.12: Probability of failure evolution as a function of number of realizations for the investigated scales of fluctuation

In addition to this, a plot of the cumulative distribution function of safety factors for a very large Θ (tending towards infinity) is presented. This is useful to check whether the mean and median solutions in this case compare well with the deterministic value of the safety factor. The result set in which $\Theta \rightarrow \infty$ simulates a distribution of solutions in which the values of the parameters are constant throughout the slope, but vary at each realization on the basis of their statistical distribution. The results match well, as it can be seen in figure 4.13.

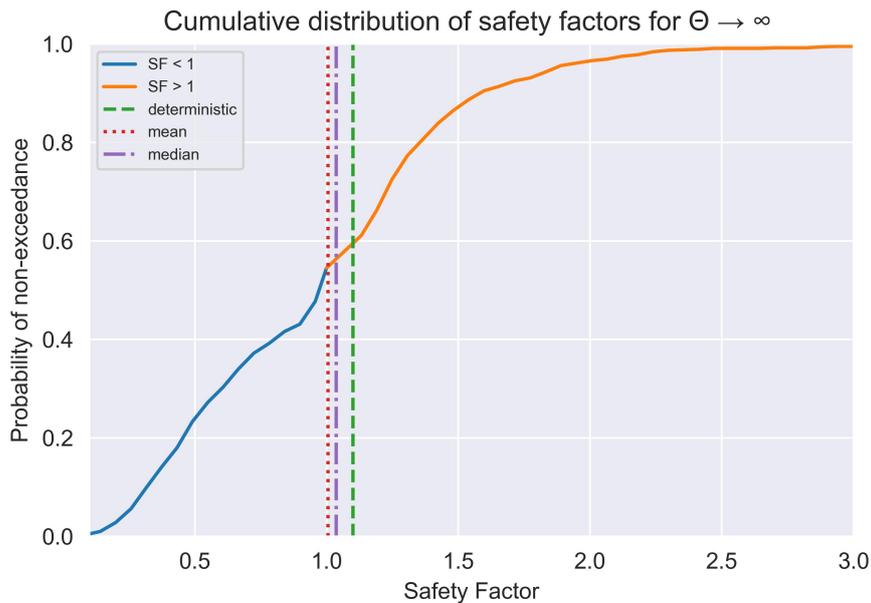


Figure 4.13: Cumulative distribution function of safety factors for the current study along with deterministic solutions for mean, median and the deterministic solution of the original study (Allahverdizadeh et al., 2015)

As mentioned previously, to obtain the safety factors below a value of "1" plotted in figure 4.13, the value recorded is equal to the proportion to which the particular stage was completed to (ΣM_{Stage} in PLAXIS) for the instances in which the model fails before reaching the "Safety" phase.

This may be an explanation for which the curve shape is different before the threshold value of "1" separating failure from non-failure. Even with this distinction, the comparison between the original results and the current ones is a good one.

This section was elaborated to help answering two questions posed in this research:

- *"How can results for multiple different random fields be calculated automatically?"*
- *"How do reliability analysis results with the implemented concept relate to results obtained via existing RFEM codes?"*

Verifying that results obtained in the current study compare well with results obtained with existing RFEM codes from a literature example proves that the methodology of implementation described in chapter 3 provides a valid answer to the first research question concerning the correct implementation of RFEM in PLAXIS.

The PLAXIS model elaboration and results in this sub-chapter provide a valid partial answer to the second research question. In this sense, it is observed that the general trends observed in literature are obtained with this implementation as well.

This preliminary conclusion is going to be verified further, by investigating another, slightly more complex model in the next sub-chapter.

4.3 VERIFICATION 2 - 1:1 $c' - \phi'$ SLOPE

In this sub-chapter, the framework implemented in this study is verified with another case from literature. In this example from Chok et al. (2015), an independent RFEM code was utilized to calculate a 1:1 slope stability for statistical distributions of c' and ϕ' based on distinct coefficients of variation.

As in the previous example, the performance parameter considered is the probability of slope failure. The model setup and assumptions are the same as the one in sub-chapter 4.2. The differences occur in the geometry of the slope, as well as in the statistical variation of strength parameters c' and ϕ' , which is not identical anymore.

The geometry of the slope can be seen in figure 4.14, while the parameters used are listed in table 4.5 (Chok, 2015). In this case, the coefficient of variation for cohesion is always considered double that of the one for the friction angle, with the combination of $V_{c'} = 0.5$ and $V_{\phi'} = 0.25$ being inspected here.

Geometry of the problem - second verification case study

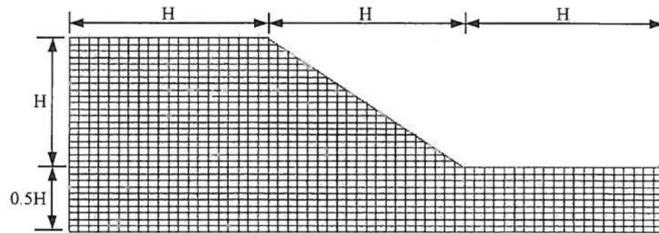


Figure 4.14: Geometry of the slope from the original study tackled for the first verification (Chok et al., 2015). H = general problem dimension [m]

Table 4.5: Inputs for the second verification (Chok, 2015)

Parameter	$\mu(c')$ [kPa]	$\mu(\phi')$ [°]	$V(c')$ [-]	$V(\phi')$ [-]	H [m]	E [Pa]	ν [-]	γ [kN/m ³]
Value	10	30	0.5	0.25	10	10^5	0.3	20

The results obtained by the authors can be seen in figure 4.15 (Chok, 2015). An isotropic scale of fluctuation, normalized to the height of the slope ($\Theta = \theta_h/H = \theta_v/H$) is used for all scenarios.

Results of the original case study in the form P_f vs. Θ

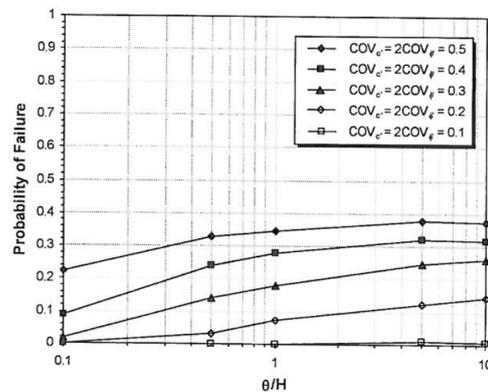


Figure 4.15: Probabilities of failure as a function of scales of fluctuation and coefficient of variation for properties in the original study (Chok, 2015)

As it was the case with the previous reference study, the authors chose 4000 Monte Carlo realizations to obtain each point on the curve. The plot used to justify the decision can be observed in figure 4.16 (Chok, 2015). However, given that in this thesis only the relatively large probabilities of failure from the uppermost curve are inspected, this large a number of realizations is not deemed necessary. Considering again equation (2.5), a number of 1000 Monte Carlo realizations have been considered sufficient to obtain the results with a 95 % degree of confidence.

Solution convergence with more realizations in the original case study

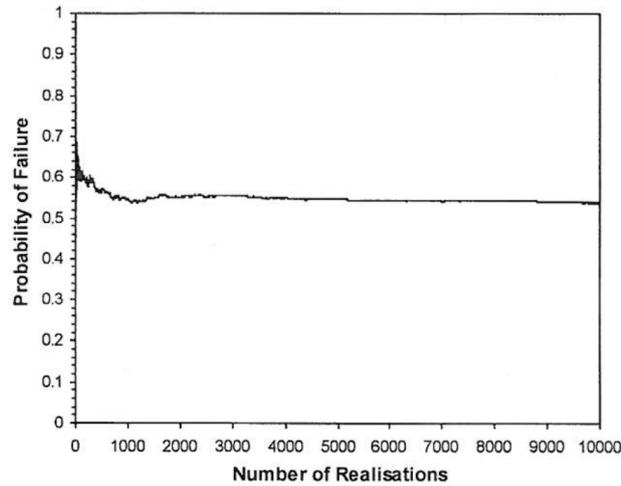


Figure 4.16: Probability of failure evolution as a function of number of realizations in the original study (Chok, 2015)

In figure 4.17 below, a comparison between the probabilities of failure reported in the reference case study and the ones obtained in the present thesis can be observed. The corresponding confidence intervals are also plotted around the curves.

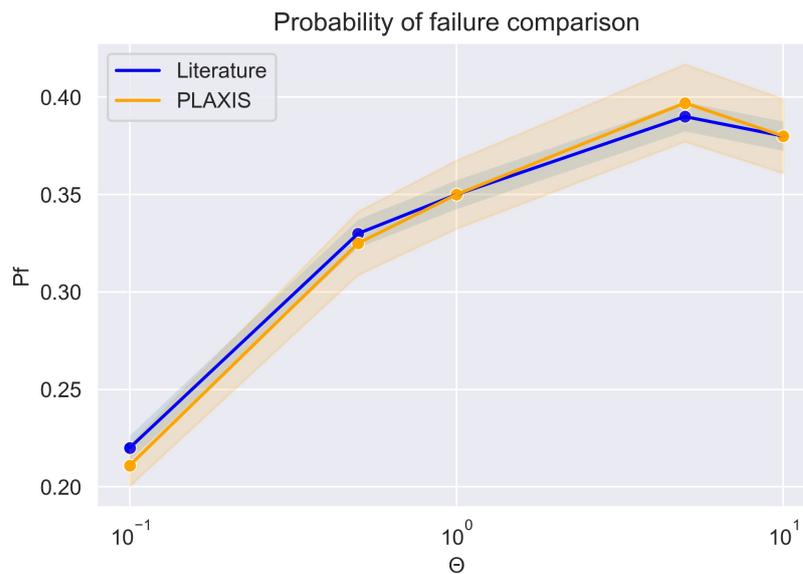


Figure 4.17: Comparison of probabilities of failure between this study and the reference study (Chok, 2015)

It can be noticed that the implementation of the RFEM concept in this study results in a very good match with the reference study for probabilities of failure for all scales of fluctuation considered.

An observation that can be made is that as in the previous example, the tool in this study often identifies slightly larger probabilities of failure. This is especially true for the higher failure probability, for which a lower number of realizations is needed to obtain a certain degree of confidence. For lower probabilities of failure, the calculation is more sensitive to the number of realizations. That being said, the number of realizations chosen in this study identifies the correct probability of failure with 95 % confidence. The differences are within this interval and the 5 % error may partially explain the mismatch between results.

Another possible explanation for the differences in results would be that for a comparable number of elements and stress points, PLAXIS may be identifying failure planes more frequently, due to the shapes of its elements and smart mesh refinement on the basis of model geometry. Besides not making any assumption on where failure planes may occur, no mesh refinement is done in the original study.

As mentioned in the previous example, in PLAXIS this is not necessarily the case. When generating the mesh automatically, the software identifies the areas where mesh refinement is necessary. For this analysis, mesh refinement is done near the slope, where sharp corners are encountered, resulting in a finer mesh in areas of interest for failure development. Consequently, PLAXIS is potentially more accurate in representing failure planes, leading to increased observed values of the P_f .

The evolution of probabilities of failure for the cases considered has been plotted against the number of realizations. This is done to provide visual proof supporting the choice of the number of realizations for the Monte Carlo simulation. These can be seen in figure 4.18 below. It can be noticed that all the curves become horizontal after a certain number of realizations indicating that the solution has stabilized and a larger number of realizations would not yield any meaningful changes.

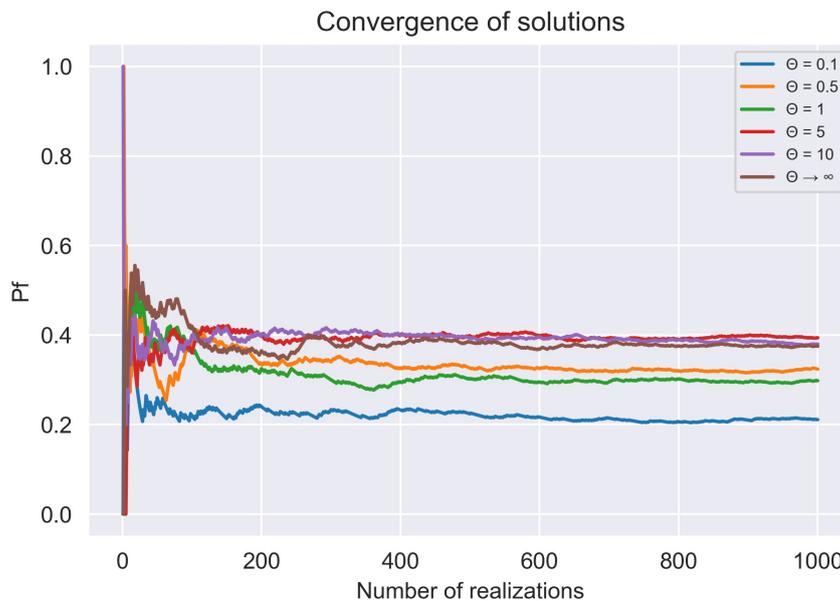


Figure 4.18: Probability of failure evolution as a function of number of realizations for the investigated scales of fluctuation

A solution for a very large Θ has been calculated on top of the results from literature. This is to check whether the mean and median of the distribution of results in this case compare well with the deterministic calculation. As mentioned before, the solution in which $\Theta \rightarrow \infty$ simulates a distribution of solutions in which the values of the parameters are constant throughout the slope, but vary at each realization with values chosen from their underlying statistical distribution.

As a result, given the properties of the underlying log-normal distribution for parameters, it is expected that the mean and median solutions will be very close to the deterministic solution obtained with the mean parameters. This is indeed the case, with the mean solution for the safety factors comparing well with the safety factor declared in the paper – 1.12 – as it can be seen in figure 4.19.

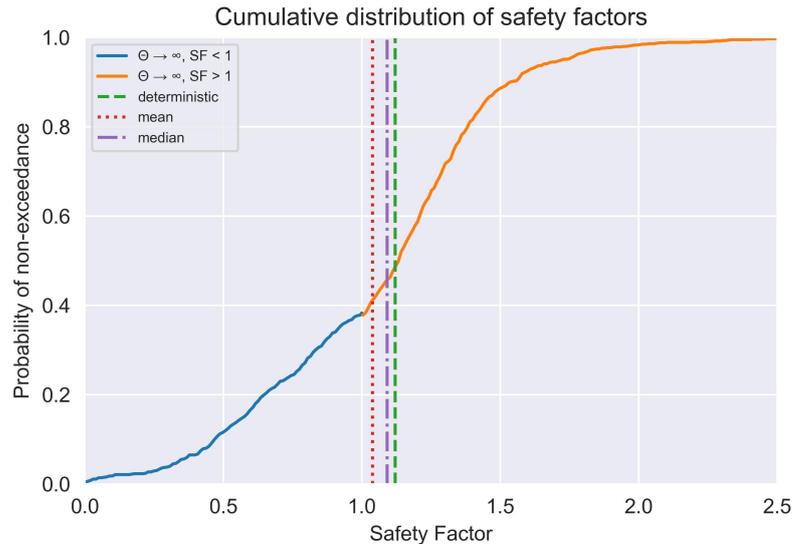


Figure 4.19: Cumulative distribution function of safety factors for the current study along with deterministic solutions for mean, median and the deterministic solution of the original study (Chok, 2015)

On top of this, in appendix F an illustration that presents the CDF_s of safety factors for different scales of fluctuation is presented. This confirms the trend observed in literature, that with smaller spatial correlation lengths, the distribution of solutions tends to be narrowly spread around the mean solution. This is likely due to the mechanism being less affected by local values of strength (smaller or larger) due to considerable averaging of soil property values over potential failure surfaces. On the other hand, with larger scales of fluctuation local (semi-)continuous weak zones are more likely to develop, resulting in a larger range of possible solutions.

Last but not least, in the appendix G readers can consult a series of plots of random fields from PLAXIS' output panel as a function of the scale of fluctuation. This is to illustrate the effect the scale of fluctuation has on the distribution of a parameters' values throughout the domain of the example.

As with the previous case study, the undertaking in this sub-chapter serves to help in answering the first two research questions, concerning the proper extraction of results via RFEM in PLAXIS. By analyzing another case with slightly more complex features, a more nuanced answer becomes possible for the second research question - "How do reliability analysis results with the implemented concept relate to results obtained via existing RFEM codes?". The final step in answering this question is taken in the next chapter, in which the tool created is tested in a real-life scenario previously investigated with existing RFEM codes.

5

VALIDATION CASE STUDY

To validate the implementation of the RFEM concept in PLAXIS, a more complex example needs to be investigated. In this case, the stability of the dyke under its own weight and the influence of a high-water level is checked.

5.1 PROBLEM DESCRIPTION

A recent example of use of RFEM to assess stability is provided by Hicks et al. (2019) for a dyke in Starnmeer, North Holland, which was not meeting safety requirements based on calculations with conventional methods. The study demonstrated that accounting for spatial variability not only results in a more comprehensive safety assessment, but also a less pessimistic one. While the initial deterministic calculations showed that the structure should have failed already, giving safety factors as low as 0.59 in some areas, RFEM results are in line with the most important conclusion – the dyke still stands. This example is used as a reference case study, in order to check whether or not the implementation of spatial variability in PLAXIS yields comparable stochastic results with the original study. The geometry of the dyke before proposed re-design is highlighted in figure 5.1 below (Hicks et al., 2019).

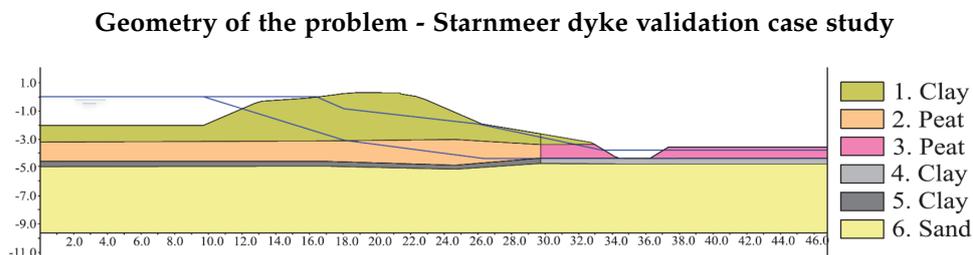


Figure 5.1: Geometry of the validation case study - the Starnmeer dyke (Hicks et al., 2019)

The stability of the dyke under the influence of its own weight and that of a high-water level is investigated. The upper water level represents the global water level, while the lower one corresponds to the water level in the underlying sand layer.

The variable parameters are the cohesion and the friction angle of the soil layers. The point statistics of these parameters are presented in table 5.1 (Hicks et al. 2019) along with values of saturated and unsaturated volumetric weight of the soils.

Table 5.1: Original study parameter inputs (Hicks et al., 2019)

Layer	$\mu(c')$ [kPa]	$\mu(\tan(\phi'))$ [-]	$V(c')$	$V(\tan(\phi'))$ [-]	γ_{unsat} [kN/m ³]	γ_{sat} [kN/m ³]
Clay 1	4.4	0.58	0.773	0.081	6.9	13.9
Peat 2	3.2	0.361	0.656	0.058	9.8	9.8
Peat 3	2	0.358	0.775	0.145	9.8	9.9
Clay 4	4.5	0.559	0.544	0.012	15	15
Clay 5	5.4	0.601	0.352	0.007	15	15
Sand 6	0	0.637	0	0	20	20

Some of the parameters in the original study may be considered problematic or unrealistic. Given PLAXIS' setting for the volumetric weight of water at " 10 kN/m^3 ", values of saturated volumetric weight lower than this value could result in improbable outcomes. Moreover, the uppermost clay layer showcases an unrealistically low value for the unsaturated volumetric weight " 6.9 kN/m^3 ". To overcome these potential issues, slightly different values were used for these entities in this study. The parameters used in the computations in this study are showcased in table 5.2

Table 5.2: Adapted parameter inputs for this study

Layer	$\mu(c') [kPa]$	$\mu(\tan(\phi')) [-]$	$V(c')$	$V(\tan(\phi')) [-]$	$\gamma_{unsat} [kN/m^3]$	$\gamma_{sat} [kN/m^3]$
Clay 1	4.4	0.58	0.773	0.081	11	13.9
Peat 2	3.2	0.361	0.656	0.058	9.8	11
Peat 3	2	0.358	0.775	0.145	9.8	11
Clay 4	4.5	0.559	0.544	0.012	15	15
Clay 5	5.4	0.601	0.352	0.007	15	15
Sand 6	0	0.637	0	0	20	20

The stochastic results obtained in the reference study as a function of horizontal scale of fluctuation can be seen in figure 5.2 (Hicks et al. 2019) below.

Results of the original case study - CDFs of SFs and deterministic solutions

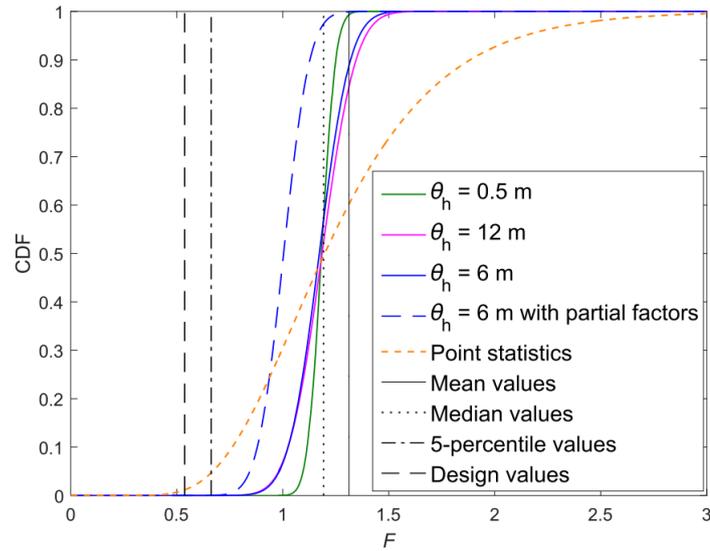


Figure 5.2: Results of the case study in the form of cumulative distribution functions for the safety factor as a function of spatial variability (Hicks et al., 2019)

The tool that was developed as a part of this thesis is utilized to simulate this problem in its original form, without the proposed redesign of the dyke.

The chosen performance parameter is the safety factor. The model setup and assumptions are the same as the one in sub-chapter 4.2 involving a "Gravity loading" phase, a "Plastic" calculation phase (also called a "Nil" phase) and a "Safety" phase in which the safety factor is calculated for each realization.

As before, if failure occurs in any of the first two phases, the safety factor recorded is equal to the proportion to which the particular stage was completed to ($\Sigma MStage$).

The vertical scale of fluctuation is kept constant at $\theta_v = 0.5 \text{ m}$ for the cases in which θ_h is finite ($\theta_h = 0.5, 6 \text{ m}$). For the "point statistics" case, $\theta_v = \theta_h \rightarrow \infty$. To simulate this, a very large value of $\theta_v = \theta_h = 10^9 \text{ m}$ was selected.

To ensure the scale of fluctuation is captured properly, an appropriate mesh needs to be selected. Using the considerations of chapter 3.2, this implies that the distance between two integration points should at least be smaller than the scale of fluctuation considered. As discussed in chapter 3.2, ideally four stress points should be distributed along a length equal to θ . This however should be balanced with operational considerations, as a too large number of stress points can make the calculation times prohibitive, making the attainment of the target accuracy unfeasible.

Taking into account that the original code does not perform any mesh refinement, the option is disabled, moving away from PLAXIS' default. As a result, a regular distribution of elements exists throughout the domain, in the attempt to represent the original situation correctly. The mesh created can be observed in figure 5.3 below, while a zoom in to illustrate the distances between integration points can be consulted in appendix H. There, it can be seen that for remote areas of the domain, the spacing between two integration points is smaller than the smallest scale of fluctuation considered in the study. However, the ideal situation in which four stress points would capture the smallest $\theta = 0.5 \text{ m}$ is unattainable here, as a mesh compliant with this requirement would make calculation time prohibitive. It was observed that for a number of 30,000 integration points, a test run of one realization resulted in the setup crashing under the computational weight proposed.

PLAXIS model mesh - Starnmeer validation case study

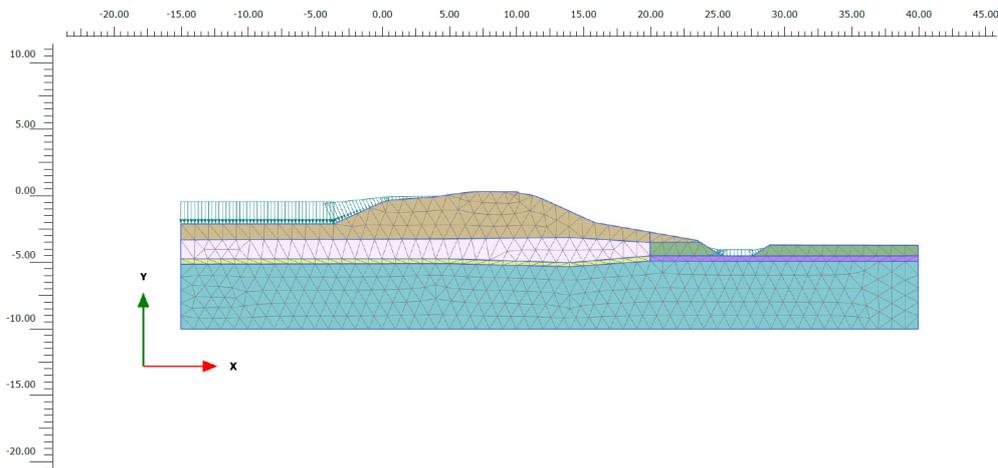


Figure 5.3: Finite element mesh generated for the analysis of the validation case study

This may cause deviations in the representation of the random fields, larger spatial correlations being effectively expressed due to the lack of enough closely-spaced integration points. However, to maintain a balance between accuracy and computation performance, it was deemed that the mesh is sufficient for the scope of this study, but this drawback should be kept in mind in the interpretation of the results.

To illustrate the random fields' distribution throughout the domain, figures 5.4 and 5.5 showcase color maps of random field values (c' and ϕ') over the dyke's geometry, for the case in which $\theta_v = \theta_h = 0.5 \text{ m}$. As expected, pronounced randomness can be observed concerning the values of cohesion and friction angle over the domain. As mentioned in the discussion about the appropriateness of the mesh, in some regions of the geometry, the scale of fluctuation is not captured properly due to insufficient mesh refinement. This causes areas in which the effective scale of fluctuation expressed is larger, potentially influencing the results.

To see how increasing spatial correlation distance influences this distribution, appendix I can be consulted where similar plots are showcased, for the combinations of $\theta_v = 0.5 m$, $\theta_h = 6 m$ and $\theta_v = \theta_h \rightarrow \infty$.

Cohesion random field distribution for $\theta_v = \theta_h = 0.5 m$

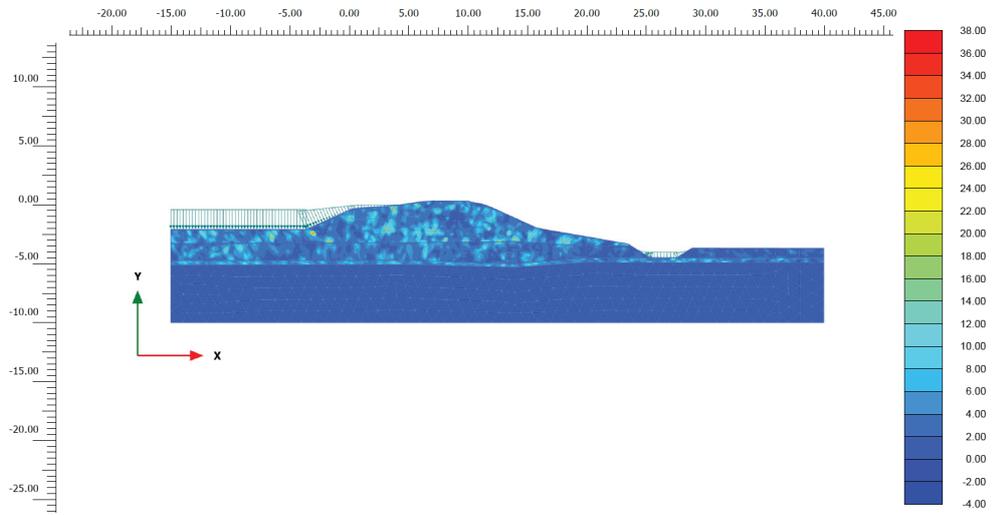


Figure 5.4: Illustration of the random field for $c' [kPa]$ for $\theta_v = \theta_h = 0.5 m$ in PLAXIS

Friction angle random field distribution for $\theta_v = \theta_h = 0.5 m$

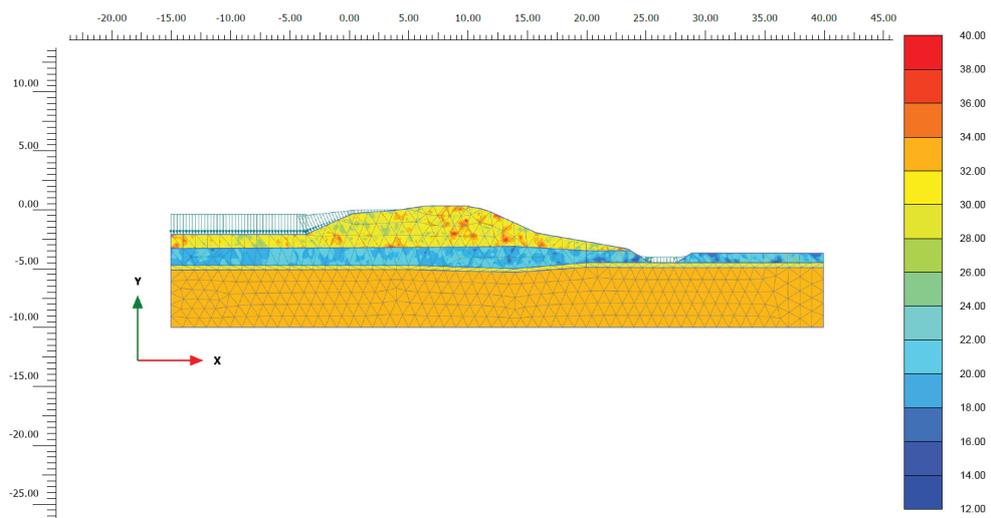


Figure 5.5: Illustration of the random field for $\phi [^\circ]$ for $\theta_v = \theta_h = 0.5 m$ in PLAXIS

5.2 RESULTS

In figure 5.6, the results obtained by the authors of the reference article (Hicks et al., 2019) are compared to the results obtained in this study using RFEM in PLAXIS. More specifically, cumulative distribution functions for the safety factors are compared. On top of this, in table 5.3, a comparison between the probabilities of failure obtained in this thesis and the ones from the reference study is presented.

Comparative results - CDFs and deterministic solutions

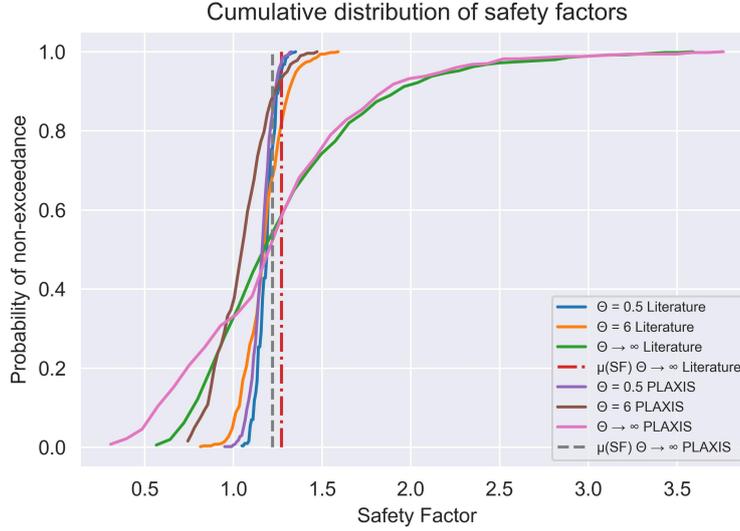


Figure 5.6: Cumulative distribution function of safety factors for the current study compared to the original study (Hicks et al., 2019)

Table 5.3: Comparison of probabilities of failure between the current and the original study

Scenario	P_f PLAXIS	P_f Reference
$\theta_v = \theta_h = 0.5 m$	0.005	0
$\theta_v = 0.5 m, \theta_h = 6 m$	0.37	0.07
$\theta_v = \theta_h \rightarrow \infty$	0.32	0.32

The curves in figure 5.6 compare generally well, especially for the extreme cases in which the scale of fluctuation is isotropic and is either very small or very large. The match for the curve where $\theta \rightarrow \infty$ indicates that the implementation of the finite element model is correct. As mentioned, when θ is very large relative to the size of the domain, results can be seen as a set of calculations with different homogeneous spatial distributions of c' and ϕ' drawn from the underlying parameters' distributions. Therefore, results are expected to be similar to the original ones.

For the situation in which $\theta_v = \theta_h = 0.5 m$, an explanation for the good match could be that with pronounced heterogeneity in the soil, there is a smaller likelihood that semi-continuous weak zones can create and as a result failure planes develop less frequently. As a result, the CDF tends to be spread narrowly around the mean solution. Moreover, in this case, this results in values of SF mostly above the "1" threshold. Hence, the likelihood that the results match is also larger.

That being said, an important difference appears between the P_f obtained in this thesis and the one from the reference study for the intermediate combination of $\theta_v = 0.5 m, \theta_h = 6 m$. In this sense, the mention of the threshold of safety factor equal to "1" is significant not only due to it setting a boundary between failure and non-failure. It also introduces a difference between this study and the original one,

which may explain some of the deviations in results. More specifically, the way in which safety factors below one are extracted is not the same.

A linearly elastic perfectly plastic model is employed in PLAXIS, which does not allow the passing of the failure envelope in loading. As a result, as described in the verification cases (sections 4.2 and 4.3), safety factors below one are counted as Σ MStage if failure occurs in the phases before the "Safety" phase. This also provides a possible explanation for the distinctive shape of the cumulative distribution curves before $SF = 1$ for the PLAXIS implementation in figure 5.6.

Conversely, the reference study uses a different method benefiting from the customization powers of creating the code from the ground up. A visco-plastic method (Zinkiewicz, 1974) is used allowing the stress state to surpass the failure envelope. Using the strength reduction method, the lowest strength reduction factor at which equilibrium cannot be achieved within 500 visco-plastic iterations is taken as the value of SF. This may explain why the largest differences between the curves in figure 5.6 occur for safety factors below this failure threshold.

Another observation which can be made with regards to the general positions of the curves is that the results yielded by the current investigation tend to be more pessimistic than the original ones. All the cumulative distribution functions obtained in this study are shifted to the left compared to the reference ones. This implies a more significant presence of lower safety factors than in the original study.

This could be owed to a variety of factors. As mentioned previously, the largest differences tend to appear in the region in which $SF < 1$, in which a different method of extraction and accounting is utilized. This is likely the most significant reason for the deviation in results between the two studies.

Another possible explanation could be numerical. The shapes of elements are different, with the original study using rectangular elements as opposed to the 15-noded triangular elements taken as default by PLAXIS. Moreover, the number of integration points is not identical. In this study, only the logical assumption that the distance between two points should not be larger than the scale of fluctuation has been made. However, no information is known about how many stress points or elements have been generated for the calculation in the original study.

Moreover, as mentioned when discussing the mesh in figure 5.3, in some remote parts of the mesh the conditions discussed in chapter 3.2 for the stress points' spacing are not complied with. This can also influence the accuracy of the results, although the influence of this component is likely relatively small when comparing it to the aforementioned core numerical factors.

Finally, a potential reason for the different results may be the adaptation of parameters. Even though the changed set of parameters yields the same deterministic result, it is possible that under the influence of randomness and the other factors of deviation (different way of choosing $SF < 1$, element shapes and stress point distributions), the error propagates, causing a difference in the stochastic solutions.

A more detailed view of the same curves as the ones in figure 5.6 can be consulted in appendix J, in which comparisons are plotted specifically for each scale of fluctuation. Using this vantage point, one can observe that the match between curves tends to be much better for the extreme cases than for the intermediate situation. Moreover, the influence of using a different method for extracting safety factors below the failure threshold on the curves shapes can be observed more clearly.

This chapter was the final step in elaborating an answer to the second research question *“How do reliability analysis results with the implemented concept relate to results obtained via existing RFEM codes?”*.

In analyzing a real life case study, a more nuanced answer can now be formulated for this question, understanding the merits and shortcomings of RFEM in PLAXIS. Preliminary insights include:

- General trends identified in literature (Hicks, 2014) with respect to the influence of spatial variability on slope stability are confirmed with this implementation as well. More specifically:
 1. For relatively small scales of fluctuation, the failure mechanism passes through strong and weak zones in virtually equal measure. As a result of this averaging of soil properties, the distribution of solutions tends to be narrowly spread around the mean solution.
 2. For intermediate scales of fluctuation relative to domain size, it becomes possible for failure to propagate through semi-continuous zones of weakness. This happens as failure planes seek the paths of least resistance, hence the lengthier the weak zones, the likelier failure becomes. As a result, the distribution of safety factors widens, yielding larger probabilities of failure as values of horizontal spatial correlation lengths increase.
 3. For very large spatial correlation distances, the “effective” distribution of parameters tends towards the underlying parameter distribution. As a result, a large range of possible solutions is obtained, with the mean of solutions approximately coinciding with the deterministic solution obtained with the mean of parameters. This could be used by users as a conservative solution in a stochastic analysis, in the case in which no information exists concerning the scale of fluctuation.
- As seen in the results of sections 4.2 and 4.3, and the set of cumulative distributions in figure 5.6, PLAXIS tends to identify larger probabilities of failure in geotechnical problems, for a suitable selection of mesh size and number of realizations. This is potentially owed to the more sophisticated meshing possibilities in-built in PLAXIS, such as selective mesh density increases in key areas. Moreover, as the failure plane is not fixed a priori, the failure mechanism is allowed to find the path of least resistance through the soil.

That being said, interpretation should be meticulously done, as for cases in which the mesh density is not sufficient, the effective scale of fluctuation captured by the model would be larger, resulting in semi-continuous weak zones not expected from the original definitions of spatial correlation values. These likely ensue increased likelihood of failure, translating to larger values of P_f .

- An upper threshold exists for the extent of mesh refinement and implicitly for scale of fluctuation representation. It was observed that surpassing a number of 30,000 integration points when trying to capture θ accurately all over the domain, the Python code crashed under the computational weight proposed. Technically, the size of the auto-correlation matrix exceeds the memory available, causing the script to crash. This is problem specific and once exceeded, calculation is not feasible anymore. As a result, accuracy may come at the expense of operational performance, reason for which a balance needs to be carefully obtained by the user.

6

DEMONSTRATIVE CASE STUDY

In this chapter, a realistic problem which has not been previously analysed with RFEM is tackled. The purpose of this case study is to showcase the potential the implementation in this thesis has unlocked for the resolution of more intricate problems while accounting for the spatial variability of soil properties.

6.1 PROBLEM DESCRIPTION

In this case, the feasibility of building a foundation next to a slope is investigated. For this situation, both the variability of strength parameters (c' and ϕ') as well as that of stiffness (characterized by the shear modulus G) are considered. This allows for a multi-purpose analysis of both the stability (ultimate limit state) as well as the operational performance of the structure (serviceability limit state).

By quantifying the displacements at the corners of the foundation, one could obtain a perspective about how spatial variability may influence differential displacements occurring as a result of the construction of a foundation of a certain weight. This is further elaborated to understand if the differential displacements cause the violation of the foundation tilt serviceability criterion, as set by the Eurocode 7 (CEN, 2004).

In figure 6.1 below, the location of the potential site of the foundation on the coast of Constanta, Romania is illustrated (Google, 2022). The geometry extracted from the side view is utilized to set up the 2D model. Simplifying assumptions are made implying the soil is uniform in the 3rd dimension and that the other buildings are far enough from the site not to exert any influence on the behaviour of the foundation.

Case study tentative construction site

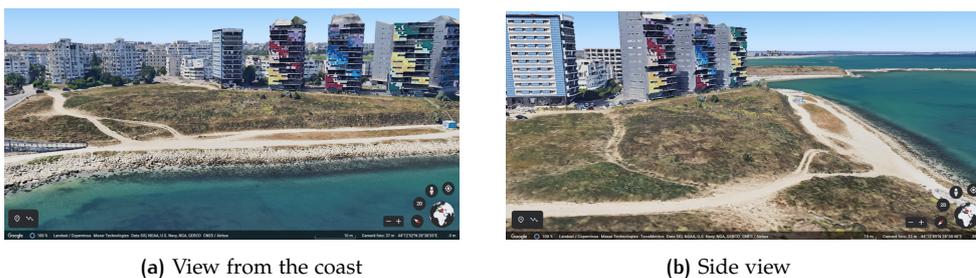


Figure 6.1: Views of the site of the foundation analyzed in this problem (Google, 2022)

Using a typical soil distribution from the area extracted from a geotechnical study, the PLAXIS model is set up. The soil column includes three layers, while the foundation is set to be a rigid foundation, modelled as a plate accompanied by a line load on top. To perform the serviceability check, the displacements at the corners of the foundation are extracted. The mesh created including the aforementioned points of emphasis can be seen in figure 6.2 on the next page.

PLAXIS model mesh with points of output extraction

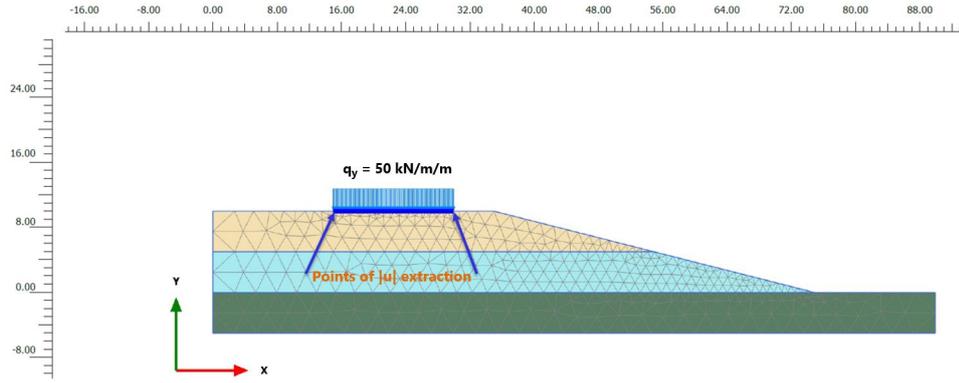


Figure 6.2: Mesh with output extraction points for the demonstrative case study

The mean parameters extracted from the site investigation report are listed in table 6.1, while the properties of the foundation are shown in table 6.2. The statistical variation of soil parameters is chosen using ranges reported by Cherubini (2000).

Table 6.1: Stochastic inputs for soil properties in the analysis of foundation serviceability

Layer	$\mu(c')$ [kPa]	$\mu(\phi)$ [°]	$\mu(G)$ [kPa]	$V(c')$ [-]	$V(\phi)$ [-]	$V(G)$ [-]
Top - Silty Sand	5	24	6,000	0.35	0.1	0.3
Mid - Calcareous Sand	21	23	9,000	0.035	0.01	0.03
Bottom - Silty Clay	66	15	10,000	0.035	0.01	0.03

Table 6.2: Inputs for plate properties in the analysis of foundation serviceability

Property	w [kN/m/m]	EA_1 [kN/m]	EI [kN/m ² /m]	$ q_y $ [kN/m/m]
Values	7	$2 * 10^7$	$54 * 10^4$	50

To exacerbate the illustrative impact of the example, the top layer's properties are considered to vary more significantly, as this top stratum is the one interacting directly with the foundation. The two soil layers below it are considered a firm base, whose strength and stiffness parameters vary only slightly.

To check whether this assumption holds, a deterministic safety analysis of the problem with mean parameters was performed. The illustration, which can be checked in appendix K shows that the failure mechanism is local and does not extend to the bottom layers. This makes the assumption reasonable, but any such hypotheses should be checked as part of the model preliminary review. This calculation of a safety stage on top of the iterative model yielded a safety factor of 2.5, showing that without spatial variability considerations, the situation is deemed far from failure by the conventional analysis. Moreover, this hints that low probabilities of failure can be expected for ULS in case of very large spatial correlation values.

Different scenarios are considered in order to capture the impact of spatial variability on structural performance. These are listed in table 6.3 below.

Table 6.3: Scenarios of analysis as a function of θ

Scenario	θ_v [m]	θ_h [m]
I	2	2
II	2	10
III	∞	∞

First, a vertical scale of fluctuation of two meters is fixed for the initial two scenarios. The first scenario simulates a situation in which a lot of spatial averaging and a relatively homogeneous behavior of the soil layers are expected.

Then, a more realistic case is considered by increasing the correlation distance horizontally. Finally, a situation in which both the vertical and horizontal spatial correlation are very large ($\theta \rightarrow \infty$) relative to the size of the domain is considered. This selection is similar to that in previous case studies considered, in order to verify whether similar trends in the distributions of results are observed.

As with every numerical model, a critical evaluation of the mesh is necessary. With RFEM, this is even more significant, as an inappropriate mesh would not fully capture the spatial variability defined by the user, as elaborated in chapter 3.2. Retaining in mind the guidelines there, figure 6.3 provides a detailed look at one of the remote areas in the mesh of the model presented in figure 6.2, where the elements, and implicitly the spacing between stress points, are the largest.

Detailed view of the largest elements in the mesh

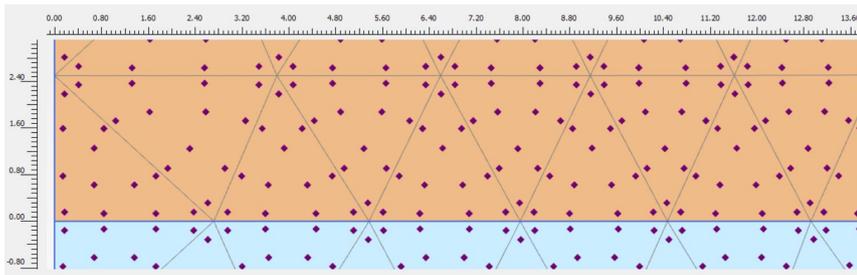


Figure 6.3: Zoom in view on stress point (purple) spacing at a remote area in the mesh

Making use of the scale and stress point illustrations provided by PLAXIS' output panel, it can be observed that for the smallest scale of fluctuation considered (two meters), at least three stress points are distributed along this distance, be it vertically or horizontally. This ensures that the spatial correlation is properly captured and that the mesh is not too coarse. Depictions of property values' distribution over the geometry can be consulted in appendix L, reinforcing this statement with illustrations of the influence of spatial variability on parameter values over the domain.

Having established the validity of the model assumptions and the appropriateness of the mesh, failure criteria need to be set. For the ultimate limit state, failure is defined as in previous examples by total structural failure in any of the stages defined for the model. For serviceability, failure is defined as the point where the value of a the serviceability criterion is exceeded. These are packaged in cumulative distributions of values and probabilities of failure as defined by equation 2.4.

The performance threshold for foundation tilt is set by the Eurocode (CEN, 2004) at "1/300" which corresponds to approximately 0.2 degrees. This is quantified irrespective of direction, although by accounting for spatial variability, one could potentially also get an idea of preferential directions of tilt for this geometry.

In the staged construction panel, the following phases are defined:

- *Initial Phase* - Gravity loading - creates a stress state over the geometry;
- *Nil Phase* - Plastic analysis - Ensures a "better" balance of the stress state;
- *Load Phase* - Plastic analysis - Structure and load activation.

For this example, at the end of each Monte Carlo realization, the output panel is opened such that results for structural elements are extracted. This is something to be taken into account when considering computation times, which are slowed down slightly by opening and closing the output application at each realization.

6.2 RESULTS

In the paragraphs that follow, results are presented, either in the form of cumulative distributions of values or as probabilities of failure. In this latter case, a goal accuracy of 90 % is selected. This is done to achieve a balance between accuracy and operational performance. Having an extra random parameter (G), coupled with the extensive geometry and fine mesh to capture the scale of fluctuation, implies that lowering the accuracy criterion is needed to keep computation times acceptable.

In conformity with equation 2.5, a number of 750 realizations was needed to ensure this accuracy is reached for all performance parameters, dictated by the cases showing the lowest probabilities of failure in the preparatory review. The lower standard of accuracy practiced compared to previous examples is a consequence of increasing model complexity. Hence, aiming for a larger accuracy would have made computation times prohibitive for the scope of this analysis.

The first check is done for the general slope stability, i.e. ultimate limit state. This is quantified in terms of the probability of failure, as described in all previous examples. The evolution of P_f for the slope under the load of the foundation, as a function of the number of realizations performed is illustrated in figure 6.4.

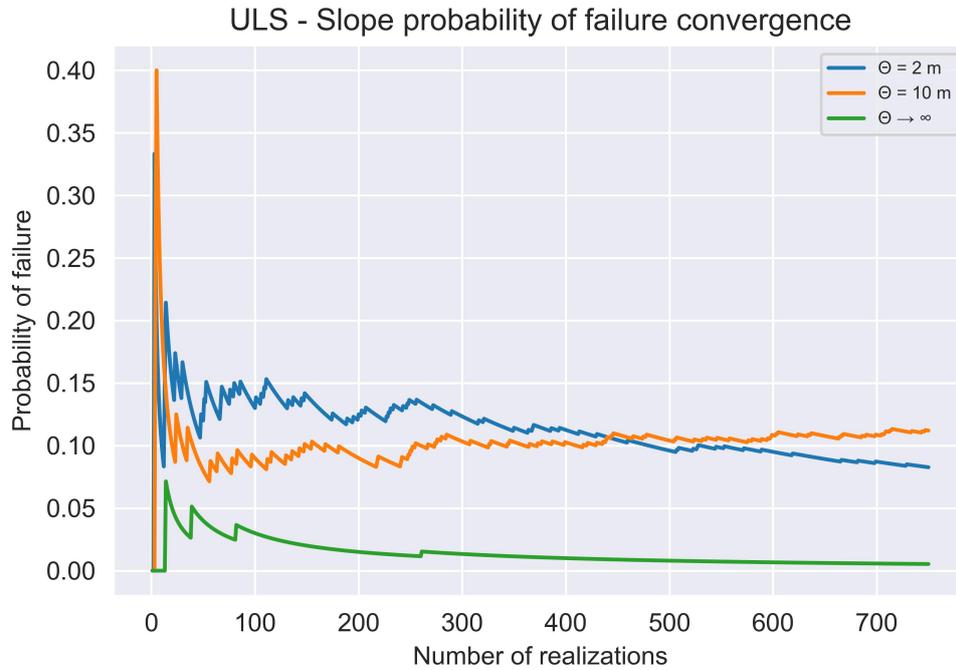


Figure 6.4: Slope failure probability as a function of number of realizations

In this particular situation, it can be noticed that in the extreme cases ($\theta_v = \theta_h = 2 \text{ m}$ or $\theta_v = \theta_h \rightarrow \infty$) the probabilities of failure tend to be lower. This could be suspected, as in both cases the behaviour of the soil is approaching that of a homogeneous block. Since slope failure is not influenced as significantly by the geometry (deterministic $SF = 2.5$), the main negative mobilizer is the load imposed by the foundation weight. Under a constant load, the worst case situation is encountered when the spatial variability is pronounced enough such that weak zones can form, which could result in failure under the imposed load.

This result proposes a very interesting practical implication. Even with a sensitivity analysis, a set of deterministic results would not be able to capture this phenomenon, as proven by the extremely low probability of failure showcased by the

situation in which $\theta \rightarrow \infty$. However, for the same geometry and load imposed, if spatial variability is accounted for, probabilities of failure become more significant and must be taken into account in the design process.

Moving on, in figure 6.5 below, the cumulative distributions of tilt can be observed for the cases in which the scales of fluctuation are finite, plotted along the serviceability limit proposed by the Eurocode. In the case in which spatial correlation is infinite, there is no tilt, so the results are not meaningful, therefore not presented.

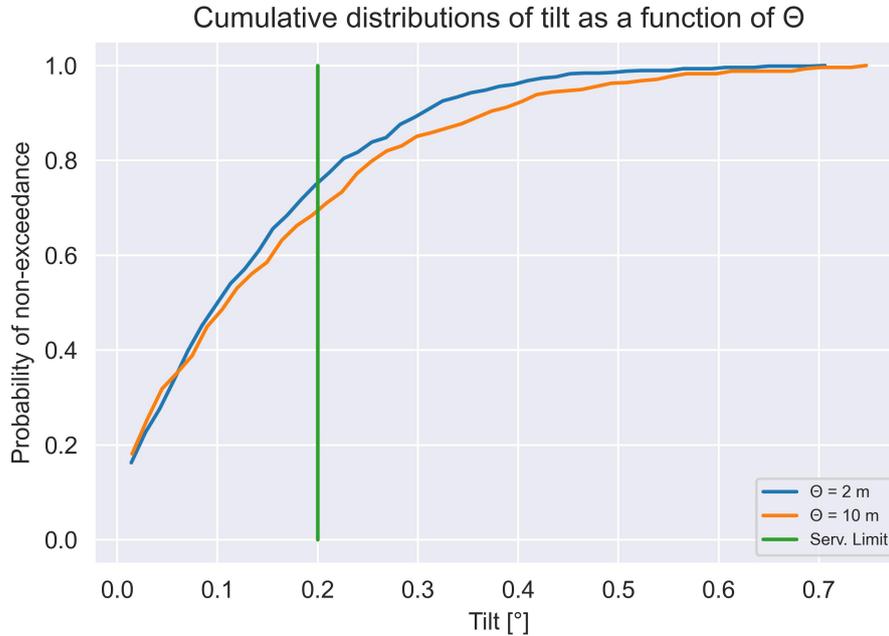


Figure 6.5: CDF of tilt [°] comparison for $\theta_h = 2m$ and $\theta_h = 10m$ (both with $\theta_v = 2m$)

It must be noted that this figure only illustrates the tilt obtained for cases in which the structure does not fail. The rationale behind this is the consideration that if ultimate limit state failure occurs, then this would imply serviceability failure also occurs. However, if the structure fails, the displacements displayed in the final phase would be inappropriately large values for the purpose of this illustration.

As in previous observations in literature, for an intermediate horizontal scale of fluctuation, the likelihood of encountering a weak zone at the interface with the foundation is larger. As a result, the probability of exceeding the serviceability limit is more significant as well. Conversely, when small scales of fluctuation relative to the foundation length ($L_{foundation} = 15m$) are considered, the soil underneath acts more uniformly due to more pronounced spatial averaging.

To show that the number of realizations performed is sufficient, the evolution of failure probability with respect to the tilt serviceability criterion ($P_{f,Tilt}$) as a function of number of Monte Carlo realizations is presented in figure 6.6 on the next page.

An important observation that needs to be made with respect to this figure relates to the number of realizations needed to approach the real solutions. It can be seen that for the first ≈ 400 realizations, probabilities of failure fluctuate and the hierarchy is blurred, before the real trend starts crystallizing. This goes to show that the importance of the number of realizations needed to reach a stable solution should not be understated. Hence, users should try to define expectations in this sense before running their analyses by using equation 2.5 for a predicted P_f .

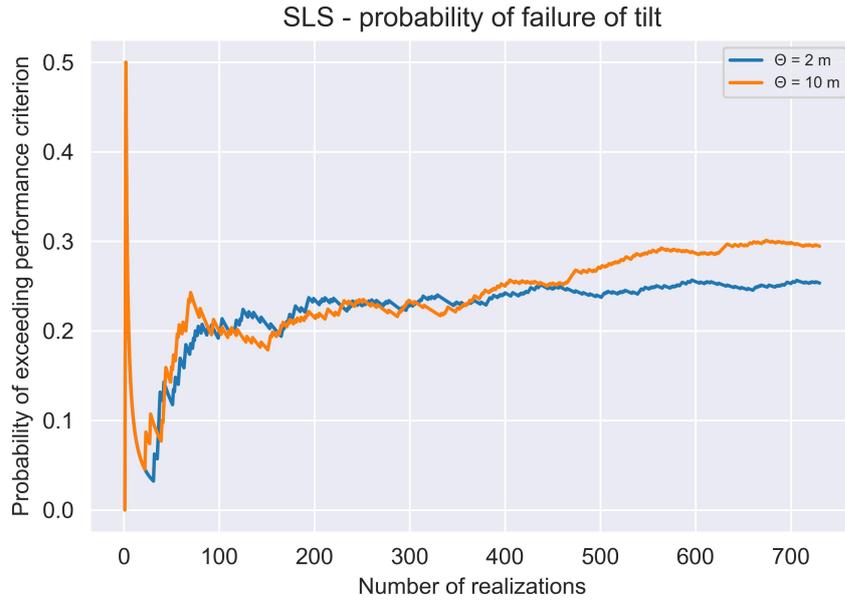


Figure 6.6: Tilt P_f as a function of number of realizations for the serviceability analysis

This chapter was entirely constructed to help answering the final research question posed in this study: “How can the created tool be utilized to solve more complex geotechnical analyses, while taking into account the spatial variability of soil properties?”. The elaboration presented in this section, built on the insights exposed previously in the report illustrates how the implementation of RFEM within PLAXIS’ capabilities unlocks a number of new avenues of investigation with RFEM:

1. Users can now utilize the concept behind RFEM to consider spatial variability in multi-purpose analyses using the MC constitutive model in PLAXIS. Calculations can involve both ULS and SLS checks, all while easily accounting for construction stages, variable water levels, or soil-structure interaction information that can be extracted through outputs available in PLAXIS.
2. By accounting for spatial variability in problems involving soil-structure interaction, meaningful outcomes can be revealed and quantified:
 - When slope failure is not decisively influenced by geometry (deterministic $SF = 2.5$), it has been observed that for a constant load, failure probability becomes significant due to the development of semi-continuous weak zones underneath the foundation. Confirming a trend identified in literature case studies without external loads imposed on the slope, the P_f observed is larger for an intermediate horizontal scale of fluctuation ($\theta_h = 10\text{ m} \rightarrow P_f = 0.12$) as opposed to a small one ($\theta_h = 2\text{ m} \rightarrow P_f = 0.08$). Crucially, a set of deterministic analyses ($\theta \rightarrow \infty$) would be oblivious to this possibility, as shown by the $P_f \rightarrow 0$ in figure 6.4.
 - The magnitude of differential settlements can be grasped in a more realistic manner. The probability of exceeding the EC7 serviceability criterion for tilt can also be quantified subsequently. A similar hierarchical trend is observed in the probability to exceed this performance goal (figure 6.6), with $\theta_h = 10\text{ m}$ resulting in a $P_{f,Tilt} \approx 0.3$, while for a smaller $\theta_h = 2\text{ m}$, $P_{f,Tilt} \approx 0.25$, assumed to be the result of the same mechanism of semi-continuous weak zones development. In this case, results of a set of deterministic analyses ($\theta \rightarrow \infty$) also deemed the violation of this criterion impossible ($P_{f,Tilt} = 0$), reinforcing the merits of this method.

7

GOOD PRACTICES AND REMARKS ON FUNCTIONALITY

Given the intricate nature of this elaboration, a section on good practices in modelling is considered indispensable. First, the chapter provides a general recipe for the utilization of the created tool. Then, factors influencing the performance of modelling for this particular implementation are discussed.

7.1 RFEM IN PLAXIS RECIPE

While a certain degree of user customization is possible, it is always useful to have a systematic approach when it comes to modelling procedures. This sub-chapter provides this as a distillation of experiences acquired by the author throughout this research. A visual representation of the steps involved in the employment of RFEM with PLAXIS is presented in figure 7.1 below.

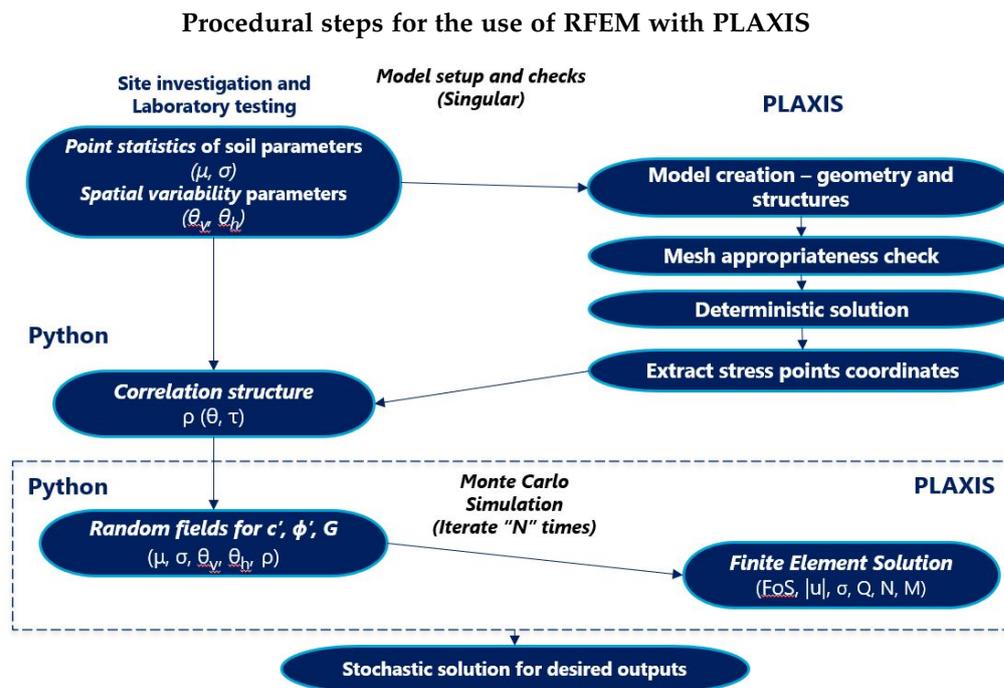


Figure 7.1: Recipe for systematic use of RFEM in PLAXIS

The procedure is described in more detail in the enumeration on the next pages. The first three steps are not specific to RFEM and are general good practices for site investigation data evaluation and finite element modelling. That being said, some features need to be selected to accommodate for spatial variability, such as a fine enough mesh to capture θ properly. Steps 4 to 7 are specific for RFEM and their execution could be done independently after setting up a PLAXIS model "manually".

1. Obtain parameter distributions

One of the crucial steps in the practical implementation of RFEM is the determination of the statistical distributions for the parameters employed. Just as importantly, the spatial correlation parameters should be obtained by:

- Laboratory testing:
 - Mean and standard deviation (μ, σ) of parameters.
- Site investigation:
 - Mean and standard deviation (μ, σ) of parameters;
 - Scale of fluctuation values (θ_h, θ_v) - ideally using a suitable number of closely-spaced data points (de Gast, 2020).

2. Set up the PLAXIS model

This important building block starts with creating the geometry and material data sets based on the mean/characteristic properties inferred from the site investigation. Between the setup of structural elements and the definition of water levels and construction stages, a mesh review is imperative. To ensure the compliance of the mesh with the criteria discussed in chapter 3.2, the following checks are needed when using the framework in this study:

- Ensure θ is captured properly in both directions. The distance between two stress points should not be larger than the spatial correlation length considered in that particular direction;
- Ideally, 4 stress points should exist over a distance equal to the spatial correlation length (Spencer, 2007). For instance, if $\theta = 1 \text{ m}$, the distance between two stress points should ideally be $\leq 0.25 \text{ m}$.

3. Deterministic analysis

This step is important for a number of reasons, ranging from expectation management to operational goals. It may be followed by a mesh sensitivity analysis in a range filtered using the previous step. Functions include:

- Setting a benchmark in analysis and complying with conventional calculation standards. These benchmark outputs can be used to define a performance criterion. For instance, a realization could be counted as attaining failure if the total displacement exceeds the value obtained in the deterministic analysis with characteristic values. With a sensitivity analysis and this reference value, users can define an expectation of distance from failure of the model, in relation to the selected performance criterion. This expected P_f value is then used to predict the number of realizations needed to achieve a goal accuracy using equation 2.5;
- Allowing the user to check the appropriateness of the model, concerning boundary influences or any other assumptions made. For instance, the demonstrative case study in this thesis (chapter 6) implied significant parameter variation only in the top layer, assuming the two bottom layers act as a firm basis. If the deterministic analysis revealed influence of the latter on the failure mechanism, the assumption would be invalidated;
- Allowing the user to access the output panel, from which stress point coordinates can be extracted, manually or using the scripting interface. These are used to generate spatially correlated fields in the next step.

4. Random field generation

This step is detailed in the text around figure 3.2 of sub-chapter 3.1.1. In sum:

- Definition of point statistics (μ, σ) and spatial correlation parameters (θ_v, θ_h) ;
- Computation of the distance matrix \rightarrow spatial correlation structure;
- Generation of spatially-correlated random fields from an underlying statistical distribution (here standard normal);
- (Optional) Definition of a cross-correlation matrix "R" and computation of its Cholesky factorization "L";
- (Optional) Calculation of cross-correlated random fields " ζ ";
- Transformation to log-normally based variables (using 3.1, 3.5 and 3.6):
 - " Z_{LN} " for non-cross-correlated fields (3.5);
 - " ζ_{LN} " for cross correlated fields (3.6).

5. Pre-run checks

- Saving the number of integration points to a text file which will be read by the UDSM when employing the constitutive model in calculation;
- Creation of PLAXIS objects for the construction stages and the pre-allocation of output variables in the correct size, with vector lengths corresponding to the number of realizations ordered for the Monte Carlo simulation.

6. Run the Monte Carlo loop

- Exporting the random fields from the Python script to a text file at each realization. The destination and format of this file should correspond to the way in which this was coded in the UDSM. By placing it in the loop, a different random field is utilized for each realization;
- Marking the phases for calculation;
- Ordering the calculation and extract the desired output variables;
- Quantifying the duration of the analysis for operational evaluation purposes as a final command of the loop.

7. Post-processing

- Computation and illustration of P_f evolution as a function of the number of realizations (using the definition of P_f in equation 2.4);
- Calculation and illustration of stochastic results for desired outputs (PDFs, CDFs of safety factors, displacements, forces etc.)

This is of course the recipe found most appropriate by the author on the basis of the experience acquired in elaborating this thesis. Other users may adapt the order of steps to their preferences, but the logical sequence is likely to be the same.

In the next section, some considerations concerning the factors influencing the operational performance of the implementation are described in more detail.

7.2 OPERATIONAL CONSIDERATIONS

As suggested in previous chapters, the upside of the tool elaborated and described in this thesis does not come without trade-offs. These are most decisively related to computation times and how choices made by users may strengthen or undermine the efficiency of calculations. A balance needs to be found between accuracy and speed, for which users need to consider the following:

1. Mesh density needed

- *Geometry*: This is one of the more stable elements to consider, as it is dictated by the problem to be solved. The smallest domain that does not impose boundary influences should be selected to limit the remote number of integration points needed for accurate problem representation.
- *Scale of fluctuation* (see 3.2 for more details): A mesh should be selected such that the smallest scale of fluctuation is captured properly. If ensuring this is not possible throughout the domain, this should at least be true in the areas deemed most influential to the calculation.
- *Number of integration points*: The combination of the previous two factors, results in a final number of integration points. This is the quantitative variable to be optimized. With a smaller number of integration points, less values need to be transported to PLAXIS easing the load to be taken by the calculation kernel. If too large, the number of stress points can make the computational expense of the analysis unacceptable.

2. Calculation characteristics

- *Number of construction stages*: This is an influential degree of freedom. Naturally, the larger the number of phases, the longer the duration of a realization. This is likely not subject to optimization. Nevertheless, it should be taken into account when assessing the performance of a model.
- *Desired output variables*: Depending on the type of variables to be extracted, the output panel may or may not require opening. In an ultimate limit state analysis, this is not needed as outputs can be extracted directly from the input panel at the end of a calculation phase.

3. Influential soil parameters

- *Number of parameters*: Depending on the type of analysis performed, one could expand the UDSM to take into consideration as large a number of variables as possible. This becomes counterproductive from a certain point on, as the increase in computation times is not linear when increasing the number of parameters. This can be seen in table 7.1, in which a comparison between calculation times for different numbers of random parameters considered in the final case study is presented. Consequently, practitioners should exercise engineering judgement and "randomize" only the most influential parameters for the calculation undertaken.

Table 7.1: Computation times as a function of random parameters in the final case study

Parameter no.	Time/realization [s]
1	53
2	92
3	190

For a complete overview of the average computation times recorded in the case studies of this thesis, readers are referred to appendix M.

8

DISCUSSION

The overarching objective of this research was to make a first step in bridging the gap between science and practice by making the consideration of inherent spatial variability of soil properties accessible in software widely used in geo-engineering practice. This undertaking does not come without caveats, prompting a need for a discussion around the implications of this project.

Thus, this chapter addresses the nuances and limitations surrounding the findings presented in this thesis. In treating separately the implementation, validation on the reference studies from literature and the new possibilities showcased with the demonstrative case study, readers can link back these observations to the three main research questions stated explicitly in section 1.2.

8.1 IMPLEMENTATION

1. General

- *Parameter determination and uncertainties*

As pointed out throughout the document, the main driver of this research direction is the need to capture soil heterogeneity, which is expensive to characterize through direct methods. Given the importance of spatial variation observed both in literature, as well as in this study, users should be aware of potential uncertainties throughout the data supply chain.

When considering spatial variability, it is worth noting that a large range of values have been reported in literature, especially for the horizontal scale of fluctuation (Nie et al., 2015; Arnold, 2016). Furthermore, it may be challenging to define a unique value of θ for a site, as more than one value could be observed depending on the source of heterogeneity (Vanmarcke, 1983). For instance, a site may be characterized by a large value of θ which can be traced back to the geological deposition trends and a much smaller one originating from the local hydrodynamic history.

Details on these, as well as methods for estimating vertical and horizontal scales of fluctuation, while quantifying the uncertainty of assessments are elaborated extensively by de Gast (2020).

Besides the inherent soil variability, uncertainty may propagate along the supply chain through statistical uncertainty associated with number of measurements (Student, 1908) as well as transformation errors occurring when translating from measurements to model parameters (Ching & Phoon, 2015; Wang et al., 2017; van der Krogt et al., 2019).

This is all to say that a critical view of statistical distributions and spatial correlation of parameters is imperative. These need to be obtained and meticulously reviewed for each project before being used in calculation.

2. RFEM in PLAXIS specific

- *Constitutive modelling and UDSMs*

The implementation of RFEM in PLAXIS in this study was performed using an example UDSM, provided by PLAXIS to support users in this endeavor. This implied an already programmed constitutive relationship, in this case based on the Mohr-Coulomb failure criterion. Generally, this soil model is only used for “crude” approximations, unable to capture more complex soil features, such as hardening, softening or stress history (Bentley Systems, 2022).

This should be taken into account by users that may want to use the tool developed in this thesis as a ready-made product, potentially justifying more advantageous outcomes. The ability to capture a previously ignored feature of the ground in a probabilistic way does not exclude the need for users to properly express relevant behavioral characteristics of different soils.

- *Dimensions considered*

In this study, the implementation of spatial variability in PLAXIS is only done for the 2D version of the software. The same concept can be expanded towards 3D modelling. This capacity to solve models accounting for the three spatial dimensions is in fact one of PLAXIS’ strong suits.

That being said, one of the main drawbacks of RFEM, which is even more glaring in this employment in PLAXIS, is the calculation expense. As a result, it is foreseeable that such undertaking would make this shortcoming even more noticeable hindering the practical appeal of the framework.

- *Further automation*

The procedure developed and proposed in this study is relatively intricate and it involves a potentially sensitive connection between many interfaces: Python for scripting, FORTRAN for DLL creation and PLAXIS for model creation. This may become an inhibiting factor for practitioners in the journey towards wide adoption in the engineering community.

With more time allowing for development, the tool could become more user-friendly, through better programming limiting the need for user interaction with the source code, all while improving the user interface.

8.2 VALIDATION

In the examples tackled for the validation of the framework proposed, trends are matching generally well, but some differences between results still occur. These could come as a result of deviations from the procedures used by authors of the original studies, or other limitations, both outlined below:

- *Calculation method*

When employing the Mohr-Coulomb model formulation in calculations, a linearly elastic perfectly plastic computation method is utilized. In the form utilized by PLAXIS, the stress state cannot surpass the failure criterion.

Conversely, the case studies in the verification steps (chapter 4), as well as the validation on the Starnmeer dyke (chapter 5) employ a visco-plastic algorithm (Zinkiewicz & Corneau, 1974). This starts by an application of gravity loading. As soon as stress states at certain points exceed the failure envelope, these yielding stresses are redistributed to neighbouring elements which still have strength reserves. Eventually, non-convergence of the algorithm for a chosen maximum number of iterations is used to define slope failure, usually simultaneous with large nodal displacements (Griffiths & Lane, 1999).

- *Finite elements characteristics*

Another meaningful difference between this study and the original literature examples treated occurs at the level of the finite elements properties. For starters, PLAXIS uses triangular finite elements, while the results of the original studies utilized for verification (chapters 4.2, 4.3) are obtained employing rectangular elements in computation.

Moreover, the number of integration points in the examples is not stated explicitly. Even if in the verification cases, this could be inferred indirectly from the shapes of the elements and the area of the slope, a level of uncertainty still exists, therefore the number of integration points cannot be replicated exactly.

Furthermore, as detailed in the validation chapter 5, in the default mesh generation PLAXIS performs a mesh refinement near areas expected to be of increased importance (sharp corners, near structures, loads). Therefore, even if the number of stress points was equal, the refinement and the differences in element shapes would make their comparative locations distinct. Consequently, results are unlikely to match perfectly.

- *Meshing to capture θ*

Several studies have shown that by accounting for the spatial variation and location of measurement data, structural response uncertainty reduces (Jaksa, et al. 2005, Lloret-Cabot et al., 2012; Li et al., 2016).

As discussed in chapter 3.2, to capture the spatial variability properly, certain stress point spacing requirements exist. In extensive geometries, it is challenging to comply with these requisites, as a large number of integration points would be needed to ensure compliance all over the domain. In turn, this would determine a more significant computational expense, which at some point can make the calculation time prohibitive.

In the validation case (chapter 5), it was observed that if more than 30,000 integration points were generated, the distance matrix creation caused the Python API to crash. This could become a significant limitation for very complex problems, requiring detailed modelling over extensive geometries.

- *Nature of verification examples*

Although very illustrative and easy to follow due to their simplicity, the verification cases (chapters 4.2, 4.3) essentially represent numerical demonstrations. Their chosen parameter combinations are not extracted from a real site investigation and are likely selected to drive home a point: the importance of spatial variability in stochastic quantification of failure probabilities.

As a result, even though they serve as indication that the implementation of RFEM in PLAXIS is performed correctly, the results should be examined critically. More focus should be placed on the validation case study (chapter 5) which is based on more realistic parameter distributions.

Another caveat of the verification in this thesis is represented by the results used as references from the original study. In this research, the curves showing the highest probabilities of failure have been chosen for comparisons, as they required the lowest numbers of realizations to obtain converged solutions with high degrees of accuracy (equation 2.5).

In practice, for structures where consequence-based risk is large, very low probabilities of failure are required. Hence, the method needs to be tested further in situations requiring low P_f and high accuracy. That being said, due to the significant computational expenses of RFEM, coupling this implementation with more advanced methods like subset simulation (van den Eijnden & Hicks, 2017) may be necessary for accurate predictions of very low P_f values.

8.3 EXPANSION

Variation in soil properties has been shown to influence system behaviour in many types of geotechnical applications. Researchers have analysed problems involving shallow, strip and pile foundations (Jaksa et al., 2005; Suchomel & Masín, 2010; Naghibi et al., 2016), retaining walls (Sert et al., 2016), liquefaction of hydraulic sand fills (Wong, 2004; Hicks & Onisiphorou, 2005; Popescu et al., 2005) and most extensively slope stability (Griffiths & Fenton, 1997; Hicks & Samy, 2002; Spencer & Hicks, 2007; Hicks & Spencer, 2010; Hicks et al., 2014; Li et al., 2016; Varkey, 2020).

The demonstrative case study tackled in chapter 6 proposes a hybrid situation in which the behaviour of a foundation next to a slope is investigated. The novelty occurs in the serviceability analysis performed which shows the importance of spatial variability in the performance of the structure with respect to tilt, as opposed to previous studies which have generally only considered ultimate limit state.

While showcasing the potential of the implementation, this study is not enough to prove the whole palette of new options this research unlocked. To increase confidence in the utility of the framework, this needs to be tested on other applications as well, such as different types of foundations, excavations, retaining walls etc.

9

CONCLUSIONS AND RECOMMENDATIONS

The practical objective of this thesis was to create, validate and test a framework of implementation of the random finite element method (RFEM) in PLAXIS. This would allow practitioners to account for spatial variability of soil properties observed in site investigations in their FEM models, getting one step closer to replicating the real soil distribution in standard calculations.

Furthermore, the overarching objective of this research line is to bridge the gap between science and practice regarding general consideration of soil properties' heterogeneity in standard calculations, all while approaching a state where stochastic analyses are engineering practice, surpassing the paradigm of safety factors.

In this sense, the findings of this thesis are summarized as answers to the research questions formulated in section 1.2. Recommendations for proper use of the framework and for further research bring this document to its natural completion.

9.1 CONCLUSIONS

1. How can spatial variability be implemented in PLAXIS?

a) *How can a random field be created, stored and made available to PLAXIS' calculation kernel for the stress point properties?*

- **Creation of random fields - general:** This step takes place in a Python code encircling the PLAXIS model, via the procedure described in detail in the text surrounding figure 3.2 of sub-chapter 3.1.1 and the example code in appendix A. Using point statistics (μ, σ) and spatial variability properties of parameters (θ_v, θ_h) from laboratory and site investigations, with stress point coordinates from the PLAXIS model, users can create spatially-correlated random fields on the basis of the distances between stress points. For this, an appropriate correlation model (here Markovian, 3.2) and a suitable underlying probabilistic distribution (here log-normal) also need to be selected as a basis for random field generation.
- **Storing and availability in kernel - RFEM in PLAXIS specific:** This step is performed by linking the aforementioned Python code with the adapted UDSM, as described in sub-chapter 3.1.3 of the methodology. The modifications required are explained in detail in sub-chapter 3.1.2 and summarized by figure 3.4. The adaptation of the UDSM implies programming it to read the random fields generated and exported from the Python code. Storing can be made efficient using a command that orders the UDSM to read the fields for a realization only once (see appendix B for syntax details).

As UDSMs define the stress-strain relationship for each integration point, the code needs to be forged to access the corresponding value at the correct integration point from the previously read random field. This is possible using existing variables from the original code and a known relationship between global and local integration points' positions (equation 3.7).

- b) *How can results (probabilities of failure, safety factors, deformations etc.) for multiple different random fields be calculated automatically?*

After setting up the phases in the PLAXIS model, this step is done by ordering the Monte Carlo simulation loop from the scripting interface as described in section 3.1.3 and exemplified in appendix C. This implies ordering the calculation of each stage of the model, while extracting the desired output at the end of each loop iteration. Depending on the desired variables, opening PLAXIS' output panel may or may not be needed.

For ULS analyses, it is sufficient to quantify at which point a non-safety phase failed (via the $\Sigma MStage$ parameter), or by extracting the reached safety factor (Reached ΣMsf) for a safety phase from the input panel. Conversely, in SLS calculations, the output panel needs to be opened at each realization to allow for the extraction of variables such as displacements, or forces experienced by structural elements of the model. This has a small adverse influence on the computation time of a realization.

2. How do reliability analysis results with the implemented concept compare to results obtained via the existing RFEM codes?

- a) As shown in chapters 4.2, 4.3 and 5, trends identified in literature for the reliability of slopes are confirmed through the present framework as well. More specifically, observations of the influence of spatial variability on slope stability are consistent with previous investigations (Hicks, 2014):
- *For small scales of fluctuation relative to domain size:* The soil acts as a block due to pronounced averaging of soil properties. This makes potential failure planes passing through weak or strong zones virtually just as likely, impeding the propagation of these failure surfaces. This results in a narrow distribution around the mean solution.
 - *For intermediate scales of fluctuation relative to domain size:* In this configuration, semi-continuous weak zones start creating. As failure follows the path of least resistance, in this case the spread of results (SF_s) increases and higher probabilities of failure are observed.
 - *For very large scales of fluctuation relative to domain size:* This situation yields the largest range of possible values. As described previously, this represents a set of deterministic analyses with random values of parameters selected from the underlying parameter distribution. As a result, the distribution of solutions tends towards the shape of the underlying parameter distribution.
- b) The probabilities of failure observed in this thesis are in general larger than the ones in the studies chosen as references for comparison. Assuming the deviations in the numbers of stress points have a minimal impact, differences could occur due to PLAXIS' more sophisticated meshing capacities and as a consequence of the different solving algorithms utilized by the two approaches compared.

The former refers to the shapes of the elements (triangular as opposed to rectangular in original studies), as well as the mesh refinement performed by PLAXIS in areas of interest (sharp corners, near structures and loads) which may induce the increased observed failure probability.

The latter refers to the fact that PLAXIS uses a linear-elastic perfectly plastic algorithm in which the failure criterion is never surpassed, while custom codes generally use a visco-plastic algorithm in which the criterion can be temporarily surpassed. In these existing RFEM codes, failure is quantified when convergence is not reached after a certain number of iterations in the stress redistribution phase. This difference may also be a cause of the larger probabilities of failure observed with this method, as PLAXIS proposes a more strict measure of quantification of P_f .

- c) A limiting factor when comparing this framework to the reference studies is represented by an upper threshold in the number of stress points that can be used before modelling becomes unfeasible. It was observed that surpassing a number of 30,000 integration points when trying to capture θ accurately all over the domain caused the Python code to crash under the computational weight proposed by the model. This occurs as a result of a too large correlation matrix being created when a very large number of stress points is generated.

3. How can the created tool be utilized to perform more complex geotechnical analyses, while taking into account the spatial variability of soil properties?

By coupling RFEM with PLAXIS' existing capabilities in the demonstrative case study in chapter 6, a hybrid situation was investigated, which was not previously explored in literature using existing RFEM codes. Expanding the implementation to account for stiffness by making the shear modulus (G) a random variable, both the reliability and the serviceability with respect to the tilt criterion of a foundation next to a slope have been quantified:

- It was observed that a situation that is deemed far from failure by a deterministic analysis ($SF = 2.5$) could showcase meaningful probabilities of failure in the ultimate limit state when taking into account the spatial variability of soil properties. More specifically, for a constant $\theta_v = 2\ m$ an intermediate $\theta_h = 10\ m$ results in a $P_f = 0.12$, while a smaller $\theta_h = 2\ m$ yields an observed $P_f = 0.08$. Conversely, in a set of deterministic analyses ($\theta \rightarrow \infty$) the probability of failure was found to be virtually zero ($P_f \rightarrow 0$), emphasizing the importance of spatial variability of soil properties in geotechnical reliability analyses. In this situation (ULS), results are governed by the spatial variability of strength parameters (c and ϕ).
- In terms of serviceability, the probability of exceeding the failure criterion for tilt, set at 0.2 degrees by the Eurocode 7 (CEN, 2004) is also observed to be significant when accounting for spatial variability of soil properties. For a constant $\theta_v = 2\ m$, the risk hierarchy confirmed expectations, with an intermediate $\theta_h = 10\ m$ resulting in a $P_{f,Tilt} \approx 0.3$, while a smaller $\theta_h = 2\ m$ showcased a $P_{f,Tilt} \approx 0.25$. Crucially, by making spatial correlation distance infinite ($\theta_v = \theta_h \rightarrow \infty$), the probability of exceeding the tilt criterion is virtually zero, illustrating the importance of considering the spatial variability of soil properties in serviceability checks. When quantifying serviceability (SLS) in terms of deformations, the behaviour is governed by the heterogeneity of stiffness (here modelled using G).

9.2 RECOMMENDATIONS

This final section of the report branches in two conceptual directions. The first set of recommendations is concerned with the effective use of the developed tool and avoiding potential pitfalls in its utilization. The second and final part of this undertaking provides a series of recommendations for potential future research, in terms of additional testing and possible expansion of the current framework.

1. Effective use of the framework

- (General) *Expectation management before running RFEM*

General FEM good practices involve performing a deterministic analysis to check the model appropriateness. In addition to this, in this framework a sensitivity analysis to assess relative closeness to failure is also useful. In case an objective is not preset with respect to the probability of failure, this may help a user define expectations about the number of realizations needed to achieve a certain accuracy (2.5). Quantifying the number of realizations needed could also be an iterative process, in which users run a reasonable number of simulations and on the basis of the P_f observed define a required number of realizations to achieve goal accuracy. For example, for $P_f = 0.2$, 100 realizations would yield a result with 80 % confidence, giving the user an idea of the range of possible values of P_f , consequently guiding further analysis requirements.

- *Mesh density vs θ*

The most important consideration for effective use of this framework concerns the relevance of obtaining a good balance between the accurate representation of spatial variability and an optimally low number of stress points to keep calculation times reasonable. As discussed in 3.2, users should aim to capture a scale of fluctuation length (θ) with 4 stress points to ensure a meaningful enough correlation between points, while allowing for closely spaced uncorrelated points to be well represented.

A commonly experienced pitfall when tinkering with the mesh level of refinement is forgetting to modify the coordinates being read by the code when not using the fully automatic code in appendix D. An error message would pop up in the phase execution, prohibiting the calculation.

- *Number of random parameters*

Depending on the type of analysis performed, users may choose to select a version of the UDSM modelling only the most influential parameters as random fields. For instance, in an ultimate limit state, the strength parameters are the significant ones in dictating failure or non-failure. In contrast, in a SLS analysis the ultimate strength is likely not reached and the key parameter in inferring deformations accurately is the stiffness of soils. Therefore, users may choose to run RFEM only with a limited number of the most significant parameters as random fields, improving computation times for a limited accuracy trade-off.

- *Stiffness variability*

When stiffness is used as a random parameter, it is important to first check the parameter value distributions over the geometry before running the model. For very large coefficients of variation, stiffness values may become zero at certain points resulting in unreasonable deformations and potentially causing the PLAXIS model to crash.

- *Problem size issues*

As mentioned in the answer to the second research question (2c in 9.1), a limiting factor could be imposed by a (too) large number of stress points generated in the design of the project. A potential solution would be a more efficient method of random field generation, such as one using the Local Average Subdivision (LAS) algorithm (Fenton & Vanmarcke, 1990).

2. Further testing and expansion

- *Additional testing*

As mentioned in the discussion, the framework was only tested with one situation different from existing scientific literature - a serviceability analysis of a foundation next to a slope. Time constraints did not allow for the trial of the method on other types of examples, such as different foundation types, retaining walls, or other geotechnical problems.

To increase confidence in the framework, this should be employed in a variety of other investigations where it could prove its added value for geotechnical design. For example, different types of foundations can be investigated, such as strip, or piled foundations.

- *Expanding to 3D*

This framework could be expanded to PLAXIS 3D as well, as any conceptual novelty is not required for this development. That being said, even though the concept would be the same, the calculation expense drawback would become even more apparent, making this undertaking difficult for extensive implementation with current computational capacity.

- *Expansion to more complex constitutive models*

As previously discussed, this framework was developed on the basis of an example UDSM, whose constitutive relationship is based on the Mohr-Coulomb failure criterion. This is only used for "crude" approximations in practice. Therefore, if the wider objective of extensive implementation is to be attained, this implementation needs to be expanded to more complex soil models - such as the hardening soil model. This may prove challenging without a formulation of the soil model which is proprietary, making it most likely an internal undertaking to PLAXIS.

- *User experience refinement*

That being said, even in this case, the framework is not entirely user-ready. This is because the described methodology of usage still involves interaction with the source code. While time constraints inhibited this endeavor, a more friendly user interface should be developed in which practitioners only define model characteristics (geometry, statistical parameters, scales of fluctuation) and the code would run the full simulation automatically, without any need to manipulate the source code.

REFERENCES

- Arnold, P., Fenton, G., & Hicks, M.A. (2012). *Modern geotechnical design codes of practice*. IOS Press.
- Arnold, P. (2016). *Probabilistic modelling of unsaturated slope stability accounting for heterogeneity*. [PhD. thesis, University of Manchester, UK]. UoM repository. <https://www.research.manchester.ac.uk/portal/en/theses>
- Bentley Systems. (2022). *PLAXIS 2D/3D V22.01 – User Manual*.
- Brinkgreve, R.B.J. (1994). *Geomaterial Models and Numerical Analysis of Softening*. [Doctoral Dissertation, Delft University of Technology, NL]. TU Delft repository. <https://repository.tudelft.nl>
- CEN. (2004). *Eurocode 7: Geotechnical Design. Part 1: General Rules, EN 1997-1*. European Committee for Standardisation, Brussels.
- Cherubini, C. (2000). Reliability evaluation of shallow foundation bearing capacity on c' - ϕ' soils. *Canadian Geotechnical Journal* 37(1), 264–269
- Ching, J. & Phoon, K.-K. (2015). Reducing the transformation uncertainty for the mobilized undrained shear strength of clays. *Journal of Geotechnical and Geoenvironmental Engineering* 141 (2).
- Dyson, A., Tolooiyan, A. (2019). Probabilistic investigation of RFEM topologies for slope stability analysis. *Computers and Geotechnics*, 114, 103129. <https://doi.org/10.1016/j.compgeo.2019.103129>
- van den Eijnden, A. P. & Hicks, M. A. (2017). Efficient subset simulation for evaluating the modes of improbable slope failure. *Computers and Geotechnics*, 88, 267–280.
- van den Eijnden A, Hicks M. (2018). Probability-dependent failure modes of slopes and cuts in heterogeneous cohesive soils. *Géotechnique Letters*, 8(3), 214–218. <https://doi.org/10.1680/jgele.18.00043>
- Fenton, G. A. (1999) Random field modelling of CPT data. *Journal of Geotechnical and Geoenvironmental Engineering*, 125(8), 844-857.
- Fenton, G. A., Naghibi, F. & Hicks, M. A. (2018). Effect of sampling plan and trend removal on residual uncertainty. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 12(4), 253–264. <https://doi.org/10.1080/17499518.2018.1455106>
- Fenton, G., Farzaneh, N., Griffiths, D.V. (2016a). On a unified theory for reliability-based geotechnical design. *Computers and Geotechnics*, 78, 110-112.
- Fenton, G. A., Naghibi, F., Dundas, D., Bathurst, R. J., & Griffiths, D. V. (2016b). Reliability-based geotechnical design in 2014 Canadian highway bridge design code. *Canadian Geotechnical Journal*, 53(2), 236–251. <https://doi.org/10.1139/cgj-2015-0158>
- Fenton, G. A. & Griffiths, D. V. (2008). *Risk assessment in geotechnical engineering*. John Wiley & Sons.

- Fenton, G. A. & Vanmarcke, E. H. (1990). Simulation of random fields via local average subdivision. *Journal of Engineering Mechanics*, 116(8), 1733-1749. [https://doi.org/10.1061/\(asce\)0733-9399\(1990\)116:8\(1733\)](https://doi.org/10.1061/(asce)0733-9399(1990)116:8(1733))
- de Gast, T., Vardon, P. J., & Hicks, M. A. (2017). Estimating spatial correlations under man-made structures on soft soils. *Geo-Risk 2017*. <https://doi.org/10.1061/9780784480717.036>
- Gast, T. de, Vardon, P.J., & Hicks, M.A. (2019). Observations and considerations regarding estimating horizontal scales of fluctuation around linear infrastructure. *Proceedings of the 7th International Symposium on Geotechnical Safety and Risk (ISGSR 2019)*. <https://doi.org/10.3850/978-981-11-2725-0-is12-3-cd>
- de Gast, T. (2020). Dykes and Embankments: a Geostatistical Analysis on Soft Terrain. [PhD. thesis, Delft University of Technology, NL]. TU Delft repository. <https://repository.tudelft.nl>
- Google. (2022). Map of the coastline of Constanta. Retrieved on the 1st of September from <https://earth.google.com/>
- Griffiths, D. V. & Fenton, G. A. (2004). Probabilistic slope stability analysis by finite elements. *Journal of Geotechnical and Geoenvironmental Engineering*. ASCE. 130(5), 507-518.
- Griffiths, D.V., & Lane, P.A. (1999). Slope stability analysis by finite elements. *Geotechnique*, 49(3), 387-403.
- Griffiths, D. V. & Fenton, G. A. (1997). Three-dimensional seepage through spatially random soil. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(2), 153-160. [https://doi.org/10.1061/\(asce\)1090-0241\(1997\)123:2\(153\)](https://doi.org/10.1061/(asce)1090-0241(1997)123:2(153))
- Hicks, M.A. (2014). Application of the Random Finite Element Method. *ALERT Doctoral School 2014 – Stochastic Analysis and Inverse Modelling*.
- Hicks, M. A., & Boughrarou, R. (1998). Finite element analysis of the Nerlerk underwater berm failures. *Géotechnique*, 48(2), 169-185. <https://doi.org/10.1680/geot.1998.48.2.169>
- Hicks, M. A., Varkey, D., van den Eijnden, A. P., de Gast, T. & Vardon, P. J. (2019). On characteristic values and the reliability-based assessment of Dykes. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 13(4), 313-319. <https://doi.org/10.1080/17499518.2019.1652918>
- Hicks M.A. & Spencer W.A. (2010). Influence of heterogeneity on the reliability and failure of a long 3D slope. *Computers and Geotechnics*, 37(7-8), 948-955. <https://doi.org/10.1016/j.compgeo.2010.08.001>
- Hicks, M. (2007). *Risk and variability in geotechnical engineering*. Thomas Telford.
- Hicks, M. A. & Onisiphorou, C. (2005). Stochastic evaluation of static liquefaction in a predominantly dilative sand fill. *Géotechnique*, 55(2), 123-133. <https://doi.org/10.1680/geot.55.2.123.59526>
- Jaksa M, Kaggwa W & Brooker P. (1999). Experimental evaluation of the scale of fluctuation of a stiff clay. *Proceedings of the 8th International Conference on the Application of Statistics and Probability in Civil Engineering, Sydney; 1999*, 415-422.

- Jaksa, M. B., Goldsworthy, J. S., Fenton, G. A., Kaggwa, W. S., Griffiths, D. V., Kuo, Y. L. & Poulos, H. G. (2005). Towards reliable and effective site investigations. *Géotechnique*, 55(2), 109–121. <https://doi.org/10.1680/geot.55.2.109.59531>
- van der Krogt, M. G., Schweckendiek, T. & Kok, M. (2019). Uncertainty in spatial average undrained shear strength with a site-specific transformation model. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards* 13(3), 181–190. <https://doi.org/10.1080/17499518.2018.1554820>
- Li, Y. J., Hicks, M. A. & Vardon, P. J. (2016). Uncertainty reduction and sampling efficiency in slope designs using 3D conditional random fields. *Computers and Geotechnics*, 79, 159–172. <https://doi.org/10.1016/j.compgeo.2016.05.027>
- Lloret-Cabot M, Hicks M.A, van den Eijnden A.P. (2012). Investigation of the reduction in uncertainty due to soil variability when conditioning a random field using kriging. *Géotechnique Letters*, 2(3), 123–127. <https://doi.org/10.1680/geolett.12.00022>
- Naghibi, F., Fenton, G. A. & Griffiths, D. V. (2016). Probabilistic considerations for the design of deep foundations against excessive differential settlement. *Canadian Geotechnical Journal*, 53(7), 1167–1175. <https://doi.org/10.1139/cgj-2015-0194>
- Nie, X., Zhang, J., Huang, H., Liu, Z. & Lacasse, S. (2015). Scale of fluctuation for geotechnical probabilistic analysis. *Proceedings of the 2nd International Conference on Structural Safety and Reliability*, 834–840.
- Phoon K-K, Kulhawy FH. (1999). Characterization of geotechnical variability. *Canadian Geotechnical Journal*, 36(4), 612–624. <https://doi.org/10.1139/t99-038>
- Popescu, R., Deodatis, G. & Nobahar, A. (2005). Effects of random heterogeneity of soil properties on bearing capacity. *Probabilistic Engineering Mechanics*, 20(4), 324–341. <https://doi.org/10.1016/j.probengmech.2005.06.003>
- Samy. K. (2003). *Stochastic analysis with finite elements in geotechnical engineering*. [PhD thesis, University of Manchester]. UoM repository. <https://www.research.manchester.ac.uk/portal/en/theses>
- Sert, S., Luo, Z., Xiao, J. H., Gong, W. P. & Juang, C. H. (2016). Probabilistic analysis of responses of cantilever wall-supported excavations in sands considering vertical spatial variability. *Computers and Geotechnics*, 75, 182–191. <https://doi.org/10.1016/j.compgeo.2016.02.004>
- Smith, I.M., Griffiths, D.V., Margetts, L. (2014). *Programming the Finite Element Method*. John Wiley & Sons.
- Spencer, W. A. (2007). *Parallel stochastic and finite element modelling of clay slope stability in 3D*. [PhD. thesis, University of Manchester, UK]. UoM repository. <https://www.research.manchester.ac.uk/portal/en/theses>
- Spencer, W. A. & Hicks, M. A. (2007). A 3D finite element study of slope reliability. *Numerical Models in Geomechanics*. <https://doi.org/10.1201/n0e0415440271.ch78>
- Student (1908). The probable error of a mean. *Biometrika*, 6(1), 1. <https://doi.org/10.2307/2331554>
- Suchomel, R. & Masín, D. (2010). Comparison of different probabilistic methods for predicting stability of a slope in spatially variable $c - \phi$ soil. *Computers and Geotechnics*, 37(1-2), 132–140. <https://doi.org/10.1016/j.compgeo.2009.08.005>

- Vanmarcke, E. H. (1977). Probabilistic modeling of soil profiles. *Journal of the Geotechnical Engineering Division*, 103(11), 1227–1246.
<https://doi.org/10.1061/ajgeb6.0000517>
- Vanmarcke, E. (1983). *Random fields, analysis and synthesis*. MIT Press.
- Varkey, D., Hicks, M., & Vardon, P. (2019). An improved semi-analytical method for 3D slope reliability assessments. *Computers and Geotechnics*, 111, 181–190.
<https://doi.org/10.1016/j.compgeo.2018.12.020>
- Varkey, D. (2020). *Geotechnical uncertainties and reliability-based assessments of dykes*. [PhD Thesis. Delft University of Technology, NL]. TU Delft repository.
<https://repository.tudelft.nl>
- Wang, C.-H., Osorio-Murillo, C. A., Zhu, H. & Rubin, Y. (2017). Bayesian approach for calibrating transformation model from spatially varied CPT data to regular geotechnical parameter. *Computers and Geotechnics*, 85, 262–273.
<https://doi.org/10.1016/j.compgeo.2017.01.002>
- Wong, S. Y. (2004). *Stochastic characterisation and reliability of saturated soils*. [PhD. thesis, University of Manchester, UK]. UoM repository.
<https://www.research.manchester.ac.uk/portal/en/theses>
- Yetbarek, E., Kumar, S., Ojha, R. (2020). Effects of soil heterogeneity on Subsurface Water Movement in agricultural fields: A numerical study. *Journal of Hydrology*, 590, 125420. <https://doi.org/10.1016/j.jhydrol.2020.125420>
- Zinkiewicz, O.C., Corneau, I.C. (1974). Visco-plasticity-plasticity and creep in elastic solids-a unified numerical solution approach. *International Journal for Numerical Methods in Engineering*, 8(4), 821–845.
<https://doi.org/10.1002/nme.1620080411>
- Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z. (2013). *The finite element method: Its basis and fundamentals*. Butterworth-Heinemann.



RANDOM FIELD GENERATION

```
1
2 # Importing libraries
3 import numpy as np
4 import scipy.spatial as sp
5 import sys
6 sys.path.append(r"C:/YourAddress/python/site-packages") # path for
   connection python-plaxis plugin
7 from plxscripting.easy import *
8 import imp
9 import os.path
10 import time
11
12 # %% Input and output initialization for connection with PLAXIS
13
14 localhostport`input = 10000
15 localhostport`output = 10001
16 plaxis`path=r 'C:/YourAddress/python/site-packages' # package for python
   3.7
17
18 s`i, g`i = new`server('localhost', localhostport`input, password='
   YourPLAXISPassword') # input server
19 s`o, g`o = new`server('localhost', localhostport`output, password='
   YourPLAXISPassword') # output server
20
21 # %% Open project
22
23 s`i.open('C:/YourAddress/YourProject.p2dx') # Master project
24
25 # %% Extraction of important geometrical data from textgenerated in
   PLAXIS
26
27 nip = np.loadtxt('C:/YourFileAddress/YourCoordinates.txt', usecols = 1,
   skiprows = 1, delimiter = 't') # Global integration points numbers
28 nlip = np.loadtxt('C:/YourFileAddress/YourCoordinates.txt', usecols = 2,
   skiprows = 1, delimiter = 't') # Local integration points numbers
29 X`coord = np.loadtxt('C:/YourFileAddress/YourCoordinates.txt', usecols =
   3, skiprows = 1, delimiter = 't') # X-coordinates
30 Y`coord = np.loadtxt('C:/YourFileAddress/YourCoordinates.txt', usecols =
   4, skiprows = 1, delimiter = 't') # Y-coordinates
31
32 # %% Manipulation
33
34 X`coord1 = np.reshape(X`coord, [-1, 1]) # reshape to explicit column
   vectors
35 Y`coord1 = np.reshape(Y`coord, [-1, 1])
36
37 ntip1 = len(X`coord) # total number of integration points from data
38
39 # %% RANDOM FIELD GENERATION
40
41 # no. of random fields - control from here number (and duration) of
   model runs      100 here is just an example
42
43 nfields = 100
44
45 # %% Point Statistics
46
47 thetax = 10 # horizontal scale of fluctuation
48 thetay = 10 # vertical scale of fluctuation
```

```

49 mean`c = 5
50 cov`c = 0.4 # coefficient of variation
51 std`ln`c = np.sqrt(np.log(1+cov`c**2)) # logarithmic standard deviation
52 mean`ln`c = np.log(mean`c) - 0.5 * std`ln`c**2 # logarithmic mean
53
54 mean`phi = 20
55 cov`phi = 0.4
56 std`ln`phi = np.sqrt(np.log(1+cov`phi**2)) # logarithmic standard
57 deviation
58 mean`ln`phi = np.log(mean`phi) - 0.5 * std`ln`phi**2 # logarithmic mean
59
60 # %% Correlations expressions
61
62 Dx = sp.distance`matrix(X`coord1,X`coord1) # distance matrix of x-coords
63 Dy = sp.distance`matrix(Y`coord1,Y`coord1) # distance matrix of y-
64 coords
65 corr = np.exp(-2 * np.sqrt((Dx/thetax)**2+(Dy/thetay)**2)) # Markovian
66 correlation in 2D
67 corr = np.linalg.cholesky(corr) # Cholesky factorization of correlation
68 matrix
69 # %% Creating random fields from standard normal
70 rdnl1 = corr @ np.random.normal(0,1, size = [len(X`coord),nfields]) #
71 create "nfields" random fields, for all integration points
72 rdnl2 = corr @ np.random.normal(0,1, size = [len(X`coord),nfields]) #
73 one for each varied parameter - becomes relevant if eventually cross-
74 correlations are considered
75
76 # %% Cross-correlation matrix (optional)
77
78 rho`cc = -0.5 # pearson cross-correlation between parameters
79 R = np.array([[1, rho`cc],[rho`cc,1]]) # cross-correlation matrix R
80 L = np.linalg.cholesky(R) # Cholesky factorization of R
81
82 # %% Pre-allocation of variables for cross-correlation of random fields
83
84 ct = np.zeros((len(X`coord),nfields)) # will become c after LN
85 transform
86 phit = np.zeros((len(X`coord),nfields)) # will become phi after
87 transform
88
89 # %% Obtain cross-correlated fields
90
91 for i in range(nfields):
92
93     rdnl = np.column`stack((rdnl1[:,i],rdnl2[:,i])) # 2D for 2 variables
94     , nD for n randomly varied fields
95
96     # Getting to the good shape for matmul
97     rdnl = np.transpose(rdnl)
98
99     # matmul with cholesky factorization of cross-correlation matrix
100     rdnl = L @ rdnl
101
102     # Back again to original shape
103     rdnl = np.transpose(rdnl)
104
105     #temp variables to be transformed to LN rfields in the next block
106     ct[:,i] = rdnl[:,0]
107     phit[:,i] = rdnl[:,1]
108
109 # %% Transforming to LN - cross correlated; for uncorrelated random
110 fields, perform this step right after the creation of rdnl1, rdnl2 -
111 the random fields from standard normal
112
113 c = np.exp(ct * std`ln`c + mean`ln`c)
114 phi = np.exp(phit * std`ln`phi + mean`ln`phi)
115
116
117

```

```

108 # %% Saving characteristics
109
110 file`name1 = 'ntip.dat' # The name you code in the UDSM for total no. of
      integration points
111 file`name2 = 'RandomFieldTest3.dat' # The name you code in the UDSM for
      the random fields file (for me the 3rd try was the lucky one)
112 file`name3 = 'SafetyFactors.txt' # Output of safety factors
113 file`name4 = 'Nf.txt' # Output of binary failed/not failed iterations
114 save`path1 = 'C:/YourAddress/RandomData/' # Path to folder from which
      UDSM reads
115 save`path2 = 'C:/YourAddress/PLAXISProjectFolder' # path to folder of
      the PLAXIS project
116 completeName1 = os.path.join(save`path1, file`name1) # the number of
      integration pts
117 completeName2 = os.path.join(save`path1, file`name2) # the random fields
      to be read by udsM
118 completeName3 = os.path.join(save`path2, file`name3) # the safety
      factors in proj folder
119 completeName4 = os.path.join(save`path2, file`name4) # the stand-or-fall
      results
120
121 # %% Saving number of integration points to file      to be read in UDSM
122
123 np.savetxt(completeName1, ntip1, fmt = '%1.0i')
124
125 # The random fields will be saved iteratively in the Monte Carlo loop in
      appendix C

```

A fully functional code combining the random field generation and the Monte Carlo analysis performed for the PLAXIS project desired can be obtained by clipping this appendix (appendix A) to appendix C. Both code pieces use the same variables, but are separated here into coding blocks to illustrate different conceptual purposes.

B | FORTRAN SUBROUTINE MODIFICATION (UDSM)

```

1  Implicit Double Precision (A-H, O-Z)
2      Double Precision , allocatable :: StVarTemp(:, :) ! new variable
3  declared
4      save StVarTemp ! will reuse variable after reading it in
5      Dimension Props(*), Sig0(*), StVar0(*), dEps(*), D(6,6),
6      *          Sig(*), StVar(*)
7  !
8  ! Local variables
9  !
10     Dimension DE(6,6), dSig(6), Prs'E(3), Prs(3),
11     *          xN1(3), xN2(3), xN3(3) ! StVarTemp(nStatV)
12
13     nStatV = 3 ! Modified; has to be equal to number of variables for
14     which RFs are created (here 3: c', phi and G)
15     !
16     If (IDTask .Eq. 4) Then ! Number of state parameters
17         nStat = nStatV
18         return
19     End If
20
21     If (IDTask .Eq. 1) Then ! Initialize state variables
22         Call MZeroR( StVar0, nStatV ) ! initializes StVar0 as a zero
23         vector of size nStatV
24         Call MZeroR( StVar , nStatV ) ! initializes StVar as a zero
25         vector of size nStatV
26
27         ! Debug
28         open (unit = 591, file = 'C:YourAddressRandomDataTestOutput1.
29         dat', action = 'write')
30         write(591, *) ((iEl-1)*12 +Int), iEl, Int
31
32         close(unit = 591)
33         ! Ending debug statement
34
35         If(ALLOCATED(StVarTemp) == .false.) then ! read only once
36             ! Reading number of integration points from file
37             !
38             open (unit = 592, file = 'C:YourAddressRandomDataNoIps.
39             dat', action = 'read')
40             read(592, *) n`rf ! number of integration points
41             allocate(StVarTemp(n`rf,2)) ! allocate to temporary
42             variable the size (no. of integration points, nStatV)
43             close(unit = 592)
44             ! Finished size allocation
45             ! Reading the actual random fields from file
46             !
47             open(unit = 593, file = 'C:YourAddressRandomDataYourRFs.
48             dat', action = 'read')
49             do i = 1, n`rf
50                 read(593,*) (StVarTemp(i,j), j = 1,3) ! for more variables
51                 j = 1, nStatV
52             end do
53             close(unit = 593)
54         end if
55
56         ! At this point the random field is read in
57         ! Extra debug to check if the file has been read properly
58         open (unit = 594, file = 'C:YourAddressRandomDataTestOutput2.
59         dat', action = 'write')

```

```

50     write(594, *) ((iEl-1)*12 + Int), StVarTemp((iEl-1)*12 + Int,1) !
51     ., StVarTemp((iEl-1)*12 + Int,2)
52     close(594)
53     ! Debug done - check external file "TestOutput2.dat"
54     !
55     ! Copying to the original State Variable vectors the values at
the correct global integration point (GIP)
56     !
57     StVar0(1) = StVarTemp((iEl-1)*12 + Int,1) ! Note: GIP = (Element
no. - 1) * (no.ips.per element) + (local integration point); Will
be c' at ips
58     StVar0(2) = StVarTemp((iEl-1)*12 + Int,2)
59     StVar0(3) = StVarTemp((iEl-1)*12 + Int,3)
60     StVar(1) = StVarTemp((iEl-1)*12 + Int,1) ! 15-noded elements as
defaults
61     StVar(2) = StVarTemp((iEl-1)*12 + Int,2) ! Will be Phi at ips
62     StVar(2) = StVarTemp((iEl-1)*12 + Int,3) ! Will be G at ips
63
64     C      = StVar(1)      ! C - Initialization
65     Phi    = StVar(2) / Rad ! Phi in radians
66     Psi    = Props(5) / Rad ! Psi in radians
67     sPhi   = Sin(Phi)
68     sPsi   = Sin(Psi)
69     cCosPhi = C*Cos(Phi)
70
71     Call WriVal( io, 'Phi ', phi )
72     Call WriVal( io, 'Psi ', psi )
73     Call WriVal( io, 'sPhi ', sphi )
74     Call WriVal( io, 'sPsi ', spsi )
75     Call WriVec( io, 'Props', Props, 10)
76     return
77     End If ! IDTask = 1
78
79 If (IDTask .Eq. 2) Then ! Calculate stresses
80 Call CopyRVec( StVar0, StVar, nStatV ) ! Copies values of StVar0 into
StVar
81     ipl    = 0
82     G      = StVar(3)      ! G
83     xNu    = Props(2)      ! nu
84     sTens  = Props(6)      ! tensile strength
85     C      = StVar(1) ! reading the value of c at the global
stress point from St.Var
86     Phi    = StVar(2) / Rad ! reading the value of phi at the
global stress point from St.Var
87     Psi    = Props(5) / Rad ! Psi in radians
88     sPhi   = Sin(Phi)
89     sPsi   = Sin(Psi)
90     cCosPhi = C*Cos(Phi)
91
92     If ( Abs(Sig0(15) - 1 ) < 1d-6 .And.
93     * Abs(Sig0(15) - 0 ) < 1d-6 ) Then
94     ParmSF = Sig0(15)
95     Rad    = 180d0 / Pi
96
97     Phi    = StVar(2) / Rad ! Phi in radians
98     Psi    = Props(5) / Rad ! Psi in radians
99
100    C      = StVar(1) / ParmSF ! C for SF calc strength
reduction
101    Phi    = ATan(Tan(Phi)/ParmSF ) ! Phi for SF calc strength
reduction
102    sTens  = Props(6)/ ParmSF ! tensile strength
103    Psi    = Min( Psi, Phi )
104
105    end if
106
107    sPhi   = Sin(Phi)
108    sPsi   = Sin(Psi)
109    cCosPhi = C*Cos(Phi)
110
111 End If

```

C

MONTE CARLO IN PLAXIS

These snippets of code highlight the syntax to be used when ordering PLAXIS to perform the calculation in a Monte Carlo framework. The variables used are linked with the random field generation ones utilized in the code in Appendix A. For convenience, the two appendices can be clipped together to give a plug-in-play code once the project is set up in PLAXIS. They have been split here to illustrate different conceptual points and to allow for in-text reference to separate appendices.

```
1
2 # %% Pre-allocation of output variables
3
4 sf = np.zeros((nfields,1)) # Safety factors pre-allocation
5 nf = np.zeros((nfields,1)) # Number of failed iterations
6 substring`check = 'The calculations of the following phases failed' #
   hedging for iterations that fail before full run
7
8
9 # %% Create PLAXIS objects to be used in the loop
10
11 Initial = g`i.InitialPhase
12 Null = g`i.Phase`1
13 Safety = g`i.Phase`2
14
15 # %% Loop project for all random fields
16
17 start = time.time() # to quantify durations of runs
18
19 for i in range(nfields):
20
21     c`phi = np.column`stack((c[:,i],phi[:,i]))
22
23     np.savetxt(completeName2, c`phi) # saving the two random fields to
   file
24
25     # Activate staged construction tab
26     g`i.gotostages()
27
28     # Mark for calculation
29     g`i.InitialPhase.ShouldCalculate = True
30     g`i.Phase`1.ShouldCalculate = True
31     g`i.Phase`2.ShouldCalculate = True
32
33
34
35     if substring`check in g`i.calculate(Initial): # if it fails in the
   initial phase
36         sf[i] = float(str(Initial.Reached.SumMstage))
37         nf[i] = 1
38
39     elif substring`check in g`i.calculate(Null):
40
41         sf[i] = float(str(Initial.Reached.SumMstage))
42         nf[i] = 1
43
44     else: # not adding nf here because if it reaches this phase, most
   likely sf != 1
45
46         g`i.calculate(Safety)
```

```

47     sf[i] = float(str(Safety.Reached.SumMsf))
48     nf[i] = 0
49
50 np.savetxt(completeName3, sf) # save all safety factors after whole run
    is performed
51 np.savetxt(completeName4, nf) # save all binary stand-or-fall results
    after whole run is performed
52
53 end = time.time()
54 zDuration = end - start # duration stored in a variable
55
56
57 #%% Post processing - calculating FoS CDF, Reliability and others
58
59 csf, binSF = np.histogram(sf) # safety factors histogram properties
60 pdf'sf = csf/sum(csf) # probability distribution function of safety
    factors
61 cdf'sf = np.cumsum(pdf'sf) # Cumulative distribution function of safety
    factors
62
63 #%% Computing probability of failure
64
65 Nf = np.sum(nf) # No. of iterations in which structure fails
66 Pf = Nf/nfields # Probability of failure
67
68 #%% Post processing - calculating CDF of SFs
69
70 csf, binSF = np.histogram(sf) # histogram of safety factors properties
71 pdf'sf = csf/sum(csf) # Probability distribution function of safety
    factors
72 cdf'sf = np.cumsum(pdf'sf) # Cumulative distribution function

```

D

AUTOMATION SCRIPT - FULL STARNMEER CASE CODE

```
1
2 # Fully-Automated Starnmeer Validation
3
4
5 import numpy as np
6 import scipy.spatial as sp
7 import sys
8 sys.path.append(r"C:/Users/Paul/Desktop/Needed`Software/python/site -
9     packages") # new path
10 from plxscripting.easy import *
11 import imp
12 import os.path
13 import time
14 import math as mt
15
16 # %% Input and output initialization
17 localhostport`input = 10000
18 localhostport`output = 10001
19 plaxis`path=r`C:/YourPathToPythonPackagesHere/python/site - packages`
20
21 s`i, g`i = new`server(`localhost`, localhostport`input, password=`
22     YourPasswordHere`)
23 s`o, g`o = new`server(`localhost`, localhostport`output, password=`
24     YourPasswordHere`) # opens output server
25
26 # %% New project to test geometry
27 s`i.new()
28
29 # %% Create soils for deterministic
30
31 g`i.gotosoil()
32 clay1 = g`i.soilmat()
33 peat2 = g`i.soilmat()
34 peat3 = g`i.soilmat()
35 clay4 = g`i.soilmat()
36 clay5 = g`i.soilmat()
37 sand6 = g`i.soilmat()
38
39 # %% Properties to be used in soil property definition
40
41 pi = mt.pi
42
43 # Mean values for soil declarations in deterministic phase
44
45 cc1 = 4.4
46 cp2 = 3.2
47 cp3 = 2.0
48 cc4 = 4.5
49 cc5 = 5.4
50 cs6 = 1e-5
51
52 phiclay1 = mt.atan(0.580)*180/pi
53 phipeat2 = mt.atan(0.398)*180/pi
54 phipeat3 = mt.atan(0.358)*180/pi
55 phiclay4 = mt.atan(0.559)*180/pi
56 phiclay5 = mt.atan(0.601)*180/pi
57 phisand6 = mt.atan(0.637)*180/pi
```

```

57
58 # %% Deterministic definition of soils ; "SoilModel", 2 = Mohr-Coulomb
59
60 g`i.gotosoil()
61
62 clay1.setproperties(("Identification", "Clay1"), ("SoilModel", 2), "
63     ("DrainageType", "Drained"), ("gammaUnsat", 6.9), ("
64     gammaSat", 13.9), "
65     ("Eref", 1e3), ("nu", 0.3), ("cref", cc1), ("phi",
66     phiclay1), ("psi", 0))
67     #Mohr-coulomb
68
69 peat2.setproperties(("Identification", "Peat2"), ("SoilModel", 2), "
70     ("DrainageType", "Drained"), ("gammaUnsat", 9.8), ("
71     gammaSat", 12), "
72     ("Eref", .2e3), ("nu", 0.3), ("cref", cp2), ("phi",
73     phipeat2), ("psi", 0))
74     #Mohr-coulomb
75
76 peat3.setproperties(("Identification", "Peat3"), ("SoilModel", 2), "
77     ("DrainageType", "Drained"), ("gammaUnsat", 9.8), ("
78     gammaSat", 12), "
79     ("Eref", .2e3), ("nu", 0.3), ("cref", cp3), ("phi",
80     phipeat3), ("psi", 0))
81     #Mohr-coulomb
82
83 clay4.setproperties(("Identification", "Clay4"), ("SoilModel", 2), "
84     ("DrainageType", "Drained"), ("gammaUnsat", 15), ("
85     gammaSat", 15), "
86     ("Eref", 1.5e3), ("nu", 0.3), ("cref", cc4), ("phi",
87     phiclay4), ("psi", 0))
88     #Mohr-coulomb
89
90 clay5.setproperties(("Identification", "Clay5"), ("SoilModel", 2), "
91     ("DrainageType", "Drained"), ("gammaUnsat", 15), ("
92     gammaSat", 15), "
93     ("Eref", 1.5e3), ("nu", 0.3), ("cref", cc5), ("phi",
94     phiclay5), ("psi", 0))
95     #Mohr-coulomb
96
97 sand6.setproperties(("Identification", "Sand6"), ("SoilModel", 2), "
98     ("DrainageType", "Drained"), ("gammaUnsat", 18), ("
99     gammaSat", 20), "
100     ("Eref", 25.0e3), ("nu", 0.3), ("cref", cs6), ("phi"
101     , phisand6), ("psi", 0))
102     #Mohr-coulomb
103
104 # %% Reading coordinates of geometry data
105
106 coords = np.loadtxt('C:/YourPath/YourGeometricalCoordinates.txt') #
107     Global ip no.
108
109 x = coords[:,1]
110 y = coords[:,2]
111
112 # %% Creating geometry of the problem
113
114 g`i.gotostructures()
115
116 # Layer 1 is Sand 6 (going bottom-up) - polygon 1 - index 0
117
118 g`i.polygon((x[38],y[38]), (x[39],y[39]), (x[40],y[40]), (x[34],y[34]),
119     (x[35],y[35]) "
120     , (x[36],y[36]), (x[37],y[37]), (x[42],y[42]), (x[41],y[41])
121     ) # Sand 6
122
123 # Clay 5 - polygon 2 - index 1
124
125 g`i.polygon((x[38],y[38]), (x[29],y[29]), (x[30],y[30]), (x[31],y[31]),
126     (x[28],y[28]) "

```

```

112     , (x[34],y[34]) , (x[40],y[40]) , (x[39],y[39])) # Clay 5
113
114 # Clay 4 - polygon 3 - index 2
115
116 g`i.polygon((x[34],y[34]) , (x[28],y[28]) , (x[18],y[18]) , (x[19],y[19]) ,
117     (x[32],y[32])“
118     , (x[37],y[37]) , (x[36],y[36]) , (x[35],y[35])) # Clay 4
119
120 # Peat 3a - polygon 4 - index 3
121 # Problema cu x la punctul 1 - 19.99 = 20 de fapt - x27
122 g`i.polygon((x[27],y[27]) , (x[23],y[23]) , (x[15],y[15]) , (x[16],y[16]) ,
123     (x[17],y[17])“
124     , (x[18],y[18])) # Peat 3a
125
126 # Peat 3b - same props - polygon 5 - index 4
127
128 g`i.polygon((x[19],y[19]) , (x[20],y[20]) , (x[21],y[21]) , (x[32],y[32]))#
129     Peat 3b
130
131 # Peat 2 - polygon 6 - index 5
132 #Problema cu x[22] - 19.98 e de fapt 20
133 g`i.polygon((x[29],y[29]) , (x[24],y[24]) , (x[25],y[25]) , (x[26],y[26]) ,
134     (x[22],y[22])“
135     , (x[27],y[27]) , (x[31],y[31]) , (x[30],y[30])) # Peat 2
136
137 # Clay 1b (same props as 1a) all under clay 1 - poly 7 - index 6
138
139 g`i.polygon((x[23],y[23]) , (x[13],y[13]) , (x[14],y[14]) , (x[15],y[15]))#
140     Clay 1b
141
142 # Clay 1a - Poly 8 - index 7
143
144 g`i.polygon((x[24],y[24]) ,(x[0],y[0]) ,(x[1],y[1]) ,(x[2],y[2]) ,“
145     (x[3],y[3]) ,(x[4],y[4]) ,(x[5],y[5]) ,(x[6],y[6]) ,“
146     (x[7],y[7]) ,(x[8],y[8]) ,(x[9],y[9]) ,(x[10],y[10]) ,“
147     (x[11],y[11]) ,(x[12],y[12]) ,(x[13],y[13]) ,(x[22],y
148     [22]) ,“
149     (x[26],y[26]) ,(x[25],y[25]) ,) # Clay 1a
150
151 # %% Attribute soils to polygons
152
153 g`i.setmaterial((g`i.polygons[0].Soil) , sand6) #sand6
154 g`i.setmaterial((g`i.polygons[1].Soil) , clay5) #clay5
155 g`i.setmaterial((g`i.polygons[2].Soil) , clay4) #clay4
156 g`i.setmaterial((g`i.polygons[3].Soil) , peat3) #peat3a
157 g`i.setmaterial((g`i.polygons[4].Soil) , peat3) #peat 3b
158 g`i.setmaterial((g`i.polygons[5].Soil) , peat2) #peat 2
159 g`i.setmaterial((g`i.polygons[6].Soil) , clay1) #clay1
160 g`i.setmaterial((g`i.polygons[7].Soil) , clay1) #clay1
161
162 # %% Meshing
163
164 g`i.gotomesh()
165 msize = 0.1 # relative element size (see PLAXIS user interface and
166     manual for meaning)
167 g`i.mesh(msize , True) # True for enhanced mesh refinement; False if not
168
169 # %% Flow - water level creation
170
171 g`i.gotoflow()
172
173 WL1 = g`i.waterlevel((x[50],y[50]) ,(x[54],y[54]) ,(x[44],y[44]) ,(x[45],y
174     [45]) ,“
175     (x[46],y[46]) ,(x[47],y[47]) ,(x[48],y[48]) ,(x[49],y[49]))
176     # Global - from geometry coordinates; can be defined independently
177
178 WL2 = g`i.waterlevel((x[50],y[50]) ,(x[54],y[54]) ,(x[51],y[51]) ,(x[52],y
179     [52]) ,(x[53],y[53])) # Water level corresponding to bottom sand
180
181 # %% Defining custom water lever for bottom sand
182

```

```

173 g`i`.set((g`i`.WaterConditions`1`1.Conditions),(g`i`.InitialPhase),"Custom
    Level") # Definition
174 g`i`.set((g`i`.WaterConditions`1`1.Level),(g`i`.InitialPhase),WL2) #
    Application
175
176 #%% Defining Phases - initial phase - BCs will remain in subsequent
    phases
177
178 g`i`.gotostages()
179 g`i`.InitialPhase.DeformCalcType = "Gravity loading"
180 Initial = g`i`.InitialPhase # Create object to handle later in MCS
181
182 #%% Activate soil polygons
183
184 for i in range(8):
185     g`i`.activate(g`i`.polygons[int(i)], g`i`.InitialPhase)
186
187 #%% Water Boundary conditions
188
189 g`i`.GroundwaterFlow.BoundaryXMin[g`i`.InitialPhase]= "Closed"
190 g`i`.GroundwaterFlow.BoundaryXMax[g`i`.InitialPhase]= "Closed"
191 g`i`.GroundwaterFlow.BoundaryYMin[g`i`.InitialPhase]= "Open"
192 g`i`.GroundwaterFlow.BoundaryYMax[g`i`.InitialPhase]= "Open"
193
194 #%% Null phase definition
195
196 Null = g`i`.phase(g`i`.InitialPhase)
197 Null.DeformCalcType = "Plastic"
198 Null.Deform.ResetDisplacementsToZero = True
199
200 Null.Identification = "Null"
201
202 #%% Safety phase definition
203
204 Safety = g`i`.phase(Null)
205 Safety.DeformCalcType = "Safety"
206 Safety.Deform.ResetDisplacementsToZero = True
207
208 Safety.Identification = "Safety"
209
210 #%% Calculate deterministically and extract the deterministic FoS and
    the coordinates of stress points
211
212 Initial.ShouldCalculate = True
213 Null.ShouldCalculate = True
214 Safety.ShouldCalculate = True
215
216 g`i`.calculate() # calculates all phases
217
218 sf`det` = float(str(Safety.Reached.SumMsf)) # extract safety factor from
    string
219
220 #%% Open safety phase and extract stress points coordinates
221
222 g`i`.view(Initial) # view output for initial phase
223
224 #%% Coordinate extraction as PLAXIS objects
225
226 Xini = g`o`.getresults(g`o`.ResultTypes.Soil.X, "stresspoint")
227 Yini = g`o`.getresults(g`o`.ResultTypes.Soil.Y, "stresspoint")
228 mat`id` = g`o`.getresults(g`o`.ResultTypes.Soil.MaterialID, "stresspoint")
    # extracts material IDs (numeric entity)
229
230 #%% Save coordinates and materials (needed to transform to numerical
    arrays
231
232 np.savetxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/TENTATIVE`Apps/
    Latest`Starnmeer.p2dxdat/X.txt" , Xini)
233 np.savetxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/TENTATIVE`Apps/
    Latest`Starnmeer.p2dxdat/Y.txt" , Yini)

```

```

234 np.savetxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/TENTATIVE`Apps/
      Latest`Starnmeer.p2dxdat/Z`mat`id.txt" , mat`id)
235
236 # %% Load as new arrays
237
238 X`coord = np.loadtxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/
      TENTATIVE`Apps/Latest`Starnmeer.p2dxdat/X.txt")
239 Y`coord = np.loadtxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/
      TENTATIVE`Apps/Latest`Starnmeer.p2dxdat/Y.txt")
240 mats = np.loadtxt("C:/Users/Paul/Desktop/Needed`Software/PLAXIS/
      TENTATIVE`Apps/Latest`Starnmeer.p2dxdat/Z`mat`id.txt")
241
242 # %% Manipulation into explicit 1D arrays
243
244 mats = np.reshape(mats,[-1,1])
245 X`coord = np.reshape(X`coord,[-1,1]) # reshape to column vectors of
      initially unknown size (the "-1" is for that)
246 Y`coord = np.reshape(Y`coord,[-1,1])
247
248 # %% close output
249
250 g`o.close()
251
252 # %% Create index vectors to separate soils on the basis of the material
      id
253
254
255 idx`c1 = np.transpose(np.where(mats == 1)) # Material IDs in the order
      of creation of soils
256 idx`p2 = np.transpose(np.where(mats == 2))
257 idx`p3 = np.transpose(np.where(mats == 3))
258 idx`c4 = np.transpose(np.where(mats == 4))
259 idx`c5 = np.transpose(np.where(mats == 5))
260 idx`s6 = np.transpose(np.where(mats == 6))
261
262 idx`c1 = np.reshape(idx`c1[: ,0],[-1,1])
263 idx`p2 = np.reshape(idx`p2[: ,0],[-1,1])
264 idx`p3 = np.reshape(idx`p3[: ,0],[-1,1])
265 idx`c4 = np.reshape(idx`c4[: ,0],[-1,1])
266 idx`c5 = np.reshape(idx`c5[: ,0],[-1,1])
267 idx`s6 = np.reshape(idx`s6[: ,0],[-1,1])
268
269
270 # %% Back to soil tab to define entities for RFEM simulation
271
272 g`i.gotosoil()
273
274 clay1R = g`i.soilmat()
275 peat2R = g`i.soilmat()
276 peat3R = g`i.soilmat()
277 clay4R = g`i.soilmat()
278 clay5R = g`i.soilmat()
279 sand6R = g`i.soilmat()
280
281 # %% Creating the right attributes - "16" corresponds to UDSM or "User-
      defined"
282 # NOTE: USER 1 = G; USER 2 = nu; USER 3 = c; USER 4 = phi, USER 5 = psi;
283 # The definition of c and phi is not crucial here because they are read
      from the random field
284
285 clay1R.setproperties(("Identification" , "Clay1R"),("DrainageType" , "
      Drained") , ("gammaUnsat" ,9.9) , "
286      ("gammaSat" ,13.9) , ("SoilModel" , "User - defined") , ("
      DllFile" , "a`c`phi64.dll") , "
287      ("ModelInDll" , "MC") , ("User1" ,1880) , ("User2" , 0.33)
      , ("User3" , 10) , ("User4" , 30) , "
288      ("User5" , 0) , ("EoedInter" ,5000) , ("CInter" ,1))
289
290 peat2R.setproperties(("Identification" , "Peat2R"),("DrainageType" , "
      Drained") , ("gammaUnsat" ,9.8) , "

```

```

291     ("gammaSat",11), ("SoilModel", "User-defined"), ("
    DllFile","a`c`phi64.dll"),“
292     ("ModelInDll","MC"), ("User1",940), ("User2", 0.33),
    ("User3", 10), ("User4", 30),“
293     ("User5", 0), ("EoedInter",2500),("CInter",1))
294
295 peat3R.setproperties(("Identification", "Peat3R"),("DrainageType",“
    Drained”), (“gammaUnsat”,9.9),“
296     (“gammaSat”,11), (“SoilModel”, “User-defined”), (“
    DllFile”,“a`c`phi64.dll”),“
297     (“ModelInDll”,“MC”), (“User1”,1111), (“User2”, 0.33)
    , (“User3”, 10), (“User4”, 30),“
298     (“User5”, 0), (“EoedInter”,3000),("CInter",1))
299
300 clay4R.setproperties(("Identification", "Clay4R"),("DrainageType",“
    Drained”), (“gammaUnsat”,15),“
301     (“gammaSat”,15), (“SoilModel”, “User-defined”), (“
    DllFile”,“a`c`phi64.dll”),“
302     (“ModelInDll”,“MC”), (“User1”,1880), (“User2”, 0.33)
    , (“User3”, 10), (“User4”, 30),“
303     (“User5”, 0), (“EoedInter”,5000),("CInter",1))
304
305 clay5R.setproperties(("Identification", "Clay5R"),("DrainageType",“
    Drained”), (“gammaUnsat”,15),“
306     (“gammaSat”,15), (“SoilModel”, “User-defined”), (“
    DllFile”,“a`c`phi64.dll”),“
307     (“ModelInDll”,“MC”), (“User1”,1880), (“User2”,
    0.33), (“User3”, 10), (“User4”, 30),“
308     (“User5”, 0), (“EoedInter”,5000),("CInter",1))
309
310 sand6R.setproperties(("Identification", "Sand6R"),("DrainageType",“
    Drained”), (“gammaUnsat”,18),“
311     (“gammaSat”,20), (“SoilModel”, “User-defined”), (“
    DllFile”,“a`c`phi64.dll”),“
312     (“ModelInDll”,“MC”), (“User1”,16920), (“User2”,
    0.33), (“User3”, 10), (“User4”, 30),“
313     (“User5”, 0), (“EoedInter”,45000),("CInter",1))
314
315
316 #%% Assign correct soils to polygons - indexes in g`i.polygons are
    correspondent to their order of creation
317
318 g`i.gotostructures()
319
320 g`i.setmaterial((g`i.polygons[0].Soil), sand6R) #sand6
321 g`i.setmaterial((g`i.polygons[1].Soil), clay5R) #clay5
322 g`i.setmaterial((g`i.polygons[2].Soil), clay4R) #clay4
323 g`i.setmaterial((g`i.polygons[3].Soil), peat3R) #peat3a
324 g`i.setmaterial((g`i.polygons[4].Soil), peat3R) #peat 3b
325 g`i.setmaterial((g`i.polygons[5].Soil), peat2R) #peat 2
326 g`i.setmaterial((g`i.polygons[6].Soil), clay1R) #clay1
327 g`i.setmaterial((g`i.polygons[7].Soil), clay1R) #clay1
328
329 #%% Check flow
330
331 g`i.gotoflow()
332
333 #%% Go back to stages
334
335 g`i.gotostages()
336
337 #%% Manipulation of coordinates and start of block for creation of
    separated RFs
338
339 #%% Extract coordinates for each soil in order to create the random
    fields and distance matrices
340
341 # First pre-allocation of coordinate vectors
342
343 X`coord`c1 = np.array([])# pre-allocation of unknown size array
344 X`coord`c4 = np.array([])# pre-allocation of unknown size array

```

```

345 X'coord'c5 = np.array([])# pre-allocation of unknown size array
346 X'coord'p2 = np.array([])# pre-allocation of unknown size array
347 X'coord'p3 = np.array([])# pre-allocation of unknown size array
348 X'coord's6 = np.array([])# pre-allocation of unknown size array
349
350 Y'coord'c1 = np.array([])# pre-allocation of unknown size array
351 Y'coord'c4 = np.array([])# pre-allocation of unknown size array
352 Y'coord'c5 = np.array([])# pre-allocation of unknown size array
353 Y'coord'p2 = np.array([])# pre-allocation of unknown size array
354 Y'coord'p3 = np.array([])# pre-allocation of unknown size array
355 Y'coord's6 = np.array([])# pre-allocation of unknown size array
356
357 #%% Coordinates separation
358
359 for i in idx'c1:
360
361     X'coord'c1 = np.reshape(np.append(X'coord'c1,X'coord[i]),[-1,1])
362     Y'coord'c1 = np.reshape(np.append(Y'coord'c1,Y'coord[i]),[-1,1])
363
364 for i in idx'p2:
365
366     X'coord'p2 = np.reshape(np.append(X'coord'p2,X'coord[i]),[-1,1])
367     Y'coord'p2 = np.reshape(np.append(Y'coord'p2,Y'coord[i]),[-1,1])
368
369 for i in idx'p3:
370
371     X'coord'p3 = np.reshape(np.append(X'coord'p3,X'coord[i]),[-1,1])
372     Y'coord'p3 = np.reshape(np.append(Y'coord'p3,Y'coord[i]),[-1,1])
373
374
375 for i in idx'c4:
376
377     X'coord'c4 = np.reshape(np.append(X'coord'c4,X'coord[i]),[-1,1])
378     Y'coord'c4 = np.reshape(np.append(Y'coord'c4,Y'coord[i]),[-1,1])
379
380 for i in idx'c5:
381
382     X'coord'c5 = np.reshape(np.append(X'coord'c5,X'coord[i]),[-1,1])
383     Y'coord'c5 = np.reshape(np.append(Y'coord'c5,Y'coord[i]),[-1,1])
384
385
386 for i in idx's6:
387
388     X'coord's6 = np.reshape(np.append(X'coord's6,X'coord[i]),[-1,1])
389     Y'coord's6 = np.reshape(np.append(Y'coord's6,Y'coord[i]),[-1,1])
390
391 #%% Point statistics declaration and small manipulation to LN
392
393 nfields = 100 # no of random fields = no of monte carlo iterations
394
395 # Scales of fluctuation in X - Y
396 thetax = 0.5
397 thetay = 0.5
398
399 # mean cohesions
400
401 mc'c1 = 4.4
402 mc'p2 = 3.2
403 mc'p3 = 2
404 mc'c4 = 4.5
405 mc'c5 = 5.4
406 mc's6 = 10e-5
407
408 # coefficient of variation - cohesion
409
410 vc'c1 = .773
411 vc'p2 = .656
412 vc'p3 = .775
413 vc'c4 = .554
414 vc'c5 = .352
415 vc's6 = .0

```

```

416
417 # mean phi
418
419 mp`c1 = 30.11
420 mp`p2 = 19.84
421 mp`p3 = 19.70
422 mp`c4 = 29.20
423 mp`c5 = 31.00
424 mp`s6 = 32.49
425
426 # coefficients of variation - phi
427
428 vp`c1 = .081
429 vp`p2 = .058
430 vp`p3 = .145
431 vp`c4 = .012
432 vp`c5 = .007
433 vp`s6 = .000
434
435
436 # STD LN version - cohesions
437
438 slnc`c1 = np.sqrt(np.log(1+vc`c1**2))
439 slnc`p2 = np.sqrt(np.log(1+vc`p2**2))
440 slnc`p3 = np.sqrt(np.log(1+vc`p3**2))
441 slnc`c4 = np.sqrt(np.log(1+vc`c4**2))
442 slnc`c5 = np.sqrt(np.log(1+vc`c5**2))
443 slnc`s6 = np.sqrt(np.log(1+vc`s6**2))
444
445
446 # STD LN version - phi
447
448 slnp`c1 = np.sqrt(np.log(1+vp`c1**2))
449 slnp`p2 = np.sqrt(np.log(1+vp`p2**2))
450 slnp`p3 = np.sqrt(np.log(1+vp`p3**2))
451 slnp`c4 = np.sqrt(np.log(1+vp`c4**2))
452 slnp`c5 = np.sqrt(np.log(1+vp`c5**2))
453 slnp`s6 = np.sqrt(np.log(1+vp`s6**2))
454
455
456 # mean LN version - cohesions
457
458 mlnc`c1 = np.log(mc`c1) - 0.5 * slnc`c1**2
459 mlnc`p2 = np.log(mc`p2) - 0.5 * slnc`p2**2
460 mlnc`p3 = np.log(mc`p3) - 0.5 * slnc`p3**2
461 mlnc`c4 = np.log(mc`c4) - 0.5 * slnc`c5**2
462 mlnc`c5 = np.log(mc`c5) - 0.5 * slnc`c5**2
463 mlnc`s6 = np.log(mc`s6) - 0.5 * slnc`s6**2
464
465 # mean LN version - phi
466
467 mlnp`c1 = np.log(mp`c1) - 0.5 * slnp`c1**2
468 mlnp`p2 = np.log(mp`p2) - 0.5 * slnp`p2**2
469 mlnp`p3 = np.log(mp`p3) - 0.5 * slnp`p3**2
470 mlnp`c4 = np.log(mp`c4) - 0.5 * slnp`c4**2
471 mlnp`c5 = np.log(mp`c5) - 0.5 * slnp`c5**2
472 mlnp`s6 = np.log(mp`s6) - 0.5 * slnp`s6**2
473
474 # %% Full distance matrices
475
476 Dx = sp.distance`matrix(X`coord, X`coord) # distances between x-
      coordinates in matrix
477 Dy = sp.distance`matrix(Y`coord, Y`coord) # distances between x-
      coordinates in matrix
478
479 # %% Correlation expressions and random vectors creation on the basis of
      a standard-normal distribution
480
481 corr = np.exp(-2 * np.sqrt((Dx/thetax)**2+(Dy/thetay)**2)) # Markovian
      correlation in 2D
482 corr = np.linalg.cholesky(corr)

```

```

483
484
485 rdn11 = corr @ np.random.normal (0,1, size = [len(X`coord), nfields])
486 rdn12 = corr @ np.random.normal (0,1, size = [len(X`coord), nfields])
487
488 # %% Pre-allocation of c and phi fields
489
490 c = np.zeros((len(X`coord), nfields))
491 phi = np.zeros((len(X`coord), nfields))
492
493 # %% Attribute transformations to correct areas on the basis of their (
    LN) point stats
494
495 for i in idx`c1:
496
497     c[i,:] = np.exp(rdn11[i,:] * slnc`c1 + mlnc`c1)
498     phi[i,:] = np.exp(rdn12[i,:] * slnp`c1 + mlnp`c1)
499
500 for i in idx`p2:
501
502     c[i,:] = np.exp(rdn11[i,:] * slnc`p2 + mlnc`p2)
503     phi[i,:] = np.exp(rdn12[i,:] * slnp`p2 + mlnp`p2)
504
505 for i in idx`p3:
506
507     c[i,:] = np.exp(rdn11[i,:] * slnc`p3 + mlnc`p3)
508     phi[i,:] = np.exp(rdn12[i,:] * slnp`p3 + mlnp`p3)
509
510 for i in idx`c4:
511
512     c[i,:] = np.exp(rdn11[i,:] * slnc`c4 + mlnc`c4)
513     phi[i,:] = np.exp(rdn12[i,:] * slnp`c4 + mlnp`c4)
514
515 for i in idx`c5:
516
517     c[i,:] = np.exp(rdn11[i,:] * slnc`c5 + mlnc`c5)
518     phi[i,:] = np.exp(rdn12[i,:] * slnp`c5 + mlnp`c5)
519
520 for i in idx`s6:
521
522     c[i,:] = np.exp(rdn11[i,:] * slnc`s6 + mlnc`s6)
523     phi[i,:] = np.exp(rdn12[i,:] * slnp`s6 + mlnp`s6)
524
525 # %% Check PDF, CDF of RF properties - c / phi
526
527 crf1, binrf1 = np.histogram(c, bins = 5) # histogram properties
528 pdf`rf1 = crf1/sum(crf1) # Probability distribution function of safety
    factors
529 cdf`rf1 = np.cumsum(pdf`rf1) # Cumulative distribution function
530
531 import matplotlib.pyplot as plt
532
533 plt.plot(binrf1[1:], pdf`rf1, color = 'red', label = 'PDF')
534 plt.plot(binrf1[1:], cdf`rf1, color = 'blue', label = 'CDF')
535 plt.legend()
536
537
538 # %% Saving characteristics
539
540 file`name1 = 'ntip.dat'
541 file`name2 = 'RandomFieldTest3.dat' # !! KEEP NAME THE SAME - IT'S CODED
    IN THE UDSM
542 file`name3 = 'SafetyFactors.dat'
543 file`name4 = 'Nf.dat'
544 save`path1 = 'C:/YourPathToUDSMReading' # Path to folder from which UDSM
    reads
545 save`path2 = 'C:/YourPathToThePLAXISProject/YourProject.p2dxdat/' # path
    to folder of the PLAXIS project
546
547 IntegrationPoints = os.path.join(save`path1, file`name1) # the number of
    integration pts

```

```

548 RandomFields = os.path.join(save_path1, file_name2) # the random fields
      to be read by udsM
549 SafetyFactors = os.path.join(save_path2, file_name3) # the safety
      factors in proj folder
550 BinaryFailure = os.path.join(save_path2, file_name4) # Binary failure
      performance
551
552 # %% Plot to check RFs
553
554 import matplotlib.pyplot as plt
555 #from mpl_toolkits import mplot3d
556
557 fig, ax = plt.subplots()
558
559 scatter = ax.scatter(X_coord, Y_coord, c = c[:,0]) # first random field
560 ax.set(xlim = [-15,40]) # Your bound-limits here if needed
561 ax.set(ylim = [-15,40])
562 legend1 = ax.legend(*scatter.legend_elements(num = 6), loc = "upper
      right", title = "c")
563 plt.savefig("RandomFieldCheck.jpg", format = 'jpg', dpi = 500)
564
565 # %% Save no of integration points
566
567 ntip1 = np.reshape(np.array([len(X_coord)]),[-1,1])
568 np.savetxt(IntegrationPoints, ntip1, fmt = '%1.0i')
569
570 # %% Pre-allocation of output variables and substring check for failure
      in first 2 phases
571
572 sf = np.zeros((nfields,1)) # Safety factors pre-allocation
573 nf = np.zeros((nfields,1)) # Number of failed iterations
574 substring_check = 'The calculations of the following phases failed'
575
576 # %% Calculate
577
578 start = time.time()
579
580 for i in range(nfields):
581
582     c_phi = np.column_stack((c[:, i], phi[:, i]))
583
584     np.savetxt(RandomFields, c_phi) # saving the two random fields to
      file
585     # Mark for calculation
586     Initial.ShouldCalculate = True
587     Null.ShouldCalculate = True
588     Safety.ShouldCalculate = True
589
590
591     if substring_check in g`i.calculate(Initial): # if issues in initial
      phase
592         sf[i] = float(str(Initial.Reached.SumMstage))
593         nf[i] = 1
594
595     elif substring_check in g`i.calculate(Null):
596         sf[i] = 0.99
597         nf[i] = 1
598
599     else:
600         g`i.calculate(Safety)
601         sf[i] = float(str(Safety.Reached.SumMsf))
602         nf[i] = 0
603
604 np.savetxt(SafetyFactors, sf) # Save to file - hedge for crash + external
      processing
605 np.savetxt(BinaryFailure, nf) # Save to file - hedge for crash + external
      processing
606
607 end = time.time()
608 zDuration = end - start # duration stored in a variable

```

E | SCALE OF FLUCTUATION, ρ AND MESH CHOICES

In this appendix, illustrations are provided to highlight the role of input scales of fluctuation on the correlation values between an origin point and its surroundings, for a given correlation structure. This is done by choosing an isotropic scale of fluctuation ($\theta_x = \theta_y$) with two different values ($\theta_x = \theta_y = 2, 10 m$).

3D visualization of Markovian correlation as a function of θ

Markovian correlation function - $\Theta_x = \Theta_y = 2m$

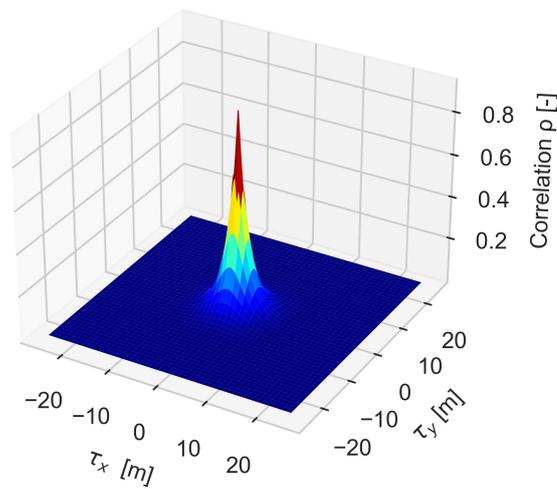


Figure E.1: 3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 2 m, \theta_y = 2 m$

Markovian correlation function - $\Theta_x = 10m, \Theta_y = 10m$

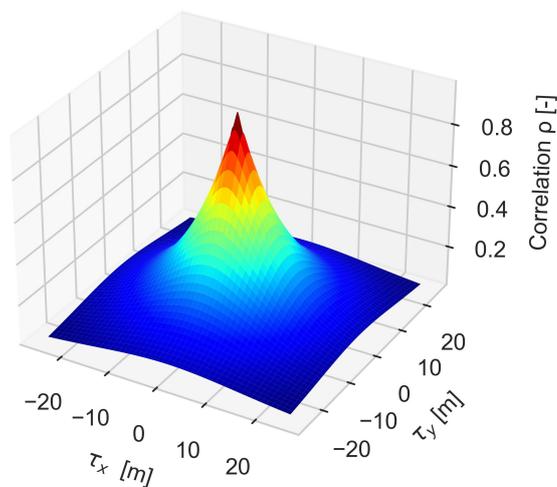


Figure E.2: 3D visualization of correlation as a function of τ_x, τ_y , for $\theta_x = 10 m, \theta_y = 10 m$

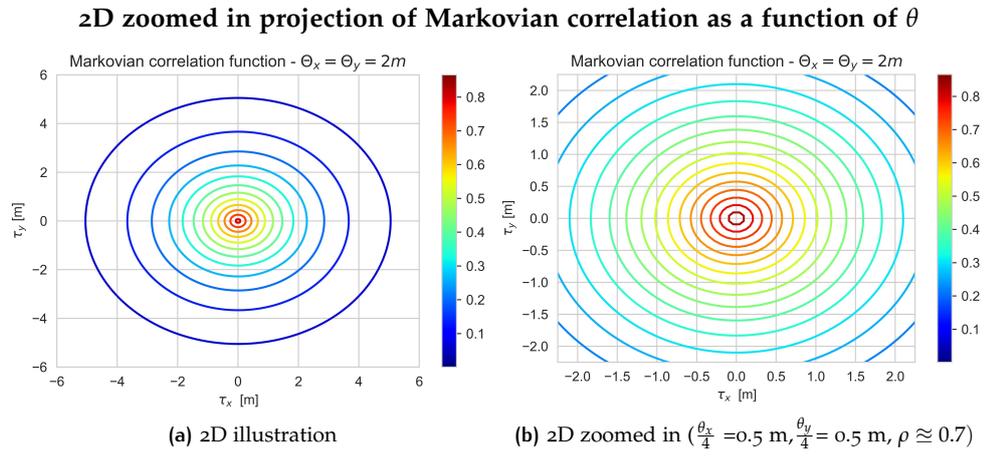


Figure E.3: Parameter Markovian correlation (ρ) as a function of an isotropic spatial correlation structure, $\theta_y = 2\text{ m}$ $\theta_x = 2\text{ m}$

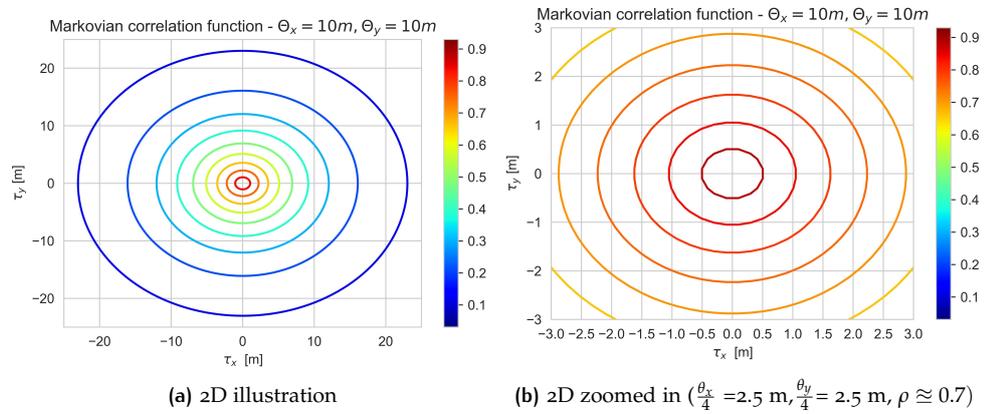


Figure E.4: Parameter Markovian correlation (ρ) as a function of an isotropic spatial correlation structure, $\theta_y = 10\text{ m}$ $\theta_x = 10\text{ m}$

F

CDF OF SF_S - VERIFICATION CASE 2

This appendix provides an illustration of the CDFs of safety factors for different scales of fluctuation, along with the mean, median and deterministic solutions. This is to prove that with a very large scale of fluctuation, the distribution of results should resemble that of a set of deterministic analyses, which for a combination of mean values of parameters should yield a value close to the deterministic safety factor obtained with those same mean values.

Moreover, results confirm the trend observed in literature, that with smaller values of spatial correlation, the distribution of solution tends to be more narrowly spread around the mean solution.

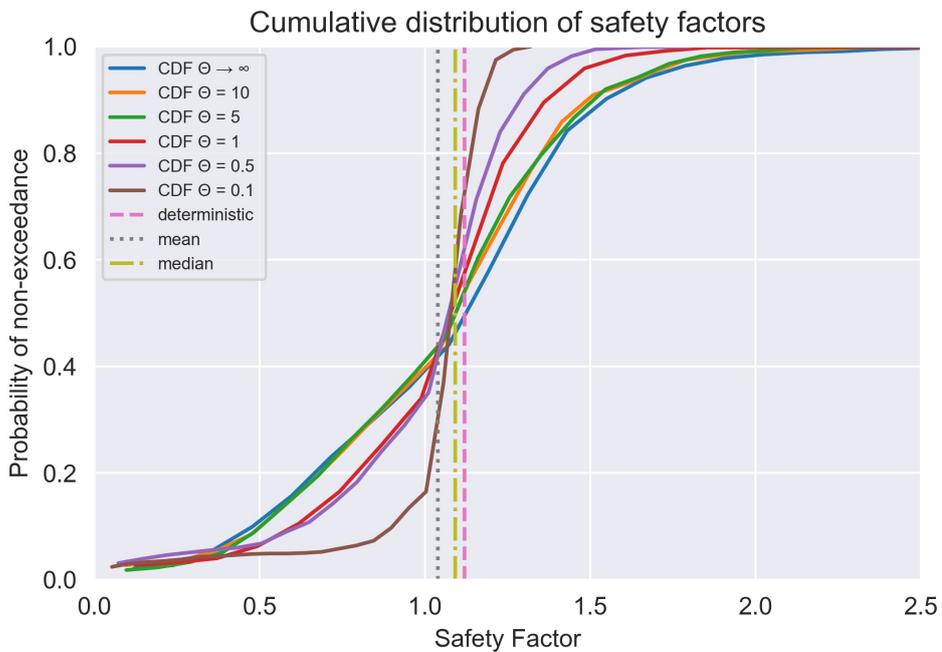


Figure F.1: Cumulative distribution functions and deterministic solutions for safety factors in the 2nd verification case study

G

RANDOM FIELDS AS A FUNCTION OF Θ - VERIFICATION CASE 2

This appendix provides a series of comparative plots of random fields distributions along a problem's geometry for different scales of fluctuation. The original study used a scale of fluctuation normalized to the height of the slope - $\Theta = \frac{\theta}{H}$, so results here are expressed in terms of this entity. This serves to illustrate the effect the scale of fluctuation has on the distribution of a parameters' values throughout the domain, while reinforcing the verification of proper reading and storage of the random field in PLAXIS' kernel. To avoid redundancy, only the cohesion random field is used in this case, but the principle holds for the other random fields as well. Illustrations can be consulted in figures G.1 to G.5. It is worth noting the gradual increase in continuity of zones of a certain value, for larger values of scale of fluctuation. In this sense, it is also important to note the scale of values PLAXIS provides when illustrating the values of parameters.

Distribution of cohesion values over the domain for $\Theta = 0.1$

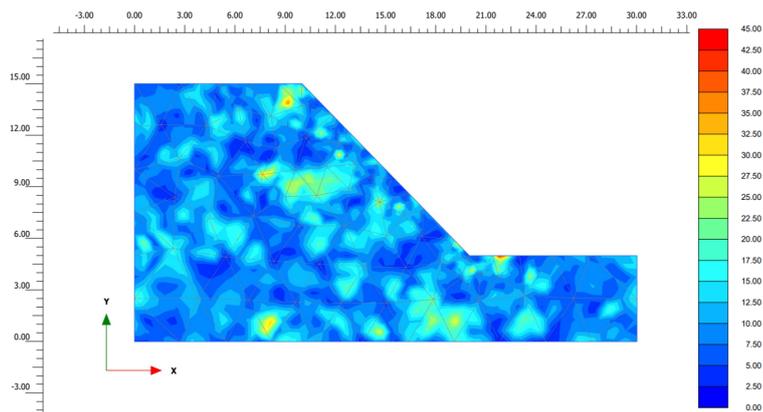


Figure G.1: Random field for c' [kPa] - Verification case 2 - $\Theta = 0.1$ ($\theta_v = \theta_h = 1 m$)

Distribution of cohesion values over the domain for $\Theta = 0.5$

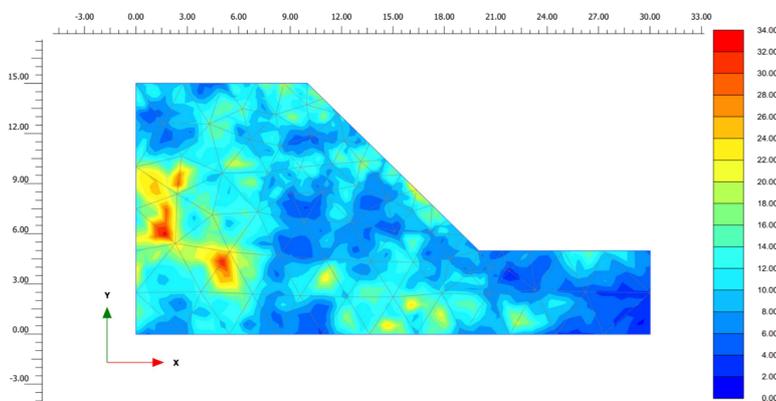


Figure G.2: Random field for c' [kPa] - Verification case 2 - $\Theta = 0.5$ ($\theta_v = \theta_h = 5 m$)

Distribution of cohesion values over the domain for $\Theta = 1$

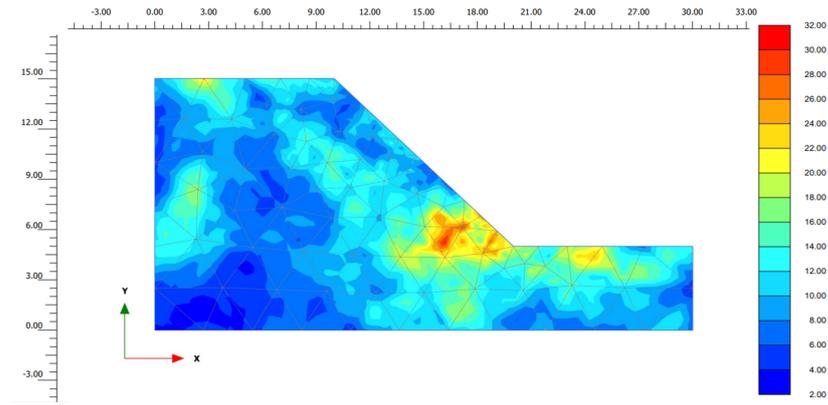


Figure G.3: Random field for c' [kPa] - Verification case 2 - $\Theta = 1$, ($\theta_v = \theta_h = 10$ m)

Distribution of cohesion values over the domain for $\Theta = 5$

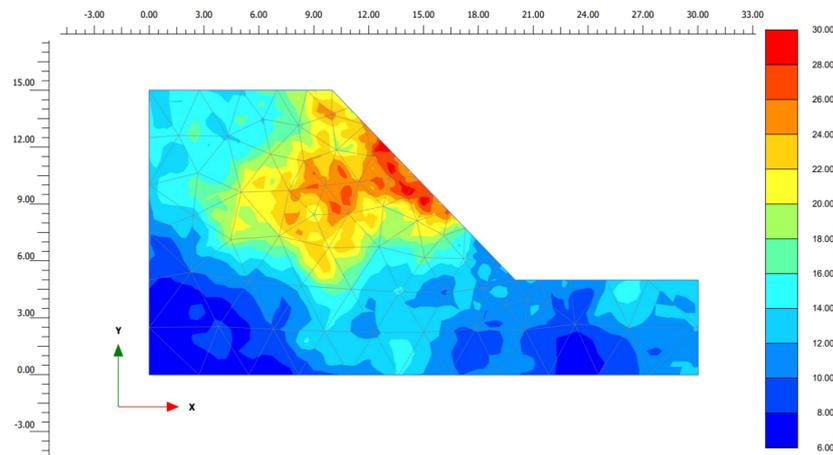


Figure G.4: Random field for c' [kPa] - Verification case 2 - $\Theta = 5$, ($\theta_v = \theta_h = 50$ m)

Distribution of cohesion values over the domain for $\Theta \rightarrow \infty$

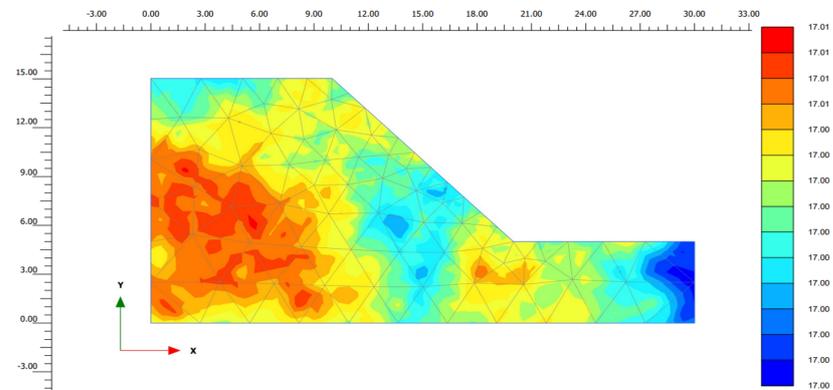


Figure G.5: Random field for c' [kPa] - Verification case 2 - $\Theta \rightarrow \infty$

H

STRESS POINTS SPACING - VALIDATION CASE STUDY

This appendix provides an illustration from a PLAXIS connectivity plot showcasing both the nodes and the stress points' distribution at a remote part of the domain. The nodes are depicted in red, while the stress points are shown with purple dots.

This serves to prove that the distances between two integration points in this zone remote from the failure area is not larger than 0.5 m, which is the smallest scale of fluctuation taken into account in this study. As a result, this can be captured by the mesh in the scenarios considered. That being said, as mentioned in the main text there is a small number of elements for which this spacing is not achieved and larger elements are generated.

As noted before, this was deemed acceptable due to the rarity of occurrence and the very large number of stress points potentially needed to comply with this requisite all over the domain. Increasing the stress point density until meeting this spacing criterion in the remote areas of the domain would make the duration of the calculation prohibitive for the purpose of this study.

Detailed view of the largest elements in the mesh

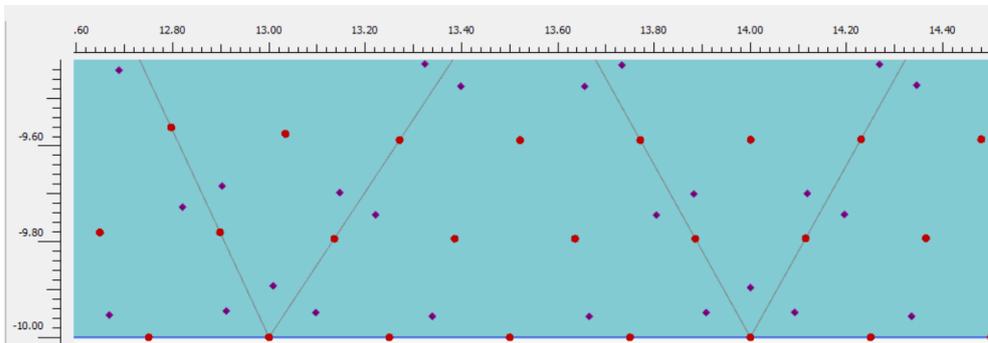


Figure H.1: Spacing between integration points in the sand layer - Starnmeer validation case (Red points = nodes; Purple points = stress points)

I

RANDOM FIELDS AS A FUNCTION OF Θ - VALIDATION CASE STUDY

Distribution of cohesion values over the domain for $\theta_v = 0.5 m$, $\theta_h = 6 m$

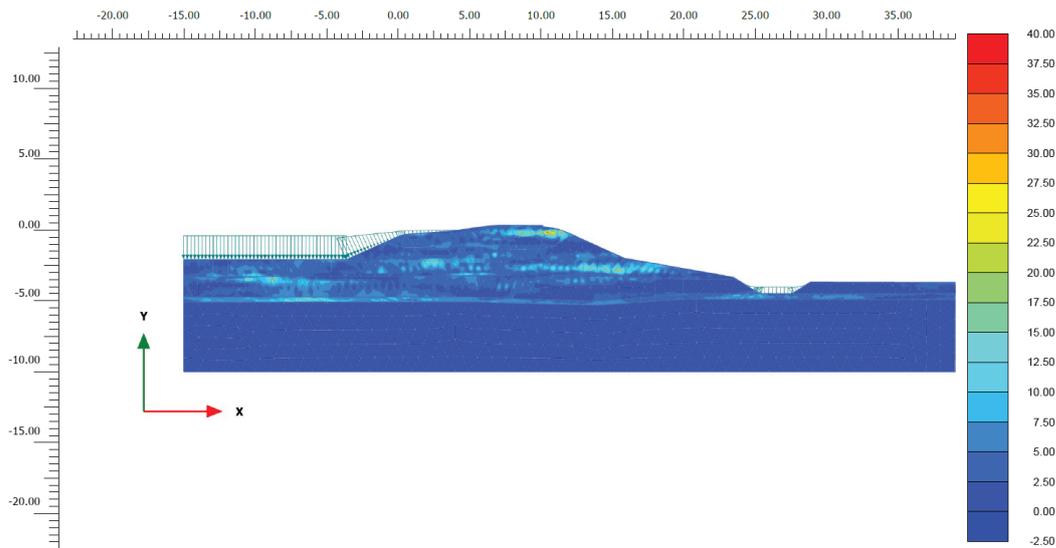


Figure I.1: Starnmeer validation case - random field for $c' [kPa]$, $\theta_v = 0.5 m$, $\theta_h = 6 m$

Distribution of friction angle values over the domain for $\theta_v = 0.5 m$, $\theta_h = 6 m$

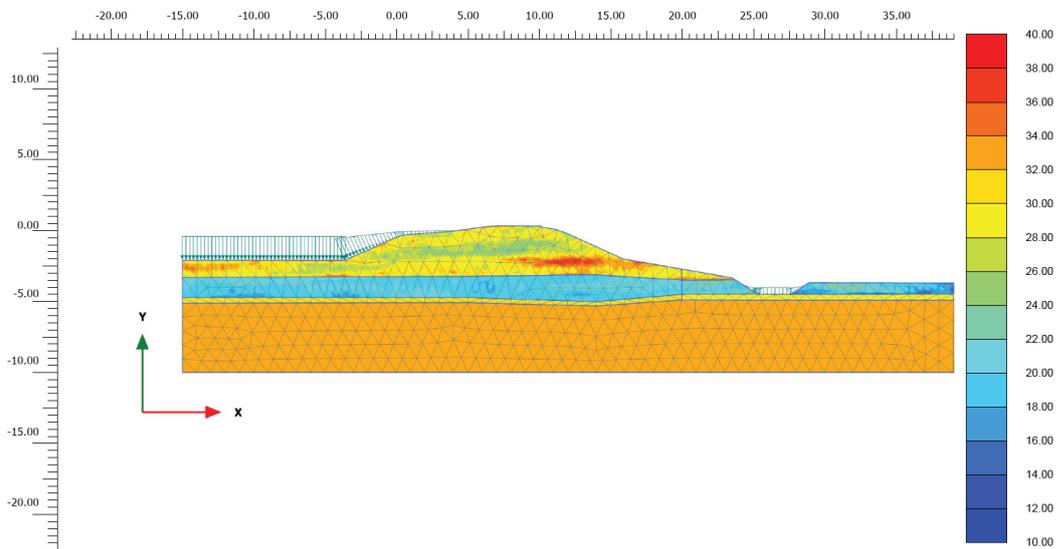


Figure I.2: Starnmeer validation case - random field for $\phi' [^\circ]$, $\theta_v = 0.5 m$, $\theta_h = 6 m$

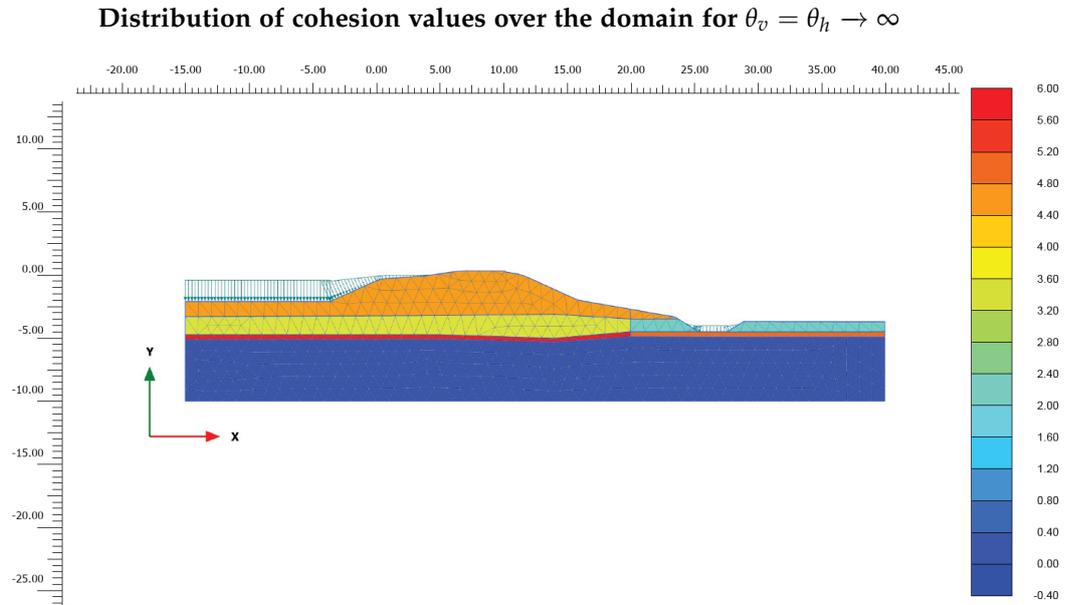


Figure I.3: Starnmeer validation case - random field for c' [kPa], $\theta_v = \theta_h \rightarrow \infty$

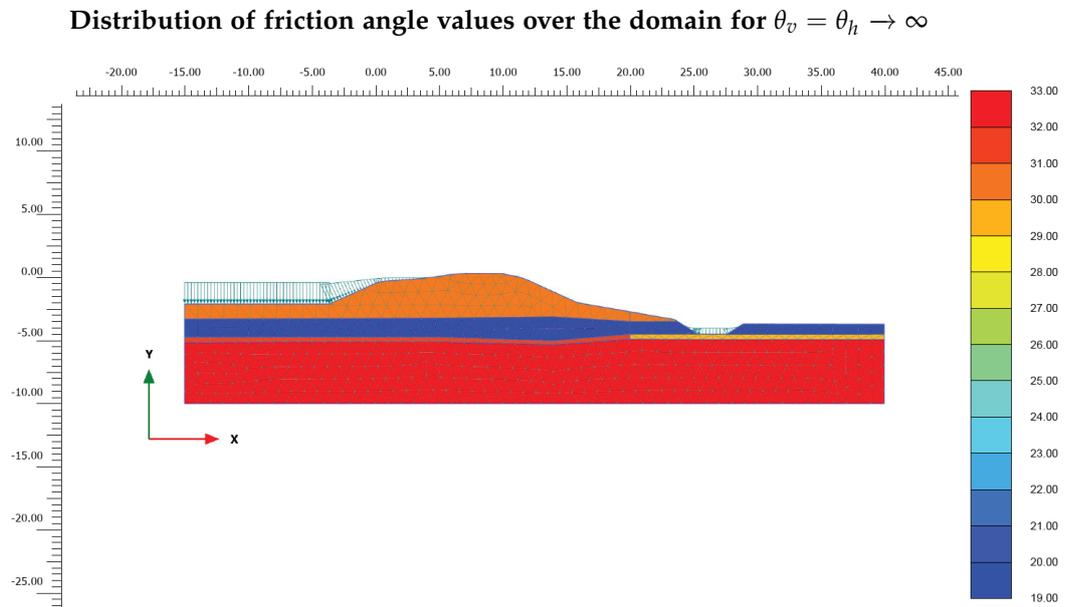
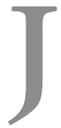


Figure I.4: Starnmeer validation case - random field for ϕ' [°], $\theta_v = \theta_h \rightarrow \infty$



SEPARATE COMPARISONS - CDF_S OF SF_S - STARNMEER VALIDATION CASE

This appendix provides illustrations of the results presented in figure 5.6 of chapter 5. To highlight the point around the meaning of "1" as a threshold safety factor value, the parts of the curve below this value are plotted with a different color.

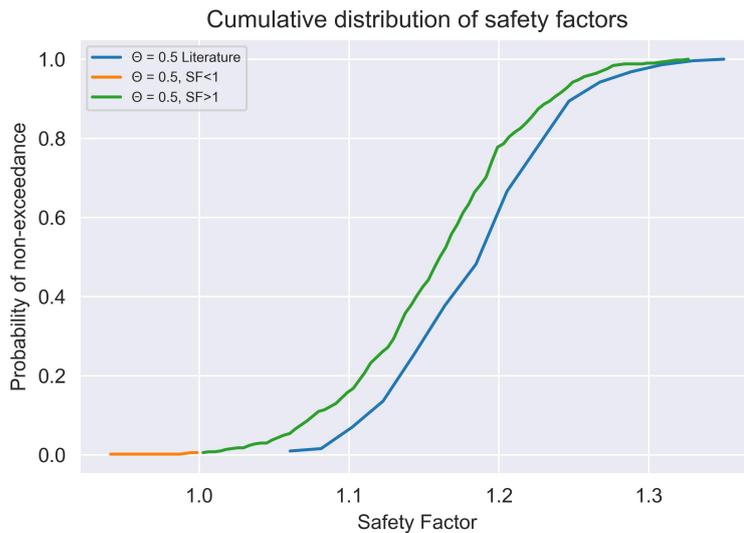


Figure J.1: Comparison of CDFs of safety factors for the Starnmeer validation case:
 $\theta_v = \theta_h = 0.5 m$

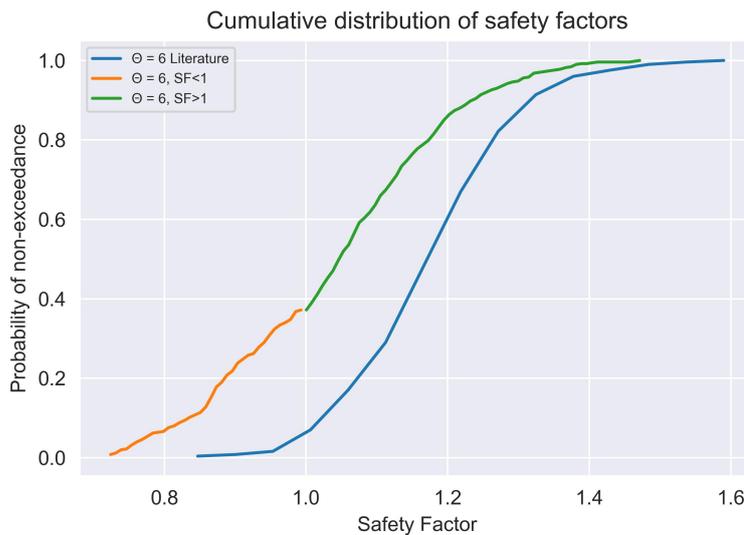


Figure J.2: Comparison of CDFs of safety factors for the Starnmeer validation case:
 $\theta_v = 0.5 m \theta_h = 6 m$

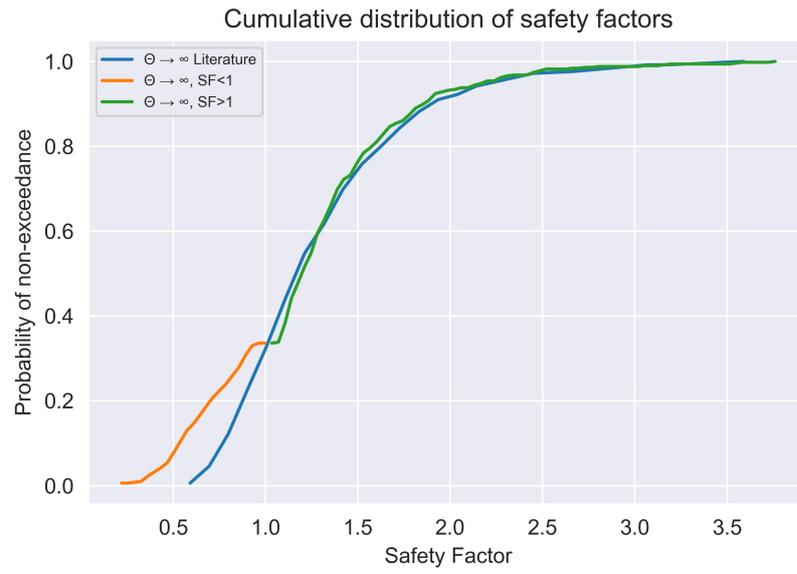


Figure J.3: Comparison of CDFs of safety factors for the Starnmeer validation case:
 $\theta_v = \theta_h \rightarrow \infty$

K

DEMONSTRATIVE CASE STUDY: DETERMINISTIC ANALYSIS

This appendix provides an illustration of the failure mechanism for the deterministic calculation in the demonstrative case study. In figure K.1, incremental displacements are plotted as a result of a "Safety" analysis performed with the mean parameters reported in table 6.1.

The failure mechanism identified in this deterministic analysis is only local, not extending to the bottom two layers. This makes the procedure of varying property values significantly only in the top layer reasonable. If the failure mechanism would have extended to the bottom two layers, this assumption would not be valid. As a result of this undertaking, a safety factor value of 2.5 was calculated.

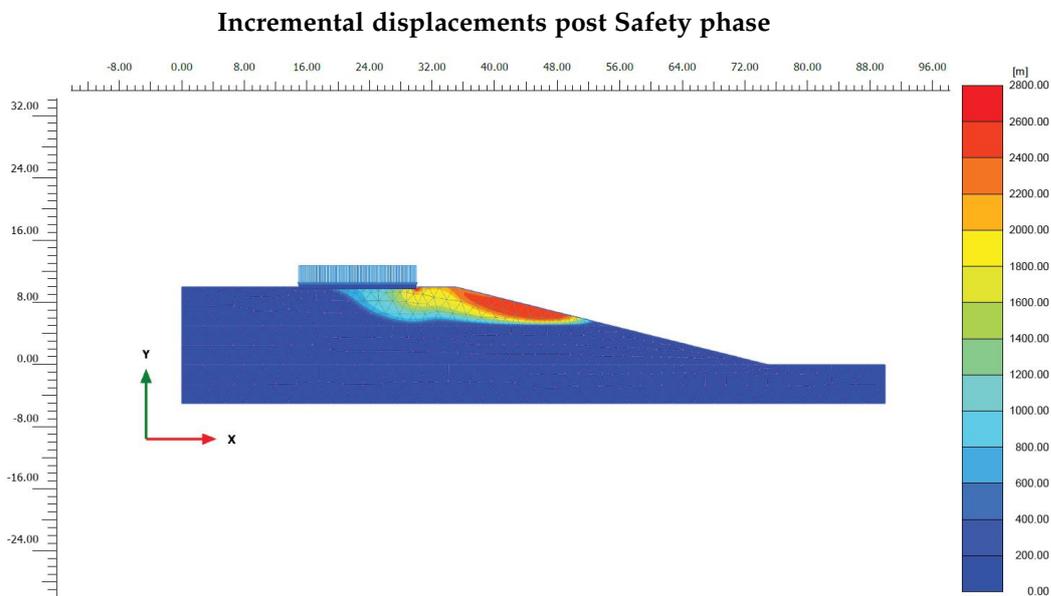


Figure K.1: Demonstrative case study - Illustration of incremental displacements $|\Delta u|$ from the Safety analysis in PLAXIS

L | DEMONSTRATIVE CASE STUDY: RANDOM FIELDS AS A FUNCTION OF θ

In this appendix, illustrations of the distribution of values for the random field of "G" in the demonstrative case study are presented. This serves two purposes:

- Visually reinforcing the argument for the appropriateness of the mesh
- Providing a visual support for understanding the results presented in 6.2.

Distribution of shear modulus values over the domain for $\theta_v = \theta_h = 2 m$

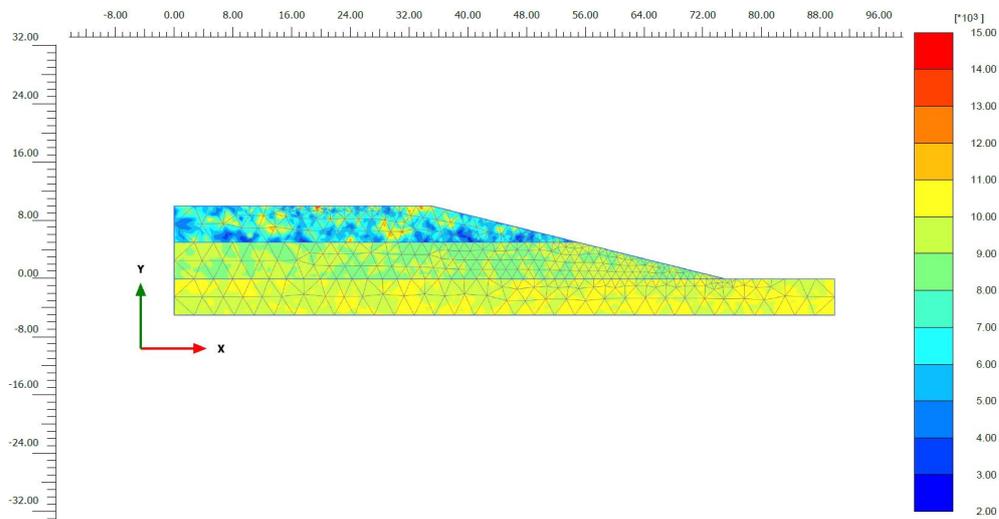


Figure L.1: Demonstrative case study - random field for $G [kPa]$, $\theta_v = \theta_h = 2 m$

Distribution of shear modulus values over the domain for $\theta_v = 2 m$, $\theta_h = 10 m$

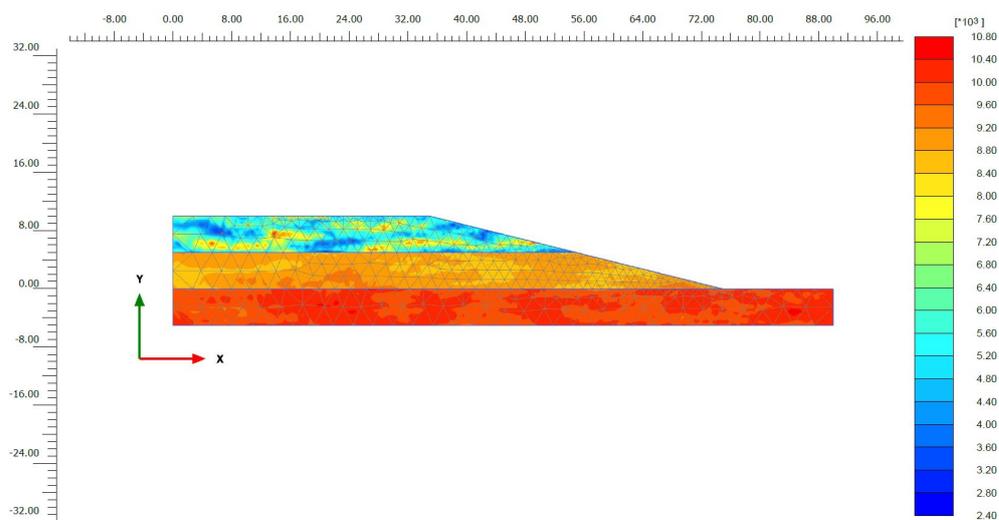
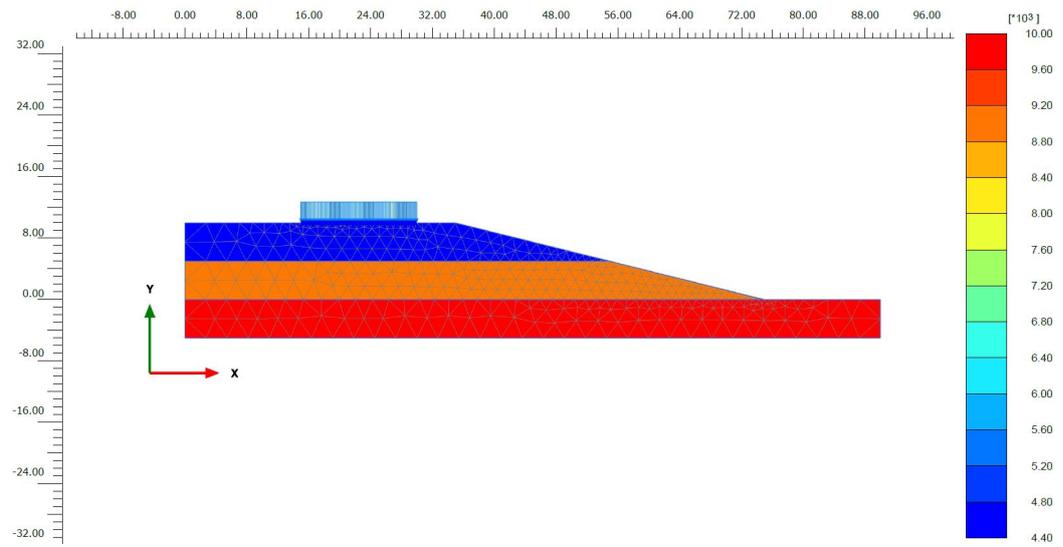


Figure L.2: Demonstrative case study - random field for $G [kPa]$, $\theta_v = 2 m$, $\theta_h = 10 m$

Distribution of shear modulus values over the domain for $\theta_v = \theta_h \rightarrow \infty$ **Figure L.3:** Demonstrative case study - random field for G [kPa], $\theta_v = \theta_h \rightarrow \infty$

In this appendix, information on average computation times for the analyses carried out in this thesis is presented. This includes the number of random parameters considered, the number of construction stages, the number of integration points and of course the average duration per realization. These are highlighted in table M.1 below. The same information is illustrated in figure M.1.

Table M.1: Average computation times for all the cases studied in the thesis

Case	Parameters	Stages	Stress points	Time/realization [s]
Verification 1	2	3	1992	23
Verification 2	2	3	2500	26
Starnmeer validation	2	3	11,000	94
Demonstrative case study	3	3	12,500	190

This provides a more comprehensive perspective regarding the influence of relevant factors on operational performance discussed in sub-chapter 7.2.

Between the second verification and the validation case study it can be seen how for the same number of random parameters and construction stages, the number of stress points significantly affects computation times.

Then, comparing the Starnmeer validation case study with the demonstrative case study, it can be seen how influential the addition of an extra parameter can be in slowing calculation time. Of course, it should also be taken into account that the final case study required a slightly larger number of stress points and that the output panel is opened at every realization, which is not the case for the validation.

Computation times for all cases studied in this thesis

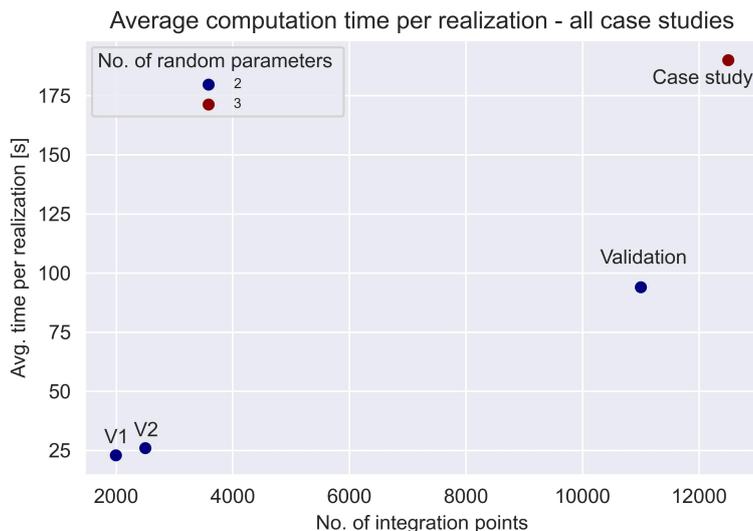


Figure M.1: Illustration of average calculation times in case studies treated in this thesis

