Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Ole Eiesland

Implementation of random finite element method in Plaxis Using API / Python scripting

Simulating spatial variable soil "biaxial-test", oedometertest, slope stability and bearing capacity problems

Master's thesis in Geotechnics and Geohazards
Supervisor: Yutao Pan
June 2022

NTNU
Norwegian University of
Science and Technology

Ole Eiesland

Implementation of random finite element method in Plaxis Using API / Python scripting

Simulating spatial variable soil "biaxial-test", oedometertest, slope stability and bearing capacity problems

Master’s thesis in Geotechnics and Geohazards
Supervisor: Yutao Pan
June 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Ole Eiesland

Implementation of random finite element method in Plaxis Using API / Python scripting

Simulating spatial variable soil "biaxial-test", oedometertest, slope stability and bearing capacity problems

Trondheim, June 2022

MASTER THESIS: TBA4900

Main supervisor: Professor Yutao Pan

Department of Civil and Environmental Engineering

Norwegian University of Science and Technology (NTNU)

NTNU – Trondheim
Norwegian University of
Science and Technology

Preface

This masters thesis is presented as final deliverable in the subject "TBA4900 - Geotechnical Engineering, Master Thesis" at the Department of Civil and Environmental Engineering, NTNU during the spring semester of 2022. The course is 30 SP credits.

I believe programming skill is important for engineers to be able to utilize the tools available and invent the tools needed to solve the challenges of society. I encourage NTNU to further education in programming skills. I would like to commend Bently systems for opening up and realizing the benefits of a Plaxis API.

I would like to thank professor Yutao Pan, NTNU for supervising the project.

Special thanks to my girlfriend, we correlate.

Trondheim, 2022-07-11

Ole Eiesland

Summary and Conclusions

Soil is a complex medium. Its inhomogeneous nature means that the physical parameters of soil vary spatially both vertically and laterally. Traditionally soil properties is modeled with a representative value, usually some kind of mean value or similar. In probabilistic methods, this variability, or uncertainty is taken into account by treating the soil as a random variable sampled from a probability distribution. By using random field theory and statistics one can try to describe how the soil parameters are distributed in space and how they vary with distance. In a slope stability or a bearing capacity problem, the spatial distribution of the soil strength governing parameters has a direct impact on the development of the failure surface, the failure mechanism and therefore the over all stability. To simulate stability a finite element program can be used with the soil model parameters input to the finite element mesh based on statistical spatial correlated random fields. To simulate variability and uncertainty, the modeling is repeated many times with different random fields. This is the random finite element method.

Current modern commercial soil modeling software do not support random finite element method, and published research random finite element software code do not have the advanced functionality and soil behaviour models as modern commercial geotechnical software. However, since the random finite element method does not change the way the problem is simulated, only the input parameters change, modern software packages can be used by automatically generating input data and running the calculation steps in loops. Plaxis, a modern software package developed by Bently, has capabilities like this through its application program interface and python scripting.

This thesis presents a method to run the random finite element method in Plaxis geotechnical software package using python API scripting interface, and demonstrate the versatility and verifies the validity of the implementation of the random finite element method on a slope stability and a bearing capacity problem. The thesis reproduce and expand on result from literature. It is further used to investigate the nature and geometry of failure mechanisms in soil and tries to demonstrate the power of fully coupled flow-deformation analysis.

Contents

Chapter 1

Introduction

1.1 Background

The inhomogeneous nature of soil means that the physical parameters vary spatially both vertically and laterally. Common cause of deviation of expected performance of geotechnical design is variability in soil properties at the site. Traditionally soil properties is modeled with a representative value, called the characteristic value, usually some kind of mean value or similar. To account for unfavorable deviations from the characteristic value the design codes introduce a partialfactor which is multiplied with the characteristic value [Bond et al., 2013]. In probabilistic methods, this spatial variability, or uncertainty is taken into account by treating the soil as a random variable sampled from a probability distribution. Traditional probabilistic methods like First order second moment (FOSM) and First Order reliability Method (FORM) are analytical solutions of limit equilibrium states, the analytical solution is limited in the complexity of the problems it can model and the assumptions of limit state allows no study of i.e magnitude of deformations. By using random field theory and statistics one can try to describe how the soil parameters are distributed in space and how they vary with distance. In a slope stability or a bearing capacity problem, the spatial distribution of the soil strength governing parameters has a direct impact on the development of the failure surface, the failure mechanism and therefore the overall stability. To simulate stability a finite element program can be used with the soil model parameters input to the finite element mesh based on statistically and spatially correlated random fields. The Finite Element Method (FEM) allows no prior assumption of how the failure surface looks, like FORM/FOSM, who must assume a fixed failure surface shape e.g a circular or ellipsoid slip surface. To simulate variability and uncertainty, the modeling is repeated many times with different random fields in a Monte Carlo simulation. This is the random finite element method.

Problem Formulation

Current modern soil modeling software does not support random finite element method, and published research random finite element software code do not have the advanced functionality like complex soil models and soil structure interactions as modern commercial geotechnical software. However, since the random finite element method does not change the way the problem is simulated, only the input parameters vary spatially, modern software packages can be used if a way to specify the input and the simulation run parameters can be controlled in an automatic and efficient manner. Plaxis, a modern software package developed by Bently, or Optum G2, has capabilities like this through its application program interface and python scripting. It is of great interest to research on spatially varying soil modeling to utilize the advanced functions and ease of access of existing software. Gaining this ability will expand the complexity of the problems allowed to be simulated by the Random Finite Element Method (RFEM). The current userbase of geotechnical engineers using Plaxis is large and providing a RFEM tool in plaxis could allow for them to incorporate probabilistic methods in their designs.

Literature Survey

The random finite element method (RFEM) has been in use since the mid-1990s [e.g, see Griffiths and Fenton, 1993]. RFEM combines random field theory to represent the spatially varying soil with finite element method (FEM) for deformation analysis. Stochastic analysis in FEM methods can be built into the finite element equations themselves [e.g., see Vanmarcke and Grigoriu, 1983], or a Monte Carlo approach can be used were multiple realizations of different spatial soil models can be analyzed together. The Monte Carlo approach can be computationally demanding, but can be very flexible by utilizing arbitrary FEM code changing just the input. The Monte Carlo RFEM and its application to many geotechnical problems including seepage, bearing capacity, earth pressure and settlement is described in more detail in book by Fenton, Griffiths, et al. [2008]. Examples of RFEM analysis for slope stability can be found in the literature [e.g, see Fenton, Griffiths, et al., 2008, Chapter 13] based on FEM code developed by Smith, Griffiths, and Margetts [2013]. The RFEM code is publicly available and extensive research has been conducted using it. A list of all but the most recent publications using the code is available here: (http://random.engmath.dal.ca/rfem/rfem_pubs.html). The RFEM code is for two-dimensional plane strain analysis of elastic perfectly plastic soils with a Mohr Coulomb failure criterion. For a detailed discussion of the method [e.g, see Griffiths and Lane, 1999]. The limitation of the failure criteria and the soil model can restrict the application of the method to more complex soils who displays different characteristics. Software with a range of failure criteria and soil models exist, such as Plaxis by Bently [Brinkgreve et al., 2010] or Optum G2 [Krabbenhøft et al., 2016]. Plaxis do not have built-in random field functionality.

What Remains to be Done?

To be able to describe more complex geotechnical problems, a method to unify material models and numerical methods is needed. Implementation of new soil models into FEM code is not straightforward. Plaxis allows for python scripting through an application programming interface (API). The RFEM Monte Carlo method could be implemented in Plaxis by scripting random field input and automating simulation.

1.2 Objectives

The main objectives of this thesis are

1. Implement The RFEM Monte Carlo method in Plaxis using python API interface
2. Demonstrate and verify the implementation on a simple slope stability problem
3. Demonstrate versatility by extending the implementation to a simple bearing capacity problem
4. Compare results to analytical results where available
5. Reproduce literature results produced by different software
6. Utilize the power of Plaxis by unifying advanced soil model and coupled flow deformation analysis with random soil field input in a Monte Carlo simulation.

1.3 Limitations

The field of random behavior of soils in geotechnical problems is extensive and this project work scope only scratches the surface. The methodology of finite element numerical computations and random fields are only described briefly and beyond the scope of this project course to go into in detail.

The python code presented is not attempted optimized in any way, the focus is on proof of concept. The result presented is a tool, further work, research and application will prove its usability.

1.4 Approach

A literature search was conducted to get an overview of the current implementations of the RFEM code. Source code and accompanying documentation was also studied where available. The main part of the project was to code the implementation of the RFEM method into the Plaxis 2D software using the python scripting API. Study of the Plaxis 2D software manual and online documentation was key to get familiar with Plaxis and the API functions to control and automate the program execution from the python code. Example problems from literature is used to verify the validity of the implementation. Results from other software is reproduced to compare results. Analytical results is compared to the implementation. To demonstrate versatility a slope stability and bearing capacity problem is used.

1.5 Structure of the Report

The rest of the report is structured as follows. Chapter 2 gives a brief introduction to random field theory and finite element analysis. In chapter 3 the implementation of the RFEM method in Plaxis using python API script is discussed. Chapter 4 compares results of simulations using the python API implementation of the RFEM method in Plaxis. Chapter 5 gives a summary and discussion of the results and main findings of the thesis.

Chapter 2

Theory

This chapter gives a brief introduction to the soil models, random field theory and finite element analysis used in this thesis.

2.1 Soil strength and failure criteria

A failure criteria is a mathematical model that describes the strength of a material. The failure criterion is governed by the stress components acting on the material and the material properties themselves.

2.1.1 Tresca failure criterion

In a clay, and to some extent for other low permeable soils, during and immediately after rapid load application, in undrained condition, when pore water is trapped and unable to drain due to low permeability, the failure criterion can be described as normal stress independent and total stress based, as:

$$s_{\mathrm{u}} = c_{\mathrm{u}} = \frac{1}{2}(\sigma_1 - \sigma_3) \tag{2.1}$$

where the subscript u stands for undrained condition. The Tresca failior criterion is independent of confining pressure. A illustration of the Tresca failure criterion is shown in figure 2.1 [Tresca, 1864]

Table 2.1 shows typical values for undrained shear strength for Norwegian clays

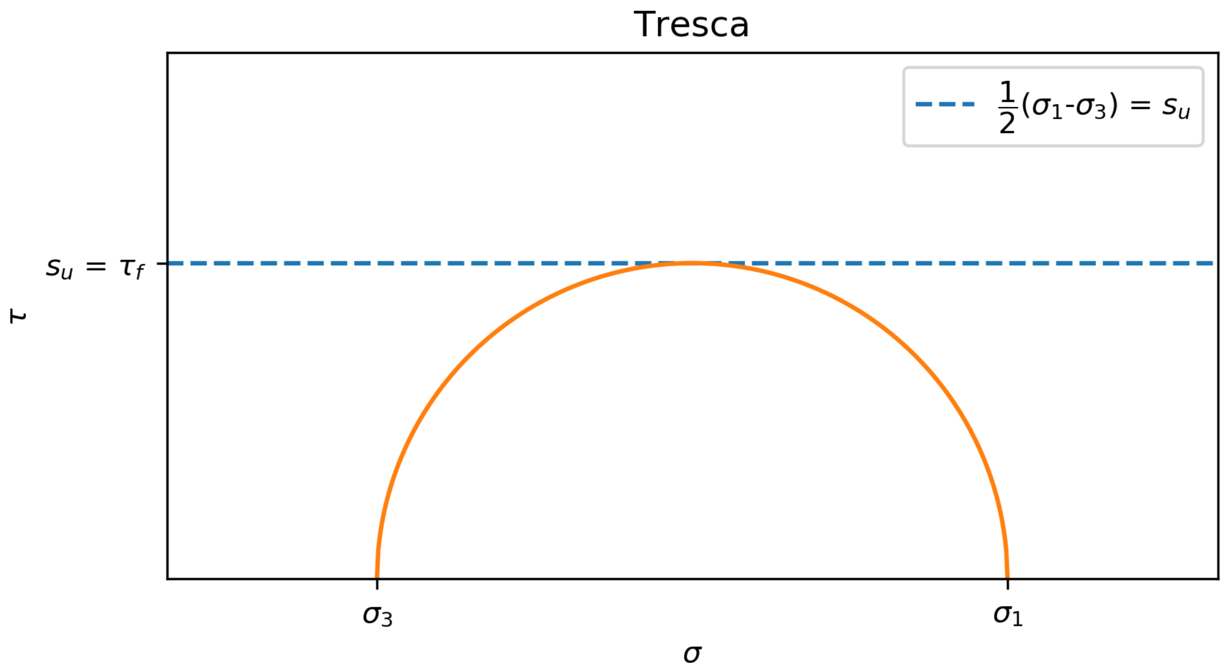


Figure 2.1: The Tresca failure criterion

Table 2.1: Typical values for undrained shear strength for Norwegian clays

Soil	Clay		
Strength/stiffness	Soft	Medium	Stiff
s_u	<25 kPa	25 to 50 kPa	>50 kPa

2.1.2 Mohr Coulomb failure criterion

In drained conditions, the strength of soils is controlled by the effective normal stresses defined as the total stress minus the pore pressure as:

$$\sigma' = \sigma - u \tag{2.2}$$

Coulombs law [Coulomb, 1773] defines the strengt limit and gives the Mohr Coulomb failure criteria as:

$$\tau_f = c + \sigma' tan(\phi) = (\sigma' + a)\, tan(\phi) \tag{2.3}$$

where $tan(\phi)$ is the friction coefficient, a the attraction, and $c = a \cdot tan(\phi)$ is the cohesion. see figure 2.2 for a illustration of the parameters and the failure criterion. Experimental results on soils show a modest strength non-linearity, when Mohr Coulomb criterian is applied, the parameters is used as curve fitting parameters to fit a straight line to the experimental results in the relevant stress range.

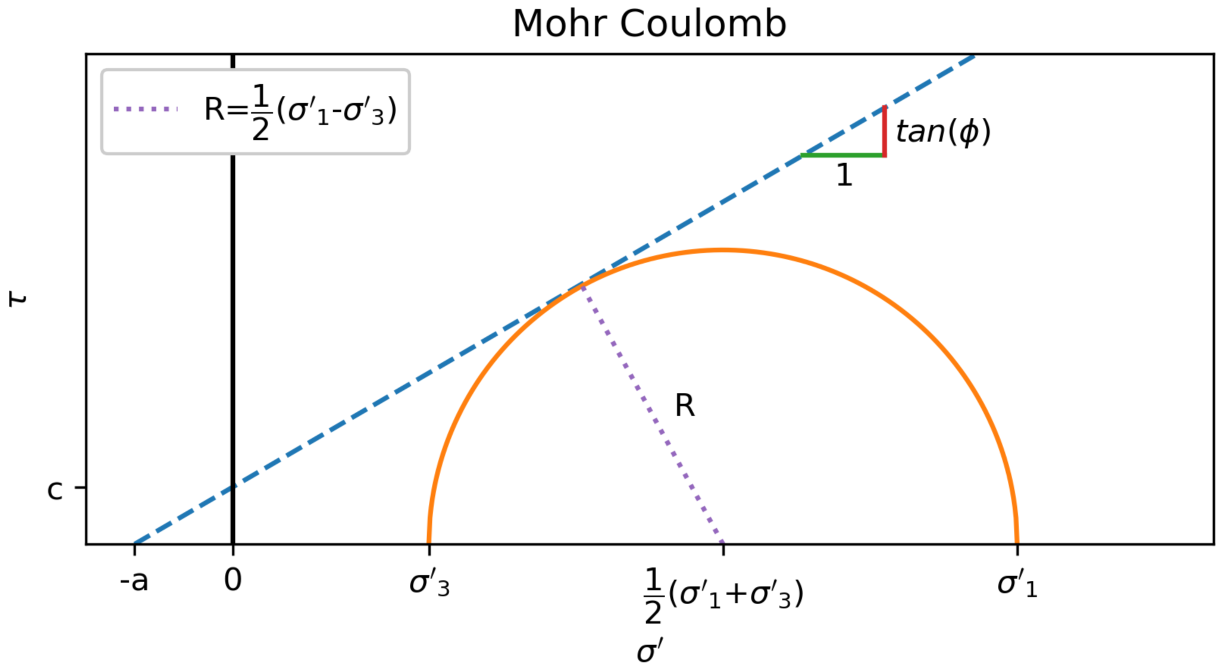


Figure 2.2: The Mohr Coulomb failure criterion

From figure 2.2, considering the right triangle given by the intersection of the Mohr Coulomb line, the center of the Mohr circle and the tangent point of the Mohr circle and the Mohr Coulomb line, the Mohr Coulomb criterion ,using the law of sines, can be reformulated based on principal effective stresses as:

$$\frac{1}{2}(\sigma_1' - \sigma_3') = (a + \frac{1}{2}(\sigma_1' + \sigma_3')sin(\phi) \tag{2.4}$$

Rearanging:

$$(\sigma_1' + a) = N(\sigma_3' + a) \tag{2.5}$$

$$N = \frac{(1 + sin(\phi))}{(1 - sin(\phi))} \tag{2.6}$$

Mohr Coulomb is independent of the intermediate stress, σ_2. For plain strain, 2D, problems described in this project, the out of plane stress component is normally the intermediate stress,

σ_2. A convenient formulation of the Mohr Coulomb criterion for plane strain is:

$$s = \frac{\sigma_1 + \sigma_3}{2} \tag{2.7}$$

$$t = \frac{\sigma_1 - \sigma_3}{2} \tag{2.8}$$

The inclination of the Mohr Coulomb line is then $sin(\phi)$:

$$\frac{t}{s+a} = \frac{(\sigma_1 + a) - (\sigma_3 + a)}{(\sigma_1 + a) + (\sigma_3 + a)} = \frac{N-1}{N+1} = sin(\phi) \tag{2.9}$$

The plot s vs t is often called NGI or MIT plot and is used in this project when estimating the friction angle from 2 dimensional plain strain "biaxial"-tests at different confinement pressures. See figure 2.3 for illustration.

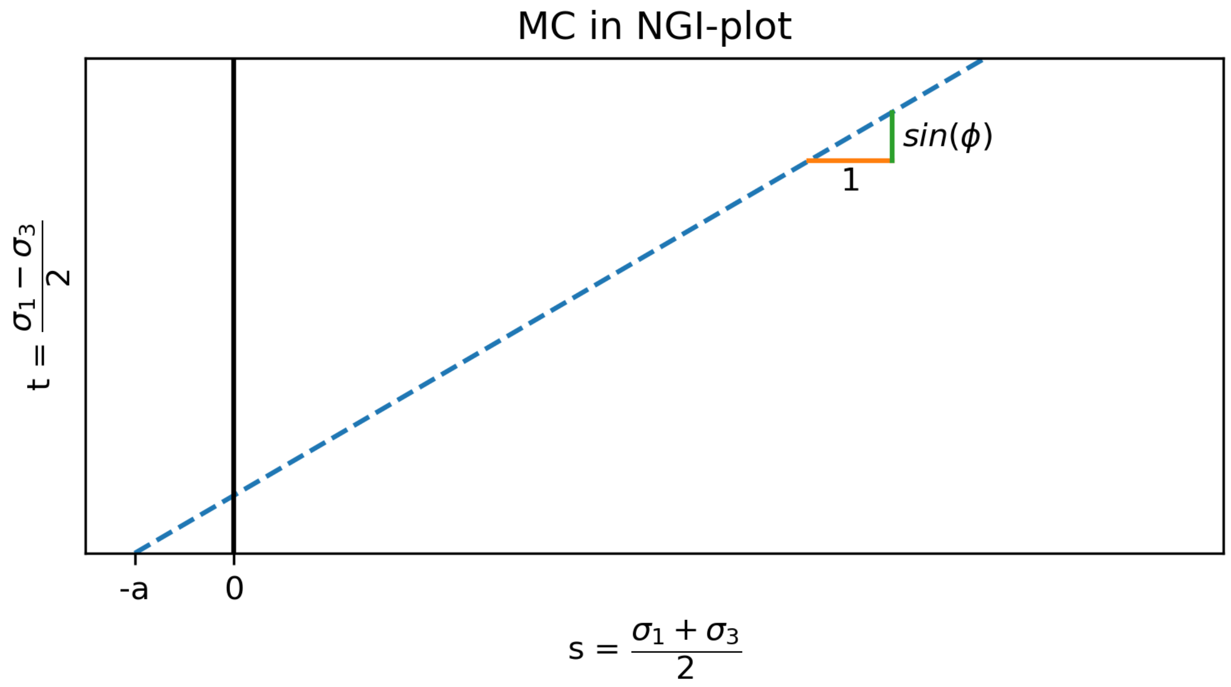

MC in NGI-plot
$sin(\phi)$
1
$t = \frac{\sigma_1 - \sigma_3}{2}$
-a
0
$s = \frac{\sigma_1 + \sigma_3}{2}$

Figure 2.3: The Mohr Coulomb failure criterion for plain strain displayed in a "NGI-plot"

Table 2.2 shows typical values for attraction and friction angle for Norwegian clays

Table 2.2: Typical values for attraction and friction angle for Norwegian clays

Soil	Clay		
Strength/stiffness	Soft	Medium	Stiff
a	5-10 kPa	15-25 kPa	30-60 kPa
ϕ	20 °	25 °	30 °

2.2 Soil models

2.2.1 Colombian - Linear elastic, perfectly plastic

In elasto-plasticity, total strain can be separated into elastic strain and plastic strain, equation 2.10.

$$\epsilon = \epsilon^{elastic} + \epsilon^{plastic} \tag{2.10}$$

Elastic strains are reversible, plastic strains are permanent. In a linearly elastic, perfectly plastic soil model, all strains are elastic when the stress state is below the failure criteria.

The linear elastic perfectly plastic coulomb model is good when focus is on shear strength problems on materials with small volumetric deformation, like undrained clay with incompressible pore water or densly packed sand. The colombian model will be used in this project to study shear strength.

2.2.2 Modified Cam Clay Model - Volumetric hardening

The coulomb model is not suitable when large permanent volumetric strains are to be considered, like for soft clay in drained conditions. Figure 2.5 shows a sketch of a clay sample in a oedometer compression test. It can be seen that the clay sample shows a non-linear relationship between stress and strain. The clay sample also exhibit a change in stiffness around the preconsolidation pressure, $p\prime_c$.

A better soil model to describe soil with large voumetric changes, like clays that behaves like in figure 2.5, is the Modified Cam Clay Model (MCCM) Wood [1990]. In a oedometer condition, as used in this project, the MCCM is given with the following equations:

$$v = 1 + e = \frac{volumeof pores + volumeof solids}{volumeof solids} = \frac{V_p + V_s}{V_s} \tag{2.11}$$

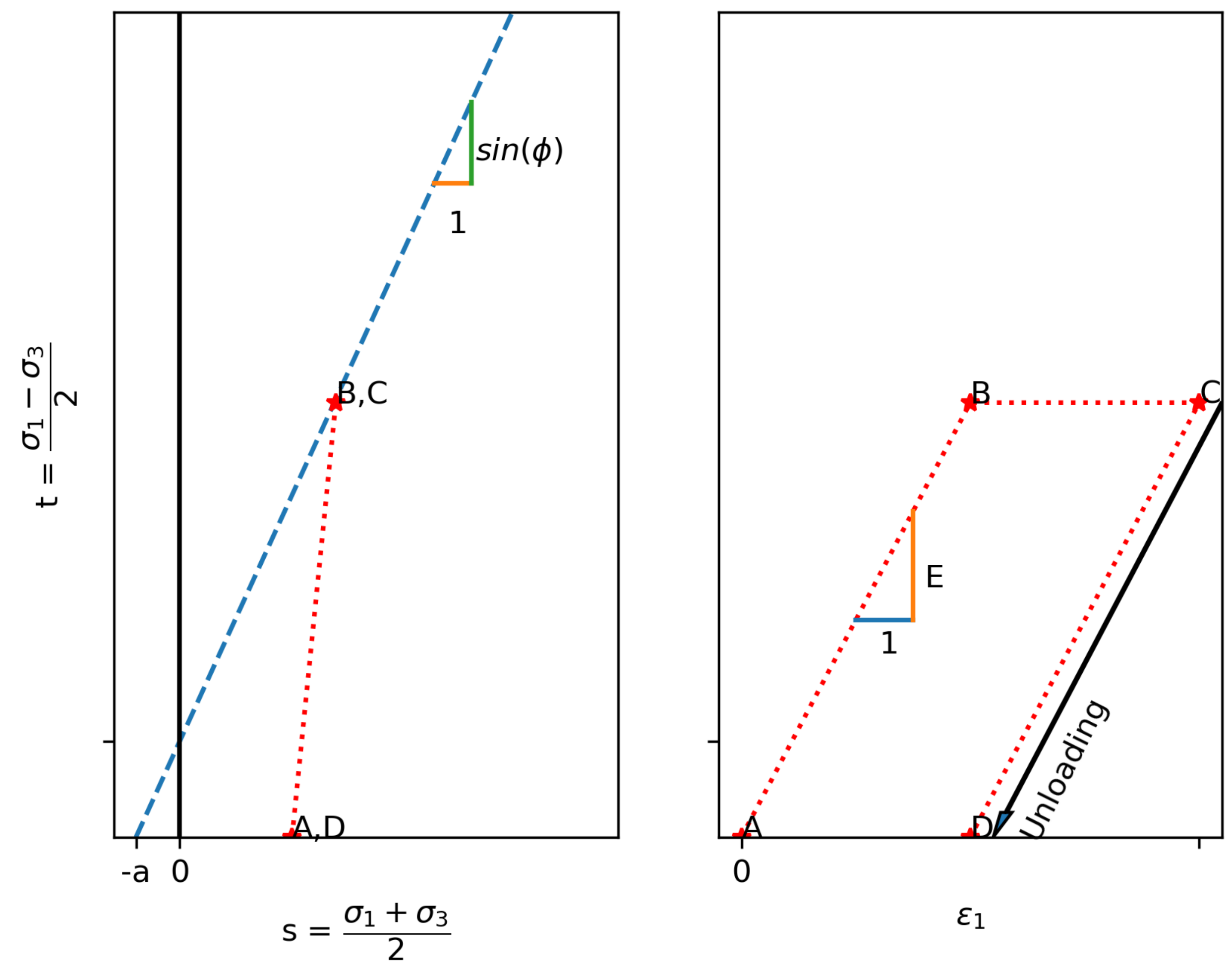


Figure 2.4: Stress path and deformation curve for a linear elastic perfectly plastic soil model

$$d\epsilon_p = -\frac{dV}{V} = \frac{dV_p}{V_p + V_s} = \frac{d(V_p + V_s)/V_s}{(V_p + V_s)/V_s} = -\frac{dv}{v} \tag{2.12}$$

$$\sigma\prime < p\prime_c : OCrange : \Delta v = -\kappa \Delta ln\sigma\prime = -\kappa ln\frac{\sigma\prime}{\sigma\prime_0} \tag{2.13}$$

$$\sigma\prime > p\prime_c : NCrange : \Delta v = -\lambda \Delta ln\sigma\prime = -\lambda ln\frac{\sigma\prime}{\sigma\prime_0} \tag{2.14}$$

where v is the spesific volume, e is the void ratio, Δv is the change in spesific volume when loading from $\sigma\prime_0$ to $\sigma\prime$.

The Janbu modulus:

$$\Delta\epsilon = \Delta\epsilon_p = \frac{1}{m_{NC}} ln\frac{\sigma\prime}{\sigma\prime_0} \tag{2.15}$$

comparing equation 2.15 to 2.14 and using equation 2.12 we get:

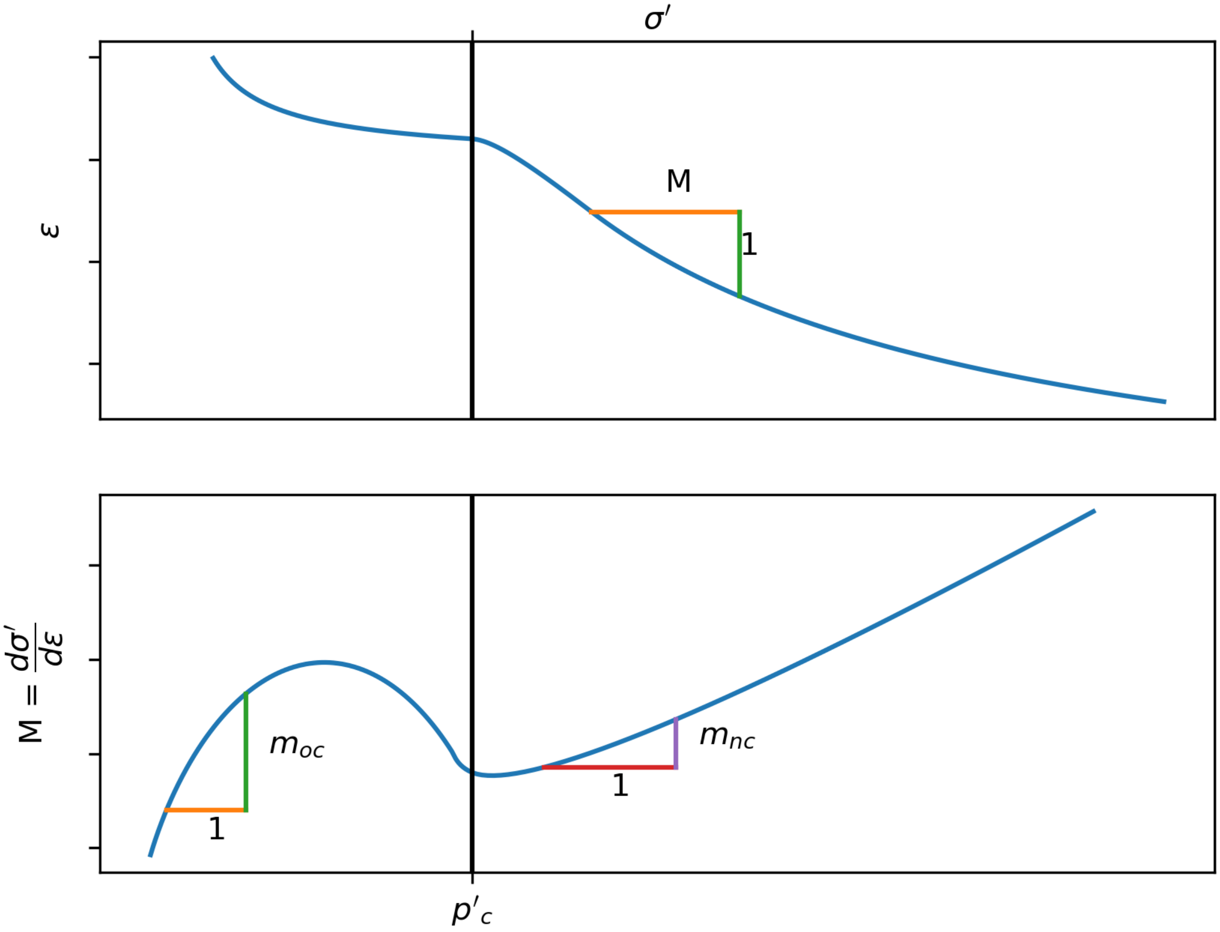


Figure 2.5: Sketch of Stress vs strain plot (top) and oedometermodulus (bottom) from a oedometer compression test on a overconsolidated clay

$$\lambda = \frac{\nu}{m_{NC}} = \frac{1+e}{m_{NC}} \tag{2.16}$$

Similarly if we interpret a linear modulus in the OC range:

$$\kappa = \frac{\nu}{m_{OC}} = \frac{1+e}{m_{OC}} \tag{2.17}$$

Tabel 2.3 shows example MCCM parameters for some Norwegian clay.

Table 2.3: Typical values for MCCM-parameters for Norwegian clays

Soil	Clay		
Strength/stiffness	Soft	Medium	Stiff
m_{NC}	10	20	30
m_{OC} about $5m_{NC}$	50	100	150
$w(\%)$	70	35	25
$e = 2.7w$	1.9	0.95	0.70
λ	0.30	0.10	0.06
κ	0.06	0.02	0.01

2.3 Random Fields

To model the spatial variability of soils random fields are used. To describe the random field of a soil parameter, e. g. the undrained shear strength of the soil, three parameters are commonly used. The Mean, μ, Standard deviation, σ of the soil parameters underlying probability distribution and the Spatial correlation length, θ also known as the scale of fluctuation. The underlying probability distribution of the random variable can be any distribution that fit the observed data. Examples are the Gaussian distribution or the log-normal distribution. Both of these parametric distributions can be described only with two parameters, mean and standard deviation. More complex distributions can be used but require more parameters. The mean is a measure of around which value the soil strength parameter is distributed. The standard deviation tells how much the values in the soil random fields varies in values. The spatial correlation length, or scale of fluctuation, is a measure of how similar in strength points in the spatial random field are depending on how far away the points are from each other. Large spatial correlation length vary smoothly and slowly over a given distance, while a small scale of fluctuation is jagged and rapidly varying. Soil samples taken close together is more likely to be similar than soil samples taken far apart. Due to the process of soil deposition the soil tend to have different properties in differing direction, soil is anisotropic. Typically for layered soils, the soil properties are more similar in the horizontal direction than in the vertical direction. This anisotropy can be represented by having a longer scale of fluctuation in the horizontal direction than in the vertical direction.

Remark: There is a difference between homogeneous media and isotropic media, a homogeneous media can be anisotropic, but the anisotropy parameter does not change depending on where it is measured.

2.3.1 SRM - Spectral Representation Method

Various methods exist for generating random fields that can be used to represent spatially variable soil. [see e.g. Fenton, Griffiths, et al., 2008, Chapter 6]. The methods differ in their efficiency and accuracy and complexity e.g. ability to describe anisotropy. In this project work the spectral representation method is used. It is showed by Shinozuka and Deodatis [1996] how to simulate multi-dimensional homogeneous stochastic fields using the spectral representation method. Sample functions of the stochastic field can be generated using a cosine series formula. These sample functions accurately reflect the prescribed probabilistic characteristics of the stochastic field when the number of terms in the cosine series,N_1 and N_2 in equation 2.18, is large.

$$f_0(x_1,x_2)=\sqrt{2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}A_{n_1n_2}[cos(\kappa_{1_{n_1}}x_1+\kappa_{2_{n_2}}x_2+\Phi^{(1)}_{n_1n_2})+cos(\kappa_{1_{n_1}}x_1-\kappa_{2_{n_2}}x_2+\Phi^{(2)}_{n_1n_2})] \quad (2.18)$$

where:

$$A_{n_1n_2}=\sqrt{2S_{f_0f_0}(\kappa_{1_{n_1}},\kappa_{2_{n_2}})\Delta\kappa_1\Delta\kappa_2} \quad (2.19)$$

$$\kappa_{1_{n_1}}=n_1\Delta\kappa_1 \quad \text{and} \quad \kappa_{2_{n_2}}=n_2\Delta\kappa_2 \quad (2.20)$$

$$\Delta\kappa_1=\frac{\kappa_{1u}}{N_1} \quad \text{and} \quad \Delta\kappa_2=\frac{\kappa_{2u}}{N_2} \quad (2.21)$$

and:

$$A_{0n_2}=A_{n_10}=0 \quad \text{for} \quad n_1=n_2=0,1,\ldots,N_1-1 \quad (2.22)$$

$\Phi^{(1)}_{n_1n_2}$ and $\Phi^{(2)}_{n_1n_2}$ are random phase angles distributed uniformly over the interval. $[0,2\pi]$. $S_{f_0f_0}(\kappa_{1_{n_1}},\kappa_{2_{n_2}})$ is the power spectral density function and is a real non-negative function of the wave numbers κ_1 and κ_2. κ_{1u} and κ_{2u} are the upper cut-off wave numbers.

To avoid aliasing, the following condition must be true:

$$\Delta x_1\leq\frac{2\pi}{2\kappa_{1u}} \quad \text{and} \quad \Delta x_2\leq\frac{2\pi}{2\kappa_{2u}} \quad (2.23)$$

See chapter 3 for details on implementation.

Using the Wiener-Khinchine relation, for a single sided one dimensional spectral density function $G(\kappa) = 2S(\kappa)$ the following relations exist between G, variance σ^2 and scale of fluctuation θ:

$$G(\kappa) = \frac{2\sigma^2}{\pi}\int_0^\infty \rho(\tau)cos(\kappa\tau)d\tau \tag{2.24}$$

$$\sigma^2 = \int_0^\infty G(\kappa)d\kappa \tag{2.25}$$

$$\theta = \frac{\pi G(0)}{\sigma^2} \tag{2.26}$$

Equation 2.18 generates a field with mean $\mu = 0$. To get the field to the desired mean value the mean value is added to the values of the field.

2.4 Finite Element Method

The Finite Element method is a numerical method that can be used to solve a multitude of geotechnical and other engineering problems. Being a numerical method, FEM gives approximate solution, care should be used when constructing the model to be calculated. Things to keep in mind when running FEM analysis is the volume extent and the boundary interface of the model, element type, density of mesh and criteria of convergence. In comparison to limit equilibrium methods (LEM), FEM methods can give deformations in a state before ultimate limit state i.e. before failure. Contrary to analytical methods like FOSM/FORM wich assumes the geometry of the slip surface, FEM have no prior assumptions of the failure mechanism i.e. no assumption of a circular failure surface in a LEM slope stability analysis. It is a very attractive property of FEM analysis combined with spatial variable soil, to allow slope failure to develop naturally by finding the path of weakest soil.

Seven steps of a finite element program is described by Nordal [2020]:

1. Element modeling, equations are built for each element, integration in points inside the element is used to form stiffness matrix

2. Global modelling, all element stiffness matrices are assembled into a system of equations to form a global stiffness matrix. An incremental load vector is found

3. Equation solving of the global equation system for the load increment. This gives a displacement increment

4. Stress evaluation of the calculated displacement increment. The displacements are used to find the strains which is used to find the stress increments

5. Testing for numerical accuracy. If the calculations show too high unbalanced forces, it will be necessary to adjust the load increment and/or add more iterations. If so step 1-5 must be recalculated. When the results converges the program proceeds to step 6

6. Updating of results by adding the deformations and stress to form total deformations and total stresses.

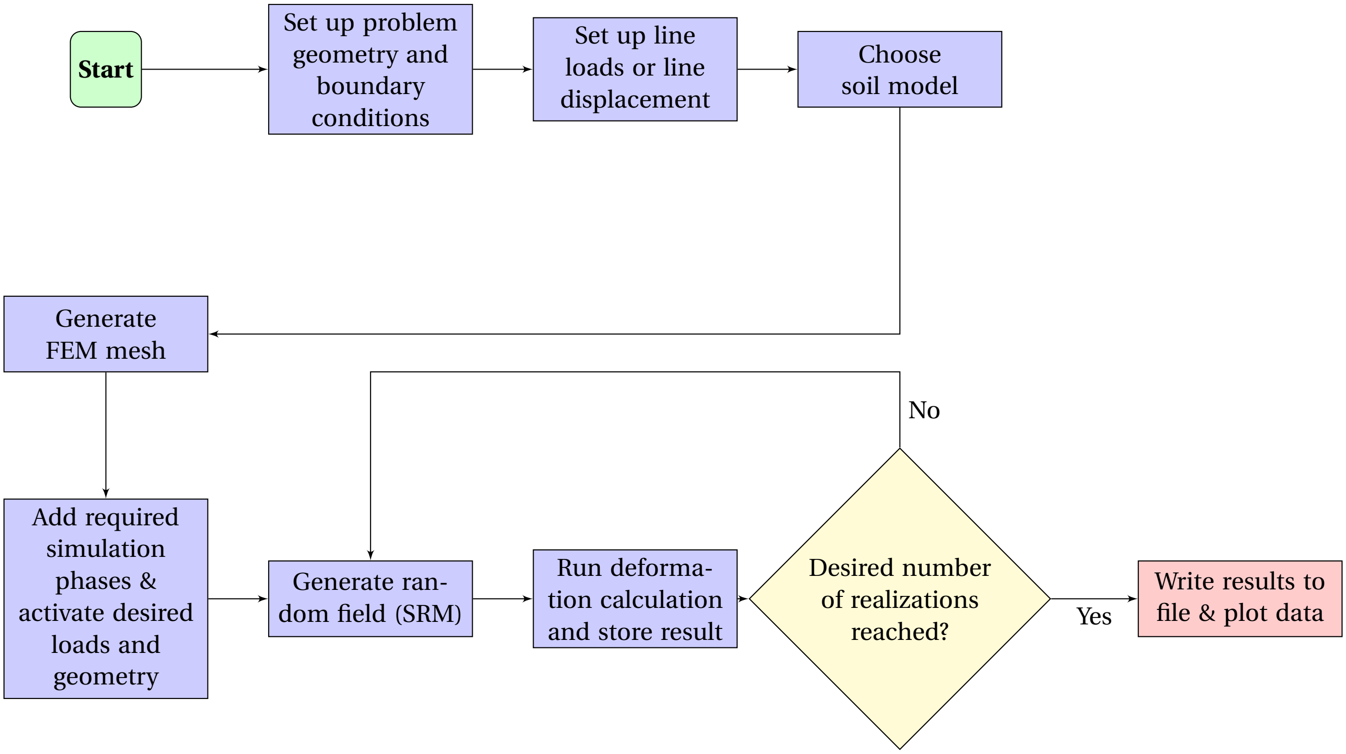
7. Calculation of new load increment. The response of the new load increment is found by repeating 1 to 6. The load is gradually ramped up in increments until the specified external load is reached, or failure occur

Remark: Plaxis measures the fraction of the applied loadstep according to step 7 above and will stop if a numerical convergence cannot be reached. The numerical inconvergence is usually a sign of failure because the applied load cannot be balanced by the model. This property is used as a failure criterion, a variable stores the fraction of the applied load and is accessed automatically to tell if we have failure or not. As mentioned above, care in the FEM parametrization, e.g choice of meshing, should be appropriate to avoid numerical instability resulting in other failure modes than the ones being studied.

Chapter 3

RFEM implementation in Plaxis 2D using python API

This chapter discusses the implementation of the RFEM method in Plaxis using python API scripting. An overview can be seen in figure 3.1, more details are given in the following subsections.



Start
Set up problem geometry and boundary conditions
Set up line loads or line displacement
Choose soil model
Generate FEM mesh
Add required simulation phases & activate desired loads and geometry
Generate random field (SRM)
Run deformation calculation and store result
Desired number of realizations reached?
No
Yes
Write results to file & plot data

Figure 3.1: flowchart

3.1 SRM implementation

In generating the random soil strength field an implementation of the SRM method written in MATLAB code by Yutao Pan out of Deodatis, Shinozuka, and Papageorgiou [1990] is converted to python code. The implementation in this project uses a Gaussian spectral density function given below in equation 3.5. Other spectral density functions exists and can be implemented.

$$S_{f_0 f_0}(\kappa_{1_{n_1}},\kappa_{2_{n_2}}) = \frac{1}{4}\frac{\theta_x}{\pi}\frac{\theta_y}{\pi} e^{(-(\frac{\theta_x^2\kappa_1^2}{4\pi})} e^{(-(\frac{\theta_y^2\kappa_2^2}{4\pi})} \tag{3.1}$$

Figure 3.2 shows example output of the SRM implementation for different input mean μ, coefficient of variation CoV and scale of fluctuation θ, together with estimates of same parameters from the generated field. The estimates shows reasonable good comparisons to the input values.

```
import numpy as np
def srm():
    inmiu=20
    incov=0.1
    sig=inmiu*incov;
    SOFx=8
    SOFy=3
    b1=SOFx/np.sqrt(np.pi)
    b2=SOFy/np.sqrt(np.pi)
    m1=64 #integration mesh
    m2=64
    m11=m1-1
    m22=m2-1
    bk1u=5
    bk2u=5
    k1u=bk1u/b1
    k2u=bk2u/b2

    detak1=k1u/m1
    detak2=k2u/m2
    g=0.0
    nrep=1

    nnx=30     #50#x direction number of mesh X
    nny=10     #50#y dir number of mesh

    nnx1=nnx +1
    nny1=nny +1
    tt=np.zeros((nnx1,nny1))
    RR=np.zeros((nnx1,nny1))
    gg=np.zeros((nrep,nnx1,nny1)) #realization at different points
    xx=np.zeros(nnx1) # x dir coor
    yy=np.zeros(nny1) # y dir coor

    lx=30 # length x dir
    ly=10 # length y dir
    spx=lx/nnx
    detax=np.pi/k1u
    spy=ly/nny
    detay=np.pi/k2u
    sq2=np.sqrt(2)
    intsd=0
    intsdf=np.zeros((nnx,nny)) #integrate sdf at different points
    lx
    limitx=period1/4
    ly
    limity=period2/4
    rng = np.random.default_rng(42)

    for nround in range(0,nrep):
        rand1=np.random.rand(m1,m2)
        rand2=np.random.rand(m1,m2)
        fai1=rand1*2*np.pi
        fai2=rand2*2*np.pi
        for nx in range(0,nnx+1):
            for ny in range(0,nny+1):

                nx1=nx#+1 -py
                ny1=ny#+1 -py
                x=nx*spx
                y=ny*spy
                xx[nx1]=x
                yy[ny1]=y

                for n1 in range(0,m11):
                    for n2 in range(0,m22):
                        n1j=n1+1
                        n2j=n2+1
                        k1=n1*detak1
                        k2=n2*detak2
                        k3=(n1+1)*detak1
                        k4=(n2+1)*detak2
                        sdf=b1*b2*(np.e**(-(b1*k1/2)**2-(b2*k2/2)**2)+np.e**(-(b1*k3/2)**2-(b2*k4/2)**2)+np.e**(-(b1*k1/2)**2-(b2*k4/2)**2)+np.e**(-(b1*k3/2)**2-(b2*k2/2)**2))
```

```
/16.0/np.pi

                    A1=np.sqrt(2*sdf*detak1*detak2)
                    A2=A1#sqrt(2*sdf2*detak1*detak2)

                    g=g+A1*np.cos(k1*x+k2*y+fai1[n1j,n2j])+A2*np.cos(k1*x-k2*y+fai2[n1j,
n2j])

                    intsd=intsd+sdf*detak1*detak2*4

                    if ((n1==0) and (n2==0)):
                        sdf00=sdf
            intsd=0;
            gg[nround,nx1,ny1]=inmiu+g*sq2*sig
            tt[nx1,ny1]=g*sq2
            g=0
tt=(tt*incov+1)*inmiu;
tttrans=np.transpose(tt)
return tttrans
```

Listing 3.1: SRM code

The estimates of the scale of fluctuation in figure 3.2 is given by [Fenton et al., 2008]. For the x-direction:

$$\hat{\rho}(j\Delta\tau_1) = \frac{1}{\hat{\sigma}^2(n_2(n_1-j)-1)} \sum_{k=1}^{n_2} \sum_{i=1}^{n_1-j} (X_{ik})(X_{i+j,k}) \quad \text{for} \quad j = 0,1,\ldots,n_1-1 \tag{3.2}$$

where for the random field sample value x at at coordinate indexes i,k, $X_{ik} = x - \hat{\mu}$, j is the distance between samples and $\Delta\tau_1$ is the increment distance in the grid for a $n_1 \times n_2$ grid.

Similarly for the y-direction:

$$\hat{\rho}(j\Delta\tau_2) = \frac{1}{\hat{\sigma}^2(n_1(n_2-j)-1)} \sum_{k=1}^{n_1} \sum_{i=1}^{n_2-j} (X_{ki})(X_{k,i+j}) \quad \text{for} \quad j = 0,1,\ldots,n_2-1 \tag{3.3}$$

The estimated scale of fluctuation is then the area under the correlation function, symmetric around 0 on the x-axis. For the x-direction and similar for y-direction:

$$\hat{\theta_x} = 2\sum_{j=1}^{n_1-1} \hat{\rho}(j\Delta\tau_1) \quad \text{for} \quad j = 0,1,\ldots,n_1-1 \tag{3.4}$$

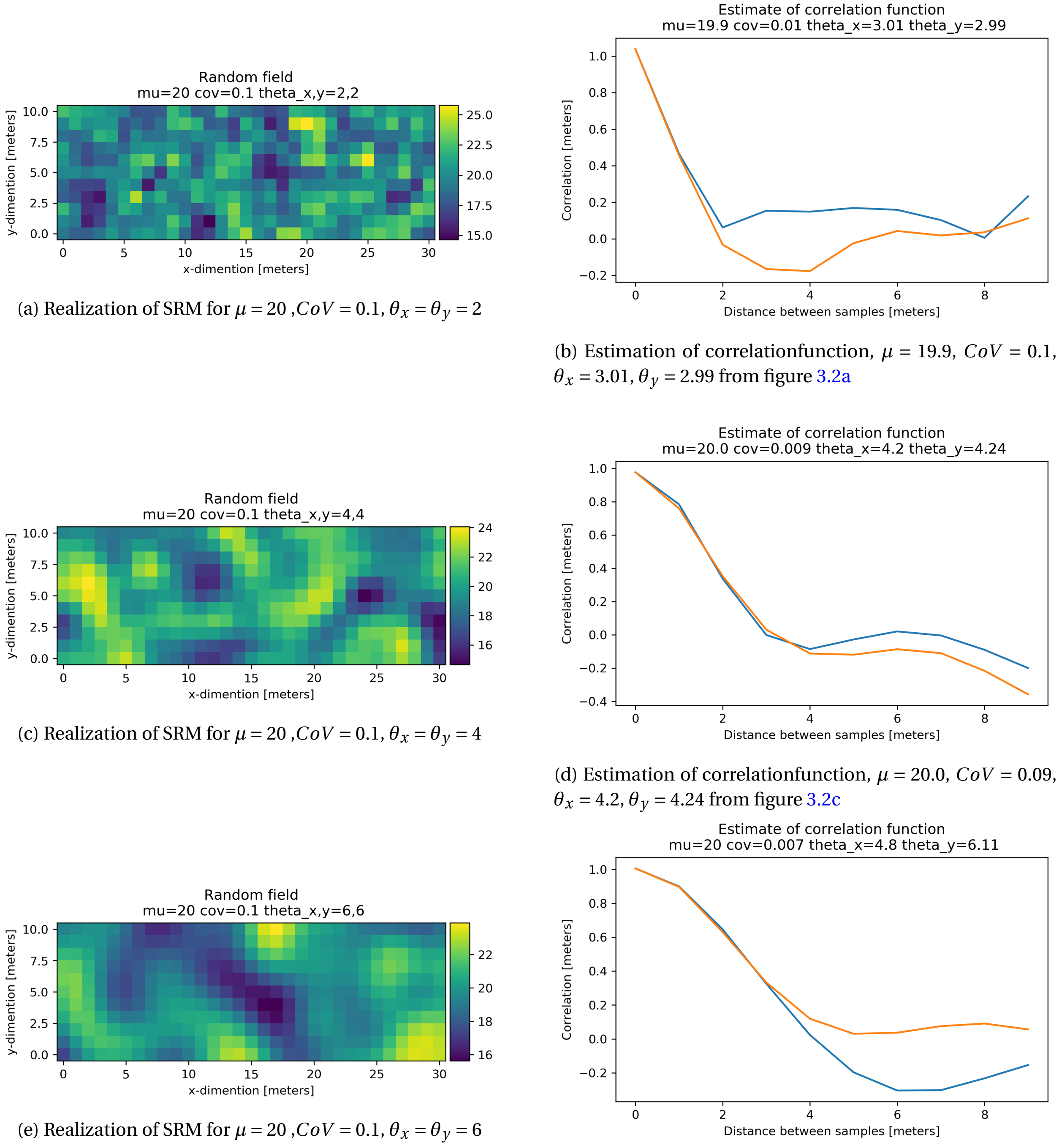
It is assumed that the principal axis of anisotropy is aligned with the x and y directions of the grid and the correlation function is seperable. It is worth noting that in real site exploration, lack of data at scale is most often the case and parameter estimations is only done at a single sampling program of the soil. As a result statistical parameters used to model a real random field can be uncertain and probabilities derived equally so.

```python
varians=np.std(randomfield)**2
mean=np.mean(randomfield)
n1=10
n2=30
rho=np.zeros(10)
for j in range(len(rho)):
    for k in range(n2):
        for i in range(n1-j):
            rho[j]=rho[j]+(randomfield[i,k]-mean)*(randomfield[i+j,k]-mean)

    rho[j]=rho[j]/(varians*(n2*(n1-j)-1))

theta = 2*np.sum(rho)
```

Listing 3.2: Estimating scale of fluctuation in python for a random field. only x-direction shown



(a) Realization of SRM for $\mu = 20$,$CoV = 0.1$, $\theta_x = \theta_y = 2$

(b) Estimation of correlationfunction, $\mu = 19.9$, $CoV = 0.1$, $\theta_x = 3.01$, $\theta_y = 2.99$ from figure 3.2a

(c) Realization of SRM for $\mu = 20$,$CoV = 0.1$, $\theta_x = \theta_y = 4$

(d) Estimation of correlationfunction, $\mu = 20.0$, $CoV = 0.09$, $\theta_x = 4.2$, $\theta_y = 4.24$ from figure 3.2c

(e) Realization of SRM for $\mu = 20$,$CoV = 0.1$, $\theta_x = \theta_y = 6$

(f) Estimation of correlationfunction, $\mu = 20.0$, $CoV = 0.07$, $\theta_x = 4.8$, $\theta_y = 6.11$ from figure 3.2e

Figure 3.2: Three different realizations of random fields created by SRM and estimates of μ, CoV, θ_x, θ_y. Blue curve = x-direction, Orange curve = y-direction. At small distances between samples the correlation function is high and goes towards zero correlation when the distance increase.

3.2 Local averaging

The input parameters in modeling, in the case of RFEM, the mean, standard deviation and spatial correlation length are assumed to be point measures. Therefore when populating a RFEM model, spatial averaging needs to be taken into account, since the element sizes is in general much greater than the size of the sample from where the parameter was derived. Similarlly, when e.g. a clay sample is sheared in the laboratory to determine strength parameters, failure develops over the whole sample when the bonds of the sample yield. The measured strength is a function of the average bond strength of the sample. The greater the sample size the stronger is the averaging effect. Therfore, going from a sample average to a point statistic, averaging needs to be considered.

It can be shown [Vanmarcke, 2010] that the reduction in variance due to local averaging is given by:

$$\sigma_A = \sigma\sqrt{\gamma} \tag{3.5}$$

where σ_A is the new spatially averaged variance that is to be used when drawing samples from the distribution to put into the finite element mesh and γ is the variance reduction function, defined for a rectangular soil element as:

$$\gamma = \frac{4}{T_x^2 T_y^2}\int_0^{T_x}\int_0^{T_y}(T_x-\tau_x)(T_y-\tau_y)\rho(\tau_x,\tau_y)d\tau_x d\tau_y \tag{3.6}$$

where T_x and T_y is the size of the element in the x and y direction respectively, ρ is the correlation function and τ_x and τ_y is the difference between the the x and y coordinates of any two points in the random field (in the interval 0 to T).

γ has the value of 1.0 when $T = 0$ [see Fenton et al., 2008, chapter 3]. Setting $T = \alpha\theta$ (i.e some scalar α times the scale of fluctuation θ) leads to the conclusion that elements much smaller than the scale of fluctuation is affected to small degree by variance reduction.

For the popular simple Markovian correlation function

$$\rho(\tau_x,\tau_y) = e^{-\frac{2\tau_x}{\theta_x}-\frac{2\tau_y}{\theta_y}} \tag{3.7}$$

the integral in equation 3.6 is seperable, and for a 1 by 1 soil element with the same scale of fluctuation $\theta_x = \theta_y = 1$ in both vertical and horizontal direction, the variance reduction factor γ

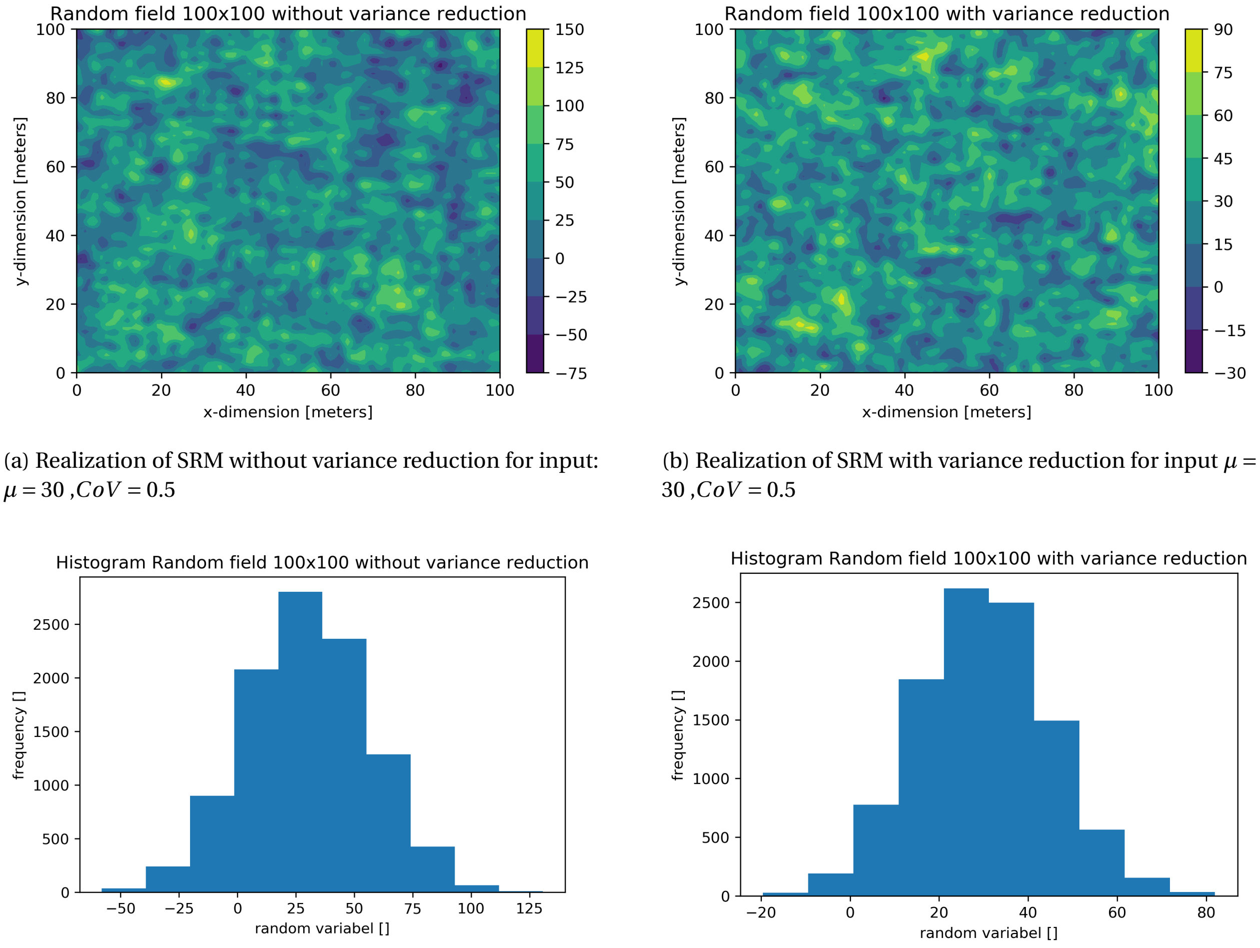
becomes

$$\gamma = \int_0^1 \int_0^1 4(1-\tau_y)(1-\tau_x)exp(-2\tau_x - 2\tau_y)d\tau_y d\tau_x = 0.322 \tag{3.8}$$

As an illustration, figure 3.3 shows two different realizations of a random fields with the same input parameterization, given in table 3.1, and its estimates. Without local averaging the variance is over estimated.

Table 3.1: Impact off variance reduction on a 100 by 100 meter random field sampled with 1 by 1 meters soil elements

Element size	1 by 1 meters		
	Input distribution	Output with variance reduction	Output without variance reduction
μ	30	29.9	30.3
$CoV = \frac{\sigma}{\mu}$	0.5	0.49	0.87
σ	15	14.8	26.4



(a) Realization of SRM without variance reduction for input: $\mu = 30$,$CoV = 0.5$

(b) Realization of SRM with variance reduction for input $\mu = 30$,$CoV = 0.5$

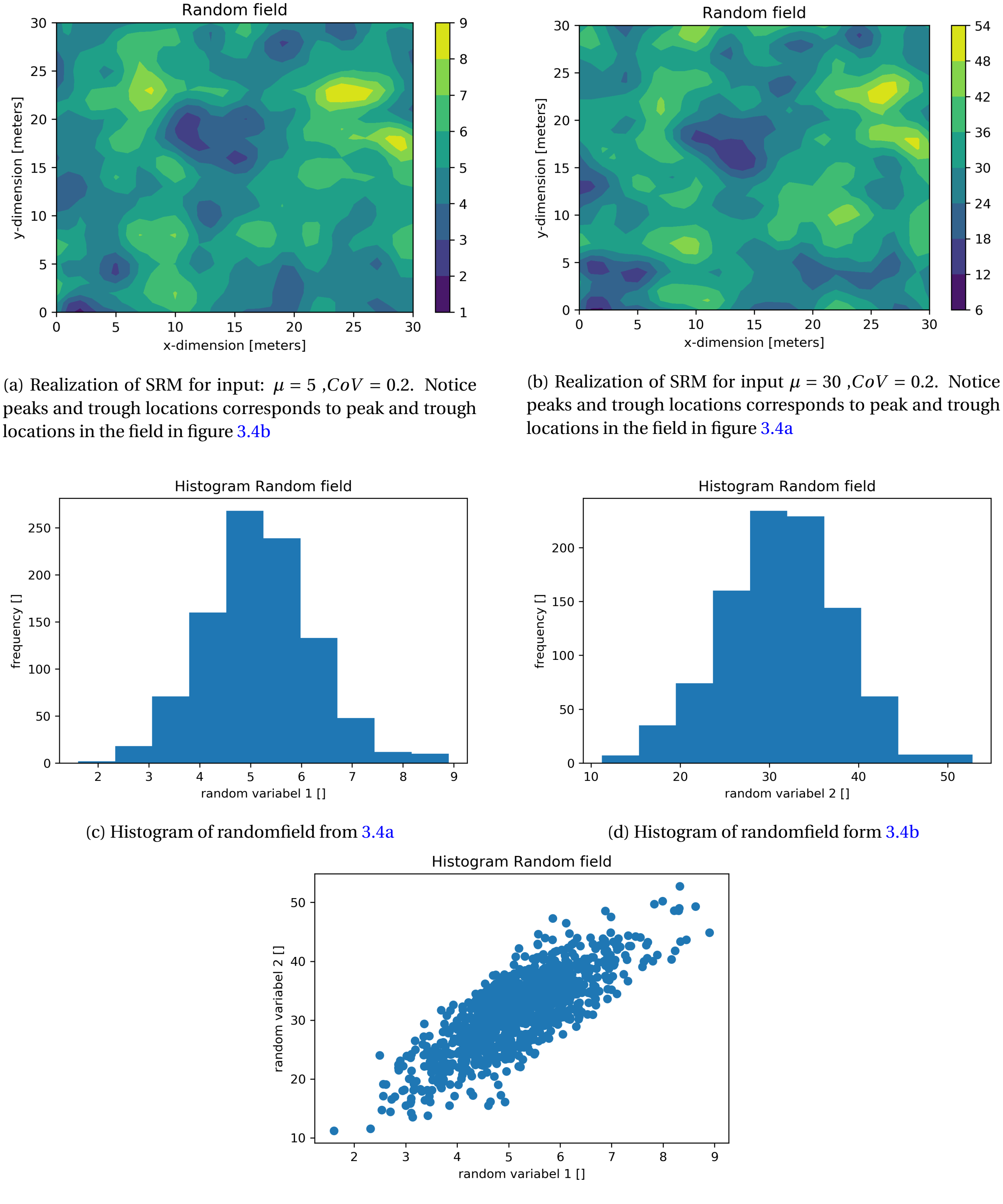
(c) Histogram of randomfield from 3.3a, without variance reduction. Measured $\mu = 30.3$,$CoV = 0.87$

(d) Histogram of randomfield form 3.3b, with variance reduction. Measured $\mu = 29.9$,$CoV = 0.49$

Figure 3.3: Difference of applying local averaging to a random field of size 100 by 100 meters with 1 meter grid cells. Notice differnce in axis and color scales

3.3 Covariance

Soils when tested in the laboratory often show correlations between properties and index parameters, see e.g [Karlsrud and Hernandez-Martinez, 2013]. The pairwise correlation information allows for constraining the uncertainties in parameters that otherwise may not be known or is experience based. [Ching and Phoon, 2012] It will be useful to be able to input random fields that show these correlations into RFEM simulations. Figure 3.4 shows the pyton implementation for two correlated random fields using the numpy multivariate function [Harris et al., 2020].



(a) Realization of SRM for input: $\mu = 5$,$CoV = 0.2$. Notice peaks and trough locations corresponds to peak and trough locations in the field in figure 3.4b

(b) Realization of SRM for input $\mu = 30$,$CoV = 0.2$. Notice peaks and trough locations corresponds to peak and trough locations in the field in figure 3.4a

(c) Histogram of randomfield from 3.4a

(d) Histogram of randomfield form 3.4b

(e) Scatter plot of samples from 3.4a plotted against samples from field in 3.4b shows a linear positive correlation. Each point in in the scatter plot represent the same (x,y)-coordinates from the respective fields.

Figure 3.4: Correlation of two spatially variable random fields.

3.4 Monte Carlo

Monte Carlo is a method that can estimate the means, variances and probabilities of the responses of complex systems to random input [see Fenton et al., 2008, chapter 6.6]. Consider the random response of a system $g(X_1, X_2)$ where X_1 and X_2 are random variables. The system fails if the value of $g(X_1, X_2) > g_{critical}$. Monte Carlo simulates a sequence of realizations of X_1 and X_2, evaluates $g(X_1, X_2)$ and checks if $g(X_1, X_2) > g_{critical}$.

The method is very versatile and can be applied to most kinds of systems. A drawback is that there are no analytical solutions, if the system, e.g. the input is changed, the simulation must be rerun, we can not predict the response to a change in input. Also, to simulate rear events, a lot of simulations are needed which can be computational demanding.

3.5 Plaxis 2D

The following procedure to run the RFEM Monte Carlo analysis is implemented in Plaxis using the python API. The user specify the input soil parameters, the problem geometry and the desired number of realizations. One realization is one stability simulation on one random field. The procedure is run fully autonomous without user interaction.

1. The script starts by creating a new empty Plaxis project
2. Next, the problem geometry is generated i.e. the slope height, inclination and extent
3. Then Plaxis grids the geometry generating the FEM mesh
4. The random field representing the soil property is generated based on the user specified mean, standard deviation (or CoV) and scale of fluctuation
5. The soil elements are populated with the soil parameter values from the random field
6. Next a new phase is added after the initial phase, activating the geometry represented by the soil
7. Finally the deformations are calculated and a result is stored, i.e. failure or no failure in a slope stability problem

8. Now step 4 to 7 is repeated generating a new realization of the random field, a user specified number of times

9. In the end statistics are gathered and written to a file, and plots of the results can be displayed

Chapter 4

Results

To investigate the geometry of the failure mechanism and to test the validity of the python Plaxis 2D API implementation, a series of simulations are run. The results from these simulations are presented in the following sections.

4.1 "Biaxial cell" Tresca failure criterion - Homogeneous isotropic soil - Zero variance

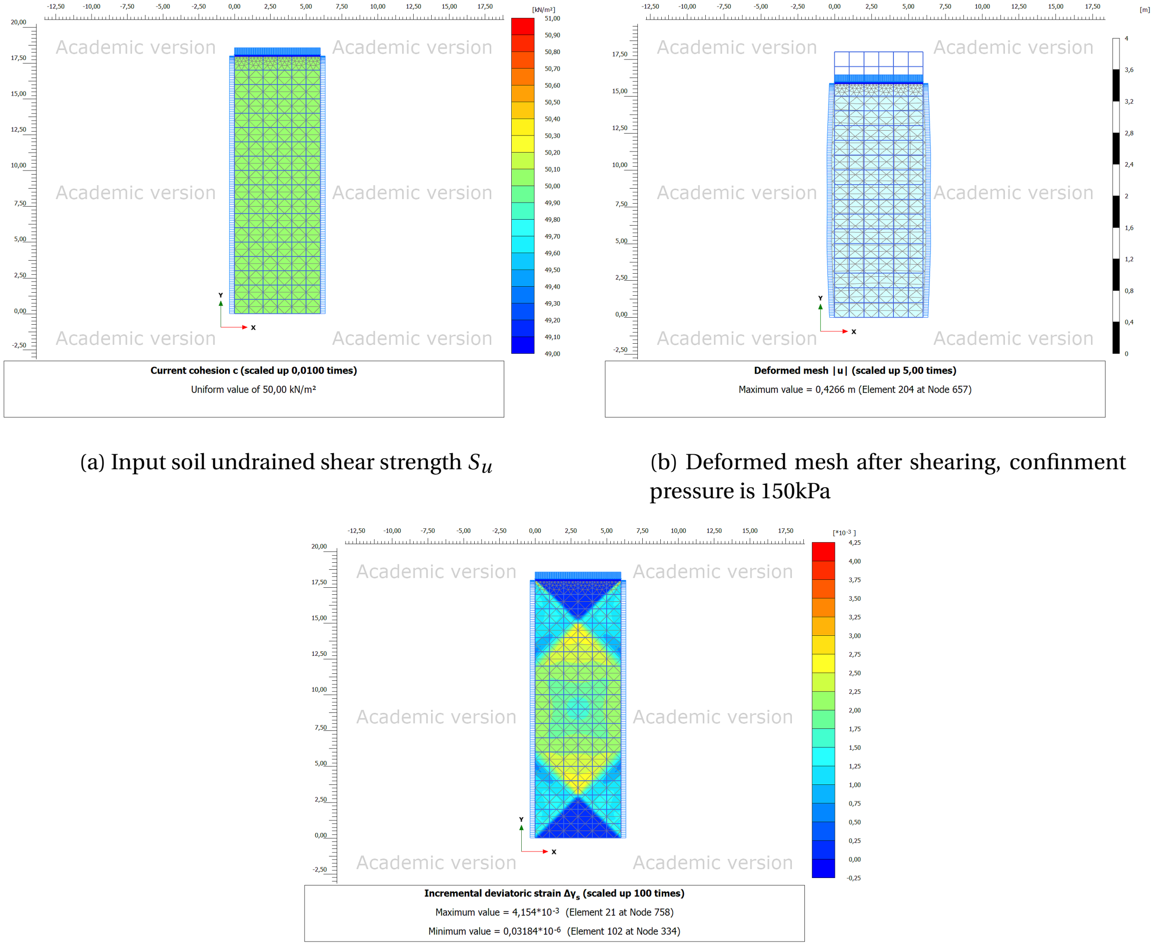
The idealization done in many formulations of geotechnical problem descriptions and designs, namely assigning an average value to a soil profile, was investigated by Ching and Phoon [2013]. Some of the results of the study is reproduced here. The goal is to show that the slip curve governs the strength of the soil sample and that the slip curve trajectory depends on the realization of the random soil field.

As a control, a homogeneous isotropic soil with zero variance is tested to verify the model based on known analytical solutions. See table 4.1 for input parameters. Experiment is done in a staged construction in Plaxis 2D, first isotropically consolidating the sample, then increasing the axial applied load pressure to failure. The simulation is repeated for three different confining pressures, 50, 100 and 150 kPa. The sample is 18 meters high and 6 meters wide. The soil cells are 1 meters by 1 meters. The FEM grid is done automatically by the software. The boundaries on the side are free to move, the top and bottom are horizontally fixed. See figure 4.1 for mesh detail and experiment geometry. Figure 4.2 shows stress strain curves for the three simulations. Figure 4.3 shows the stress paths in a "NGI-plot".

The NGI plot shows a predicted undrained shear strength of 50 kPa and no dependence of the confinement stress on the shear strength. The result is in accordance with equation 2.1 and shows a good fit to theory and good ability to predict input strength.

Table 4.1: Soil parameters for homogeneous isotropic soil in biaxial Tresca test

Soil model	Mohr Coulomb - Undrained(B)		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma_{unsat}$	0 kN/m^3	0	-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, ν	0.3	0	-
Undrained Shear Strength,S_u	50 kPa	0	-



(a) Input soil undrained shear strength S_u

(b) Deformed mesh after shearing, confinment pressure is 150kPa

(c) Sample incremental deveatoric strain just before failure

Figure 4.1: Biaxial isotropic compression test on uniform soil with a Tresca failure criterion

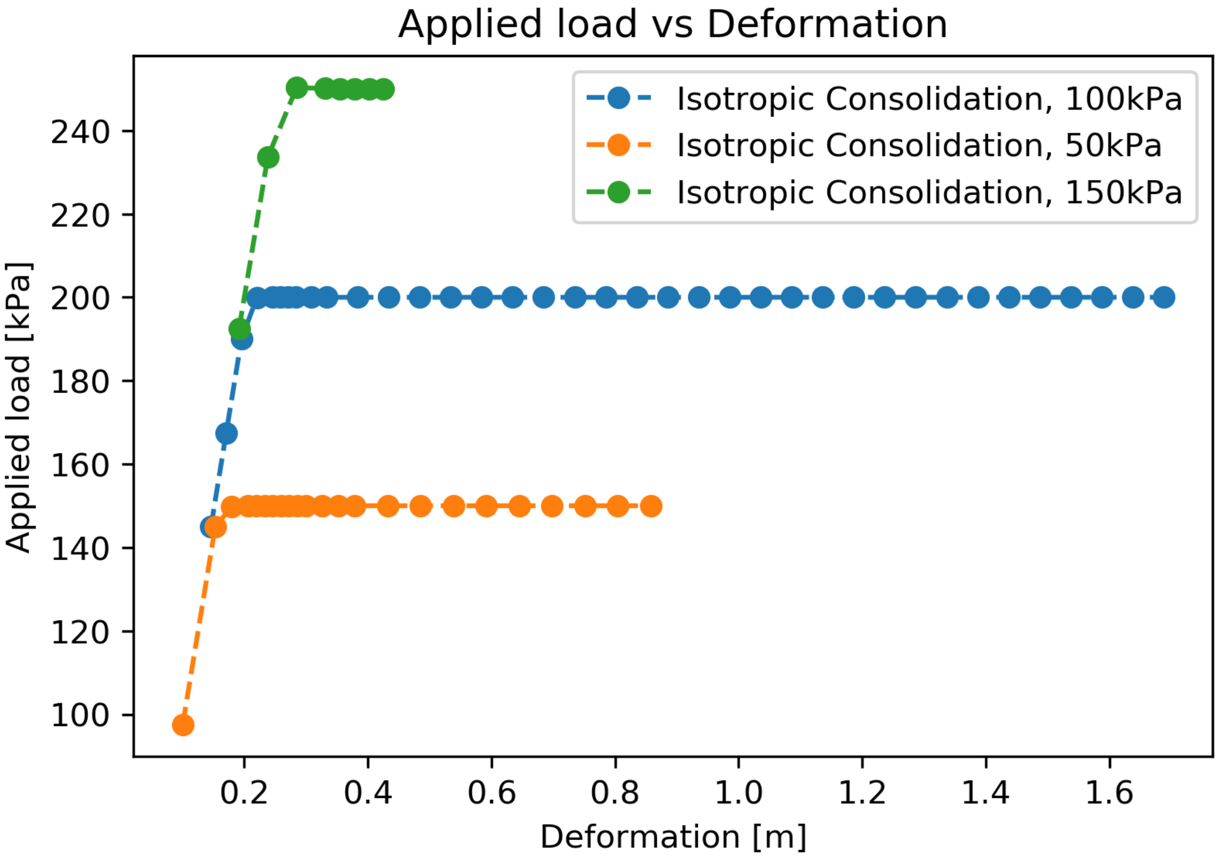


Figure 4.2: Applied loads vs vertical displacement for the three isotropically confined compression tests with a Tresca failure criterion.

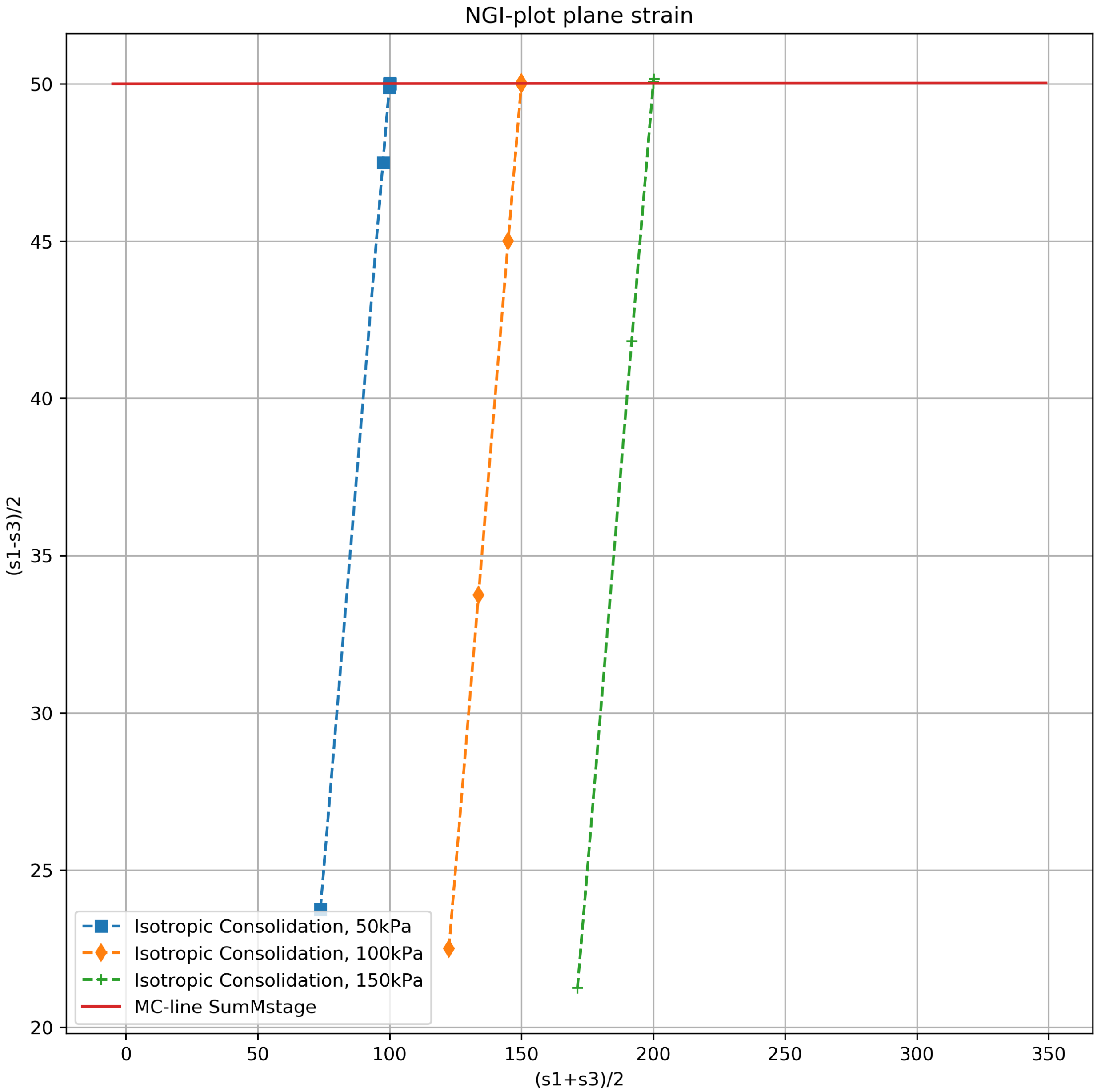


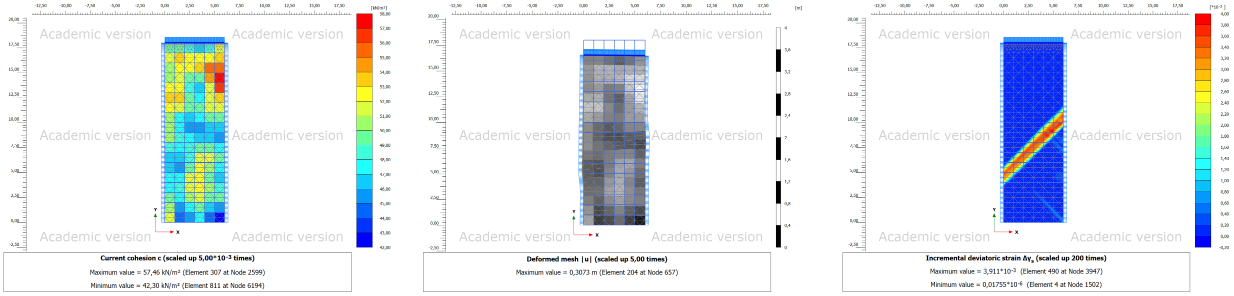
Figure 4.3: "NGI-plot" of three isotropically confined compression tests with a Tresca failure criterion.

4.2 "Biaxial cell" Tresca failure criterion - Spatially varying soil

Spatial variability is introduced in the setup from section 4.1. Table 4.2 shows the input parameters. 300 realizations of a spatially varying random soil strength field is run. Figure 4.4 shows the FEM mesh and geometry for a single realization. Figure 4.5 shows stress strain curves and figure 4.6 shows a "NGI-plot" of a single realization of the random field. The failure slip surface is estimated by taking the FEM nodes with the highest incremental deveatoric strain at the moment just before failure, see figure 4.4c and 4.7. These nodes locations are used to look up the input soil strength at these nodes and averaged to estimate the slip surface shear strength. This line average is compared to the domain average, which is calculated from the average strength of all the input nodes in the FEM mesh. The Mobilized strength is calculated from the theory and the applied load. Figure 4.8 shows the 300 results plotted together reproduces the results of Ching and Phoon [2013] well and shows that the line average estimated strength is much closer to the actual mobilized strength than the domain averaged strength, indicating that the domain average simplification can be a poor idealization.

Table 4.2: Soil parameters for spatially varying soil in biaxial Tresca test

Soil model	Mohr Coulomb - Undrained(B)		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma_{unsat}$	0 kN/m^3	0	-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, ν	0.3	0	-
Undrained Shear Strength,S_u	50 kPa	0.02	5,5 $meters$


Academic version
Current cohesion c (scaled up 5,00*10-3 times)
Maximum value = 57,46 kN/m² (Element 307 at Node 2599)
Minimum value = 42,30 kN/m² (Element 811 at Node 6194)
Deformed mesh |u| (scaled up 5,00 times)
Maximum value = 0,3073 m (Element 204 at Node 657)
Incremental deviatoric strain Δγs (scaled up 200 times)
Maximum value = 3,911*10-3 (Element 490 at Node 3947)
Minimum value = 0,01755*10-6 (Element 4 at Node 1502)

(a) Input soil undrained shear strength S_u. Domain average strength estimated to 49.6kPa

(b) Deformed mesh after shearing, confinment pressure is 150kPa

(c) Sample incremental deveatoric strain just before failure, indicating the failure surface

Figure 4.4: Biaxial isotropic compression test on spatially variable soil with a Tresca failure criterion

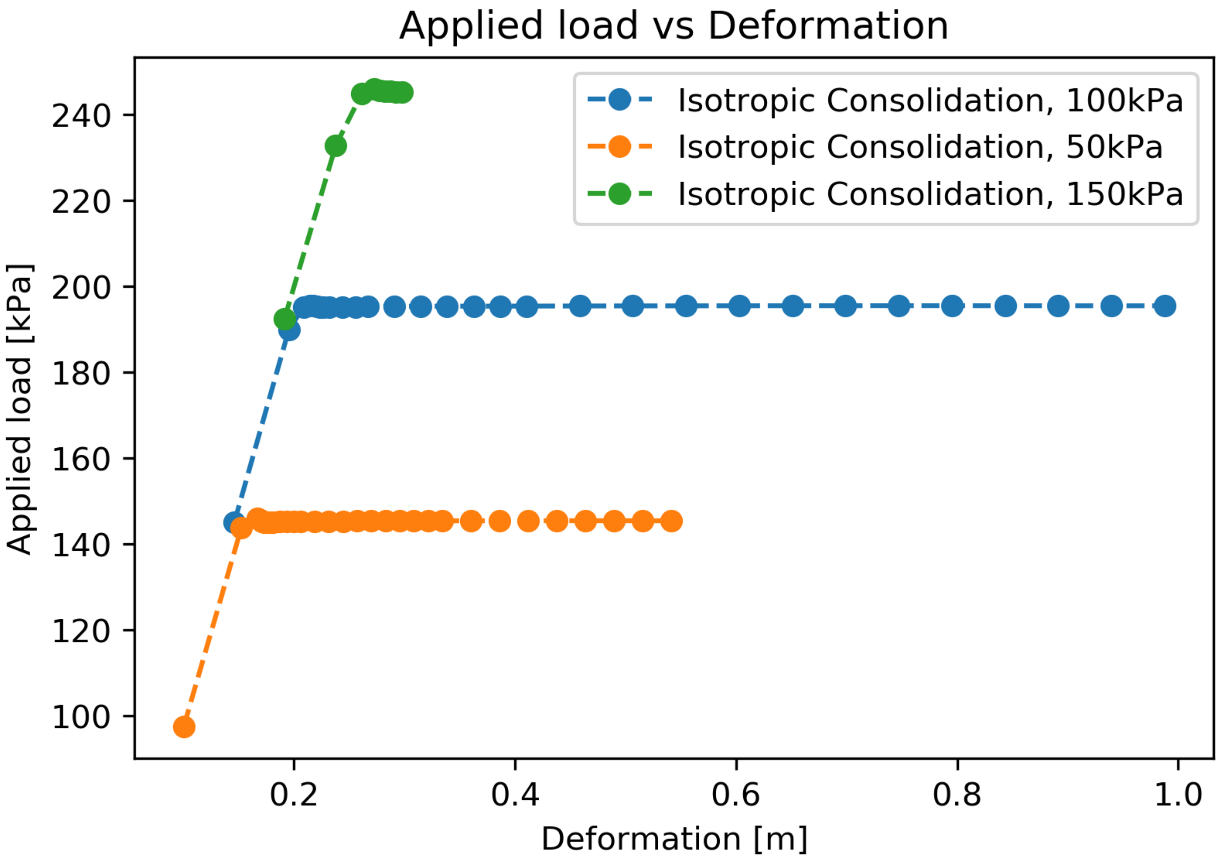


Figure 4.5: Applied loads vs vertical displacement for the three isotropically confined compression tests with a Tresca failure criterion.

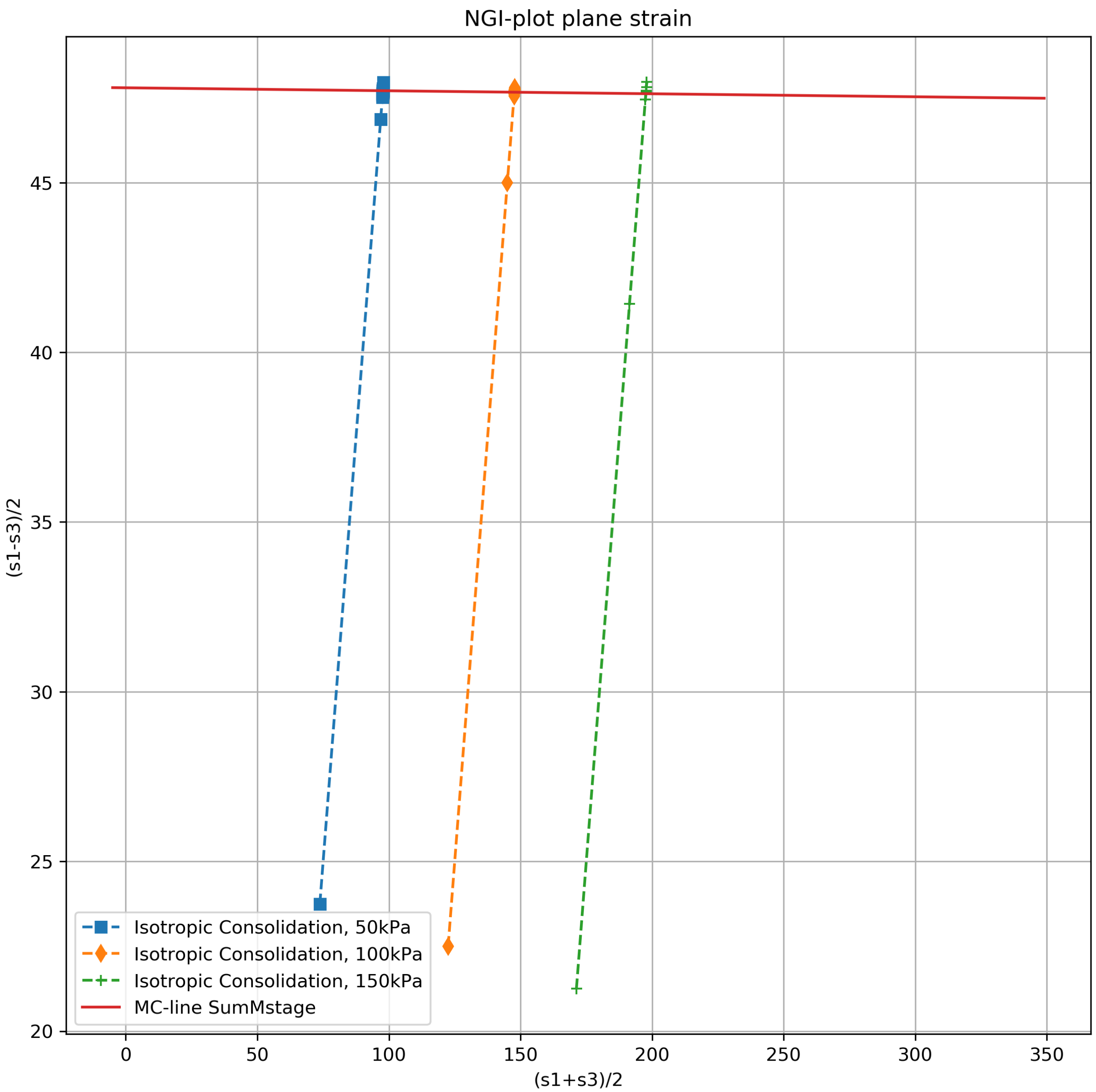


Figure 4.6: "NGI-plot" of isotropically confined compression tests with a Tresca failure criterion. The same spatially varying soil sample as in figure 4.4 at three difrent confinement pressures. Mobilized strength estimated to $47.7kPa$

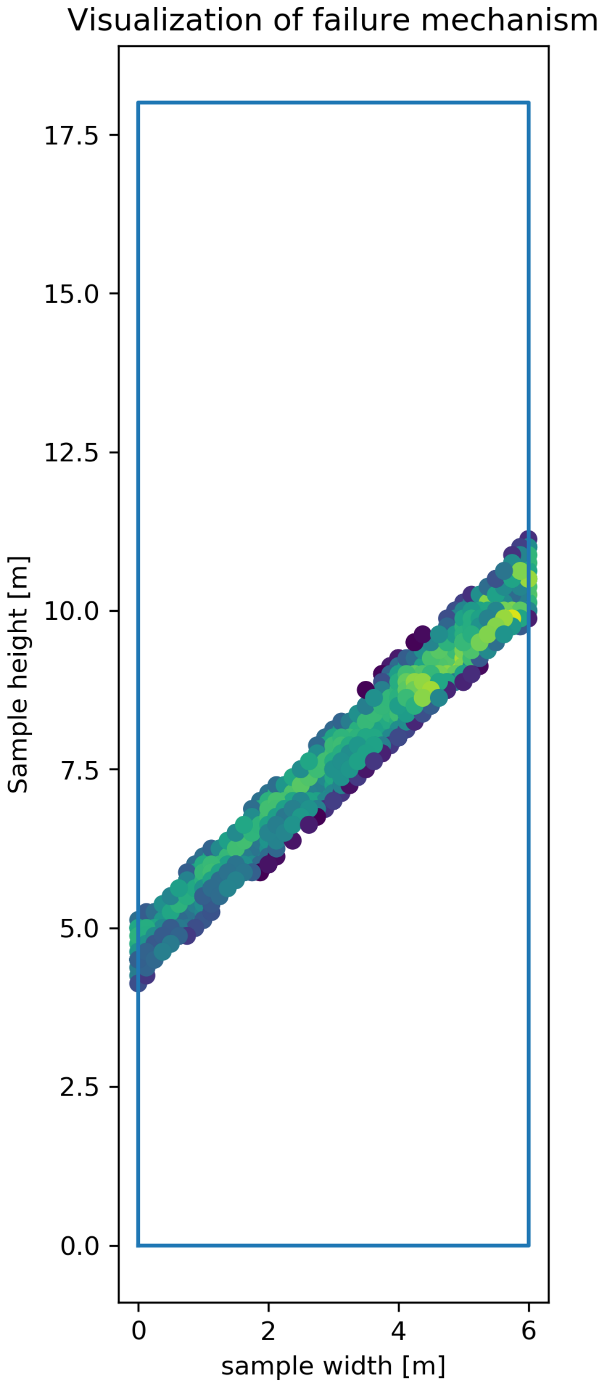


Figure 4.7: Python extraction of failure band. Line average strength estimated to $47.6kPa$

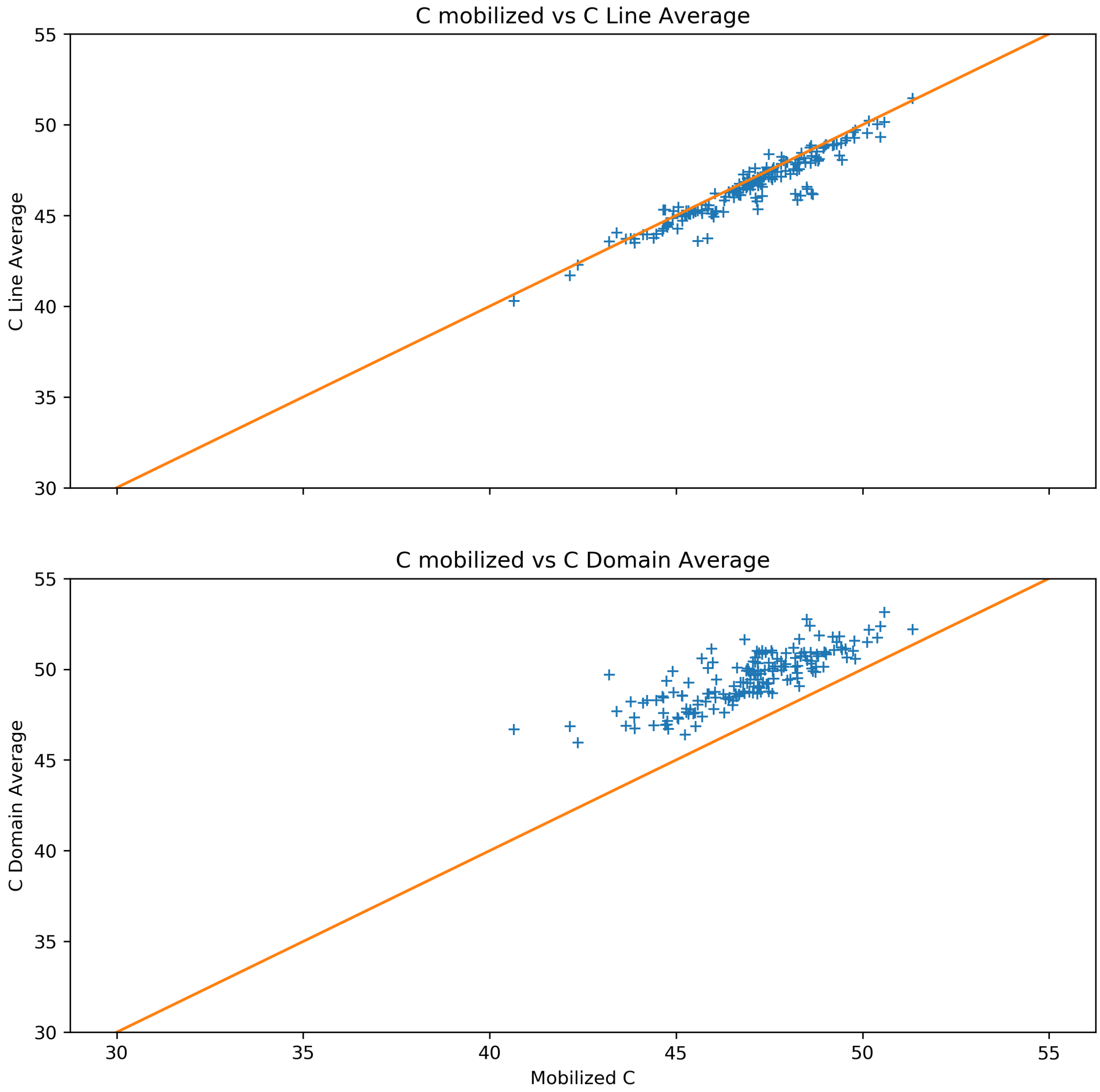


Figure 4.8: Results of 200 iterations showing that the line average strength is a better estimate of the soil sample strength than the domain average

4.3 "Biaxial cell" Mohr-Coulomb failure criterion - Homogeneous isotropic soil

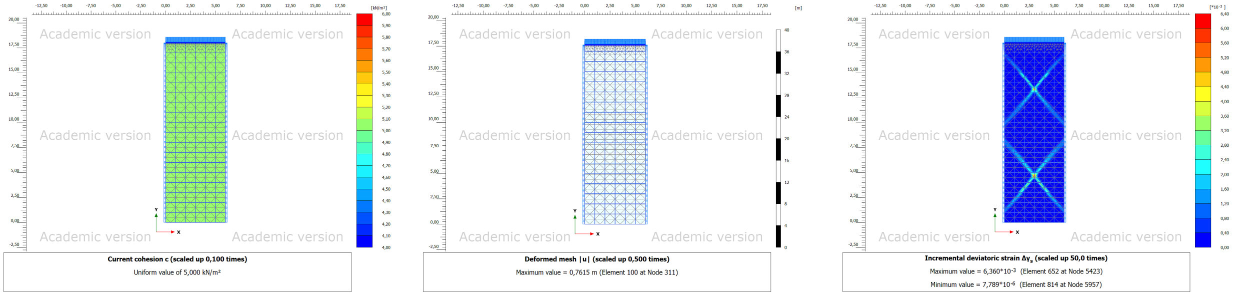
To the authors knowledge there has been no study veryfing if the results from section 4.2 and from papers like Ching and Phoon [2013] holds for simple stress cases in $c-\phi$ soils.

The set up from section 4.1 is repeated, this time with input according to table 4.3 See figure 4.9 for mesh detail and experiment geometry. Figure 4.10 shows stress strain curves for the three simulations. Figure 4.12 shows the extracted failure band. Figure 4.11 shows the stress paths in a "NGI-plot".

The measured mobilized friction angle is 29.7 °and fits well with the input friction angle, the measured mobilized cohesion is 4.6 kPa, lower than the input value of 5 kPa.

Table 4.3: Soil parameters for hogeneous isotropic soil in biaxial MC test

Soil model	Mohr Coulomb - drained		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma_{unsat}$	0 kN/m^3	0	-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, ν	0.3	0	-
Friction angle, ϕ	30 °	0	-
Cohesion, c	5 kPa	0	-



(a) Input soil cohesion. Domain average cohesion and phi estimated to $c = 5.0kPa$ and $\phi = 30.0$

(b) Deformed mesh after shearing, confinment pressure in this figure is 150kPa

(c) Sample incremental deveatoric strain just before failure

Figure 4.9: Biaxial isotropic compression test on spatially variable soil with a Mohr-Coulomb failure criterion

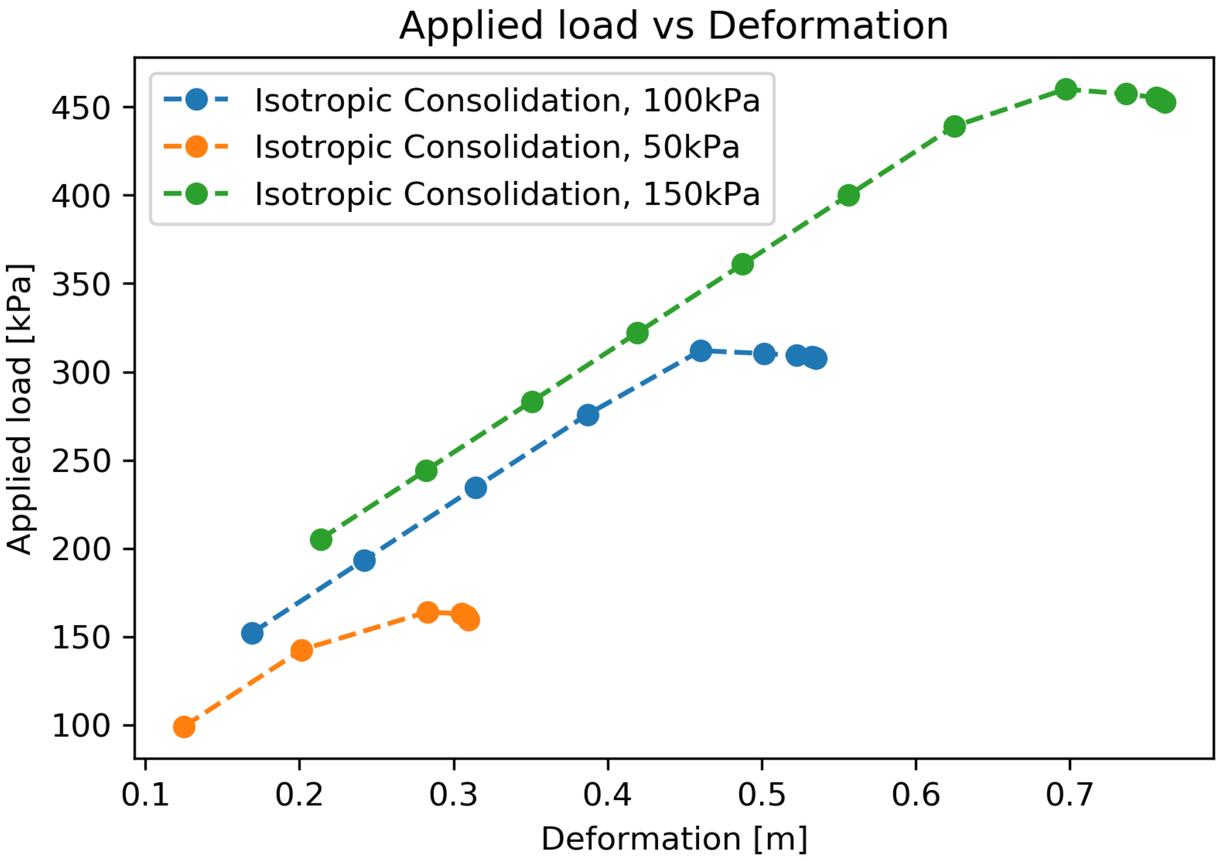


Figure 4.10: Applied loads vs vertical displacement for the three isotropically confined compression tests with a Mohr-Coulomb failure criterion.

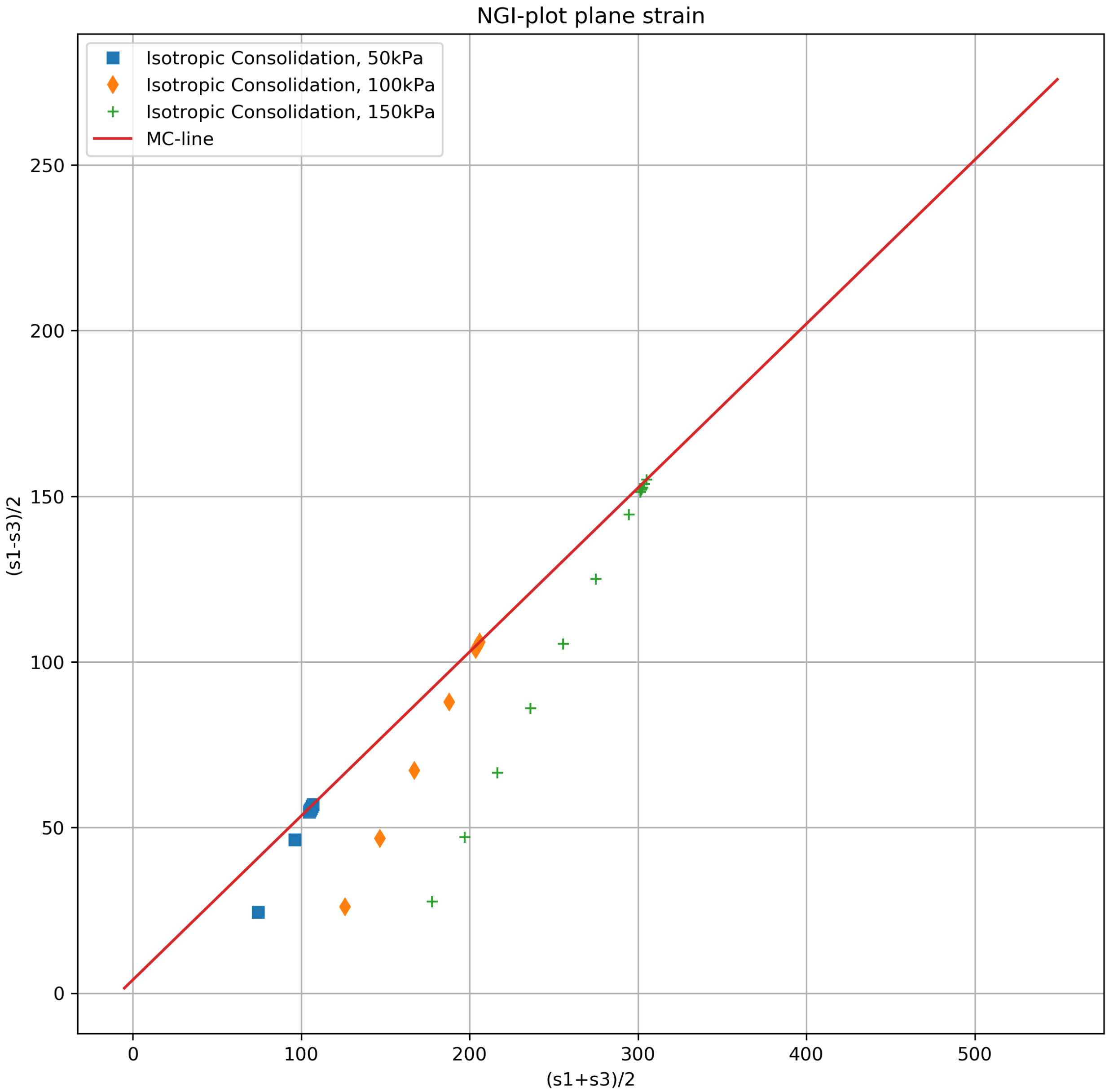


Figure 4.11: "NGI-plot" of isotropically confined compression tests with a Tresca failure criterion. The same spatially varying soil sample as in figure 4.13 at three difrent confinement pressures. Mobilized c and phi estimated to $c = 4.6kPa$ and $\phi = 29.7$

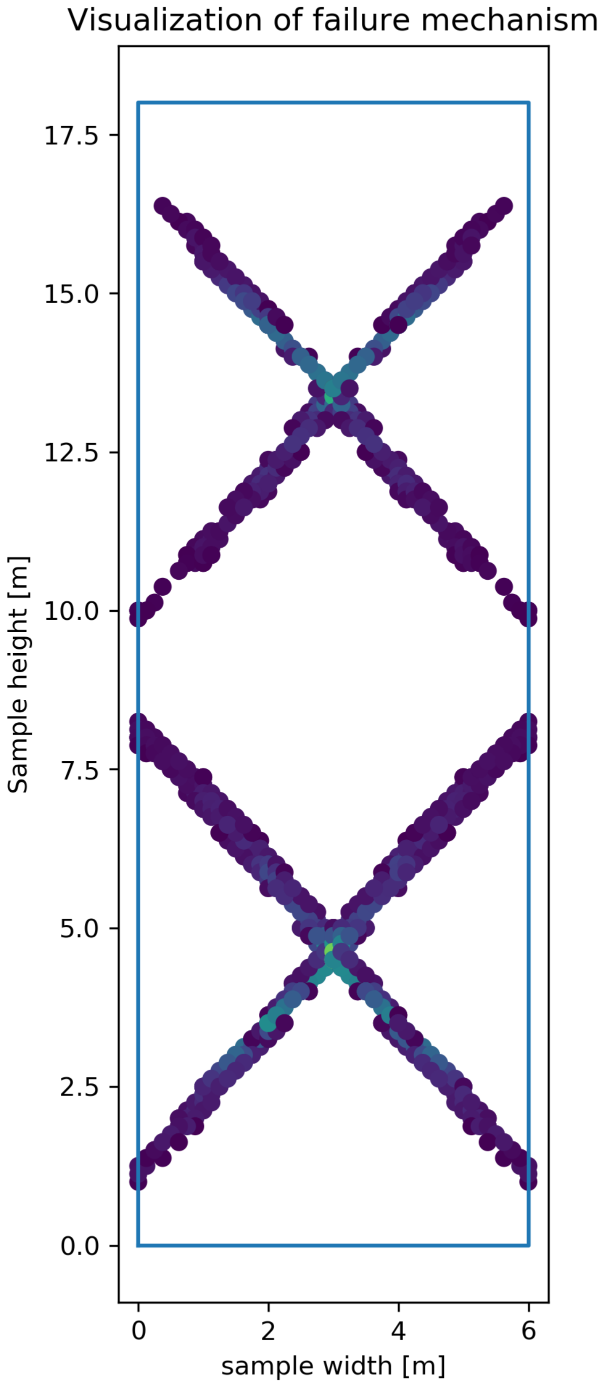


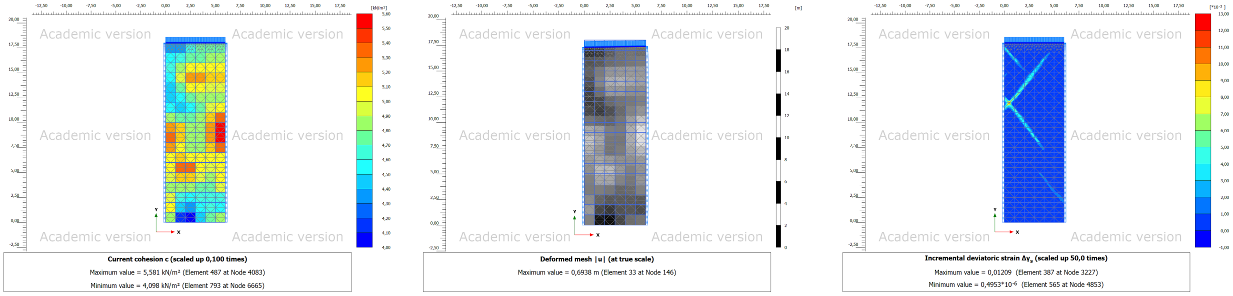
Figure 4.12: Python extraction of failure band. Line average c and phi estimated to $c = 5.0kPa$ and $\phi = 30.0$

4.4 "Biaxial cell" Mohr-Coulomb failure criterion - Spatially varying soil

Spatial variability is introduced in the setup from section 4.3. Table 4.4 shows the input parameters. The input random variabels, the cohesion and the friction angle, is positively linearly correlated (see section 3.3). 300 realizations of a spatially varying random soil strength field is run. Figure 4.13 shows the FEM mesh and geometry for a single realization. Figure 4.14 shows stress strain curves and figure 4.15 shows a "NGI-plot" of a single realization of the random field. The failure slip surface is estimated as explained in 4.2, see figure 4.13c, 4.16 and 4.17. The failure slip surface nodes locations are used to look up the input soil strength at these nodes and averaged to estimate the slip surface friction angle and cohesion. This line average is compared to the domain average, which is calculated from the c and ϕ of all the input nodes in the FEM mesh. The Mobilized c and ϕ is calculated from the theory and the applied load. Figure 4.18 shows the 300 results plotted together and shows that the line average estimated friction angle is much closer to the actual mobilized friction angle than the domain averaged friction angle. Figure 4.19 shows that the simulations was also able, although the results is more noisy, to find the same relationship between the input cohesion, slip surface averaged cohesion and the domain average cohesion as for the friction angle and undrained strength cases. The input pairwise c-phi correlation is also to some degree reproduced in the measurments as can be seen in figure 4.20.

Table 4.4: Soil parameters for spatially varying soil in biaxial MC test

Soil model	Mohr Coulomb - drained		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma_{unsat}$	0 kN/m^3	0	-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, v	0.3	0	-
Friction angle, ϕ	30 °	0.02	5,5 $meters$
Cohesion , c	5 kPa	0.02	5,5 $meters$
c and ϕ positive linear correlated			


Academic version
Current cohesion c (scaled up 0,100 times)
Maximum value = 5,581 kN/m² (Element 487 at Node 4083)
Minimum value = 4,098 kN/m² (Element 793 at Node 6665)
Deformed mesh |u| (at true scale)
Maximum value = 0,6938 m (Element 33 at Node 146)
Incremental deviatoric strain Δγs (scaled up 50,0 times)
Maximum value = 0,01209 (Element 387 at Node 3227)
Minimum value = 0,4953*10-6 (Element 565 at Node 4853)

(a) Input soil cohesion. Domain average cohesion and phi estimated to $c = 4.9kPa$ and $\phi = 29.1$

(b) Deformed mesh after shearing, confinement pressure in this figure is 150kPa

(c) Sample incremental deveatoric strain just before failure, indicating the slip surface

Figure 4.13: Biaxial isotropic compression test on spatially variable soil with a Mohr-Coulomb failure criterion

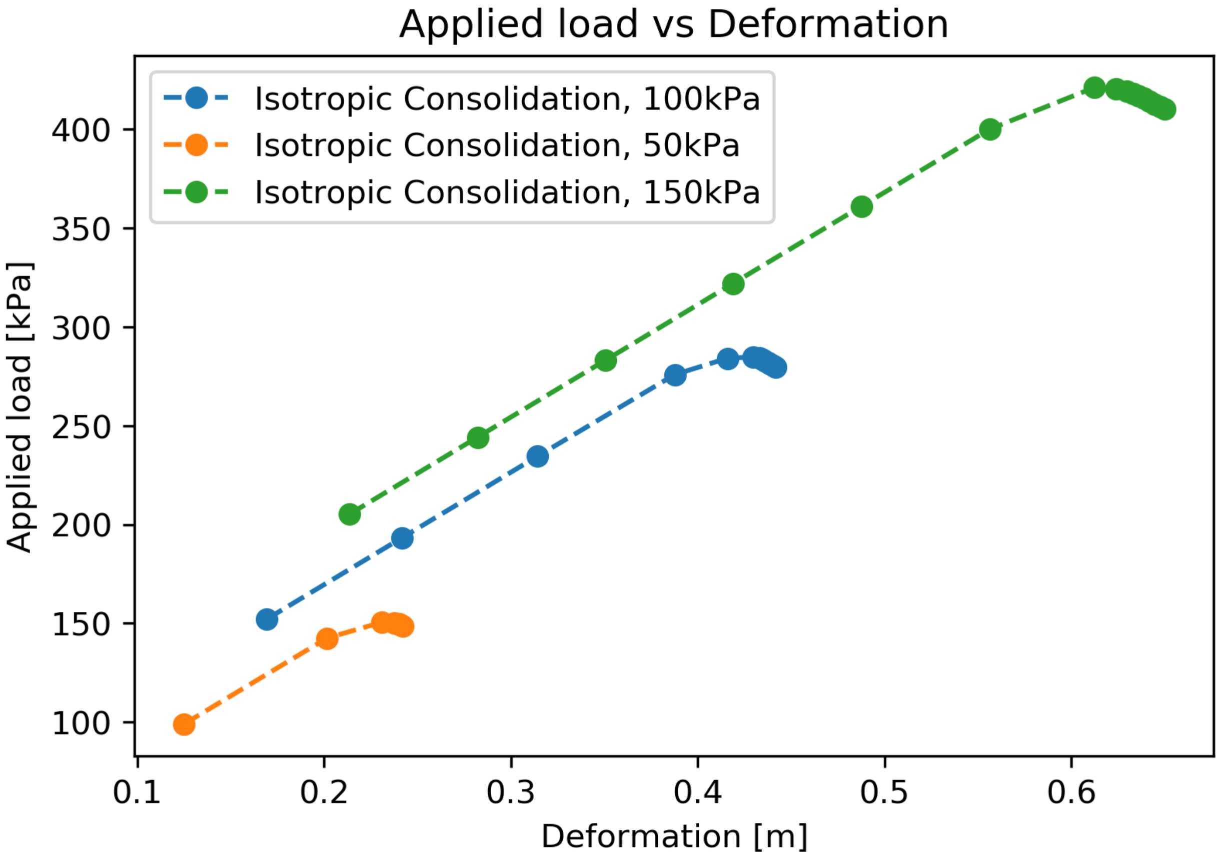


Figure 4.14: Applied loads vs vertical displacement for the three isotropically confined compression tests with a Mohr-Coulomb failure criterion.

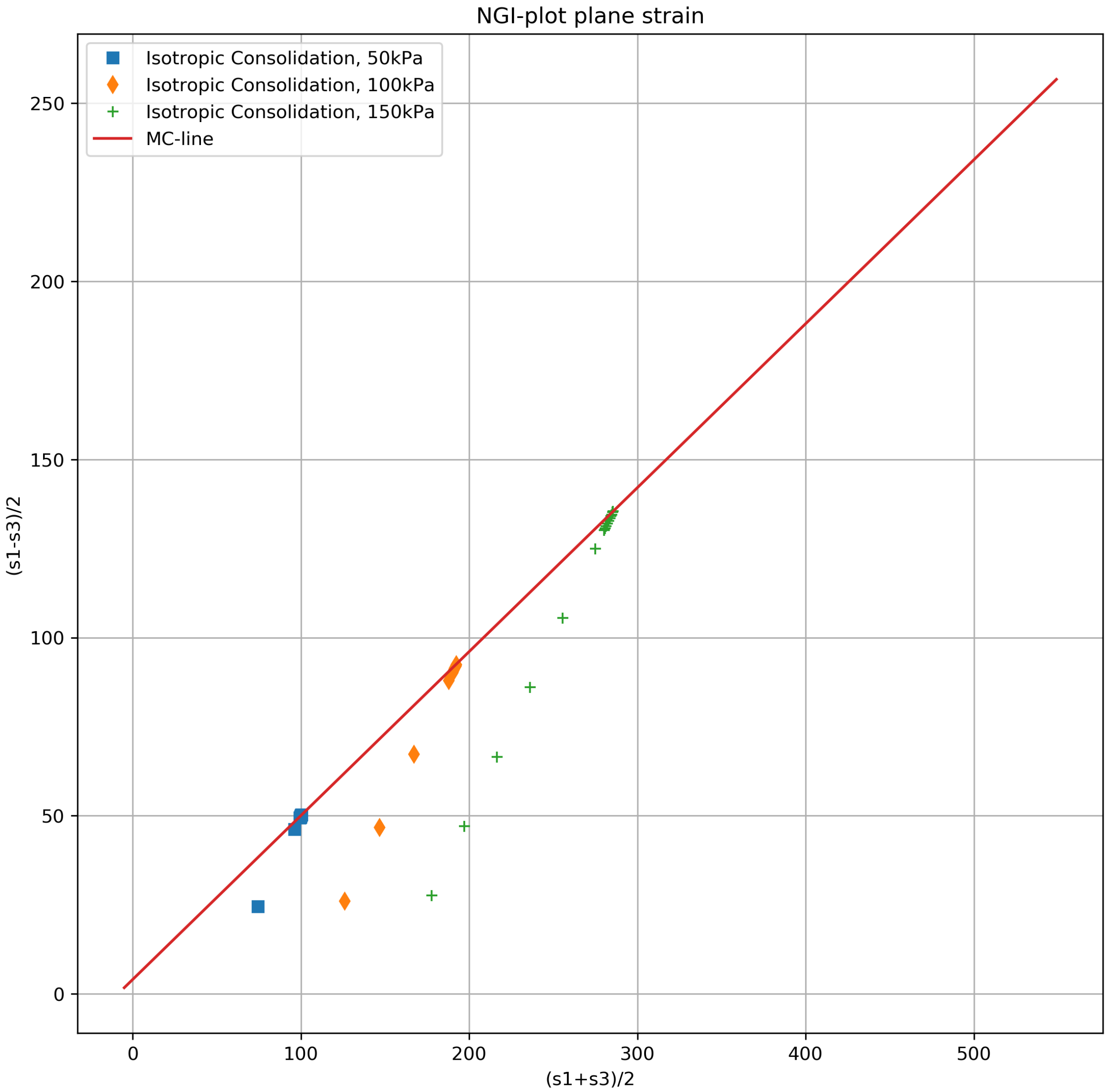


Figure 4.15: "NGI-plot" of isotropically confined compression tests with a Tresca failure criterion. The same spatially varying soil sample as in figure 4.13 at three difrent confinement pressures. Mobilized c and phi estimated to $c = 4.5kPa$ and $\phi = 27.4$

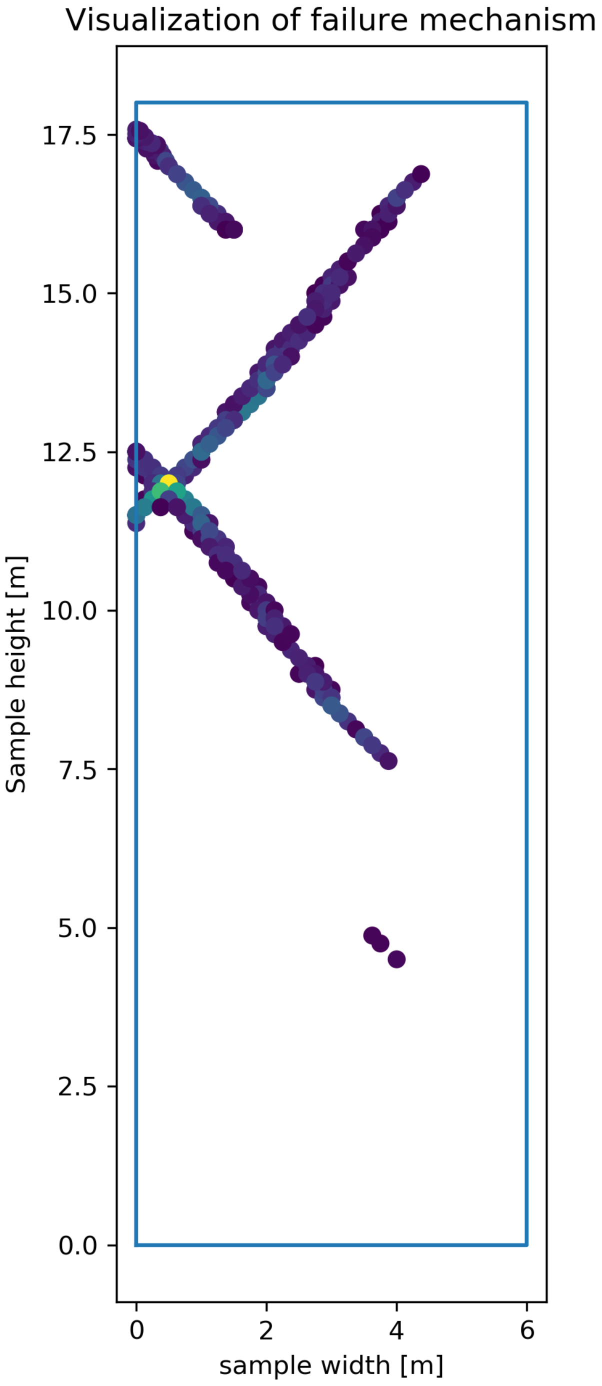


Figure 4.16: Python extraction of failure band. Line average c and phi estimated to $c = 4.7kPa$ and $\phi = 27.6$

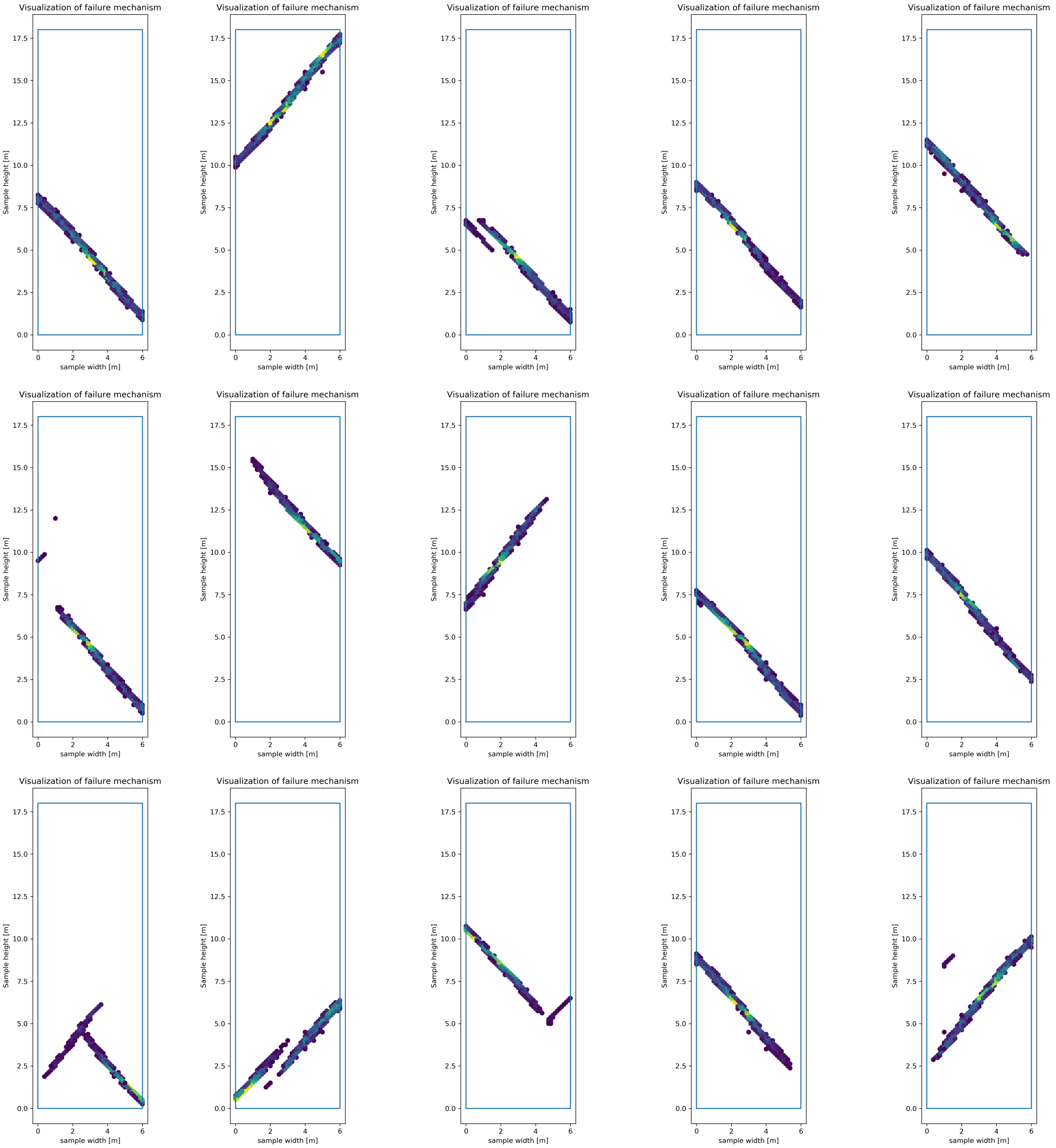


Figure 4.17: 15 extracted slip surfaces from the 300 realizations showing variation in failure surfaces. The lines are in most cases fairly straight. They seem to be flatter towards the edges of the sample and steeper in the middle.

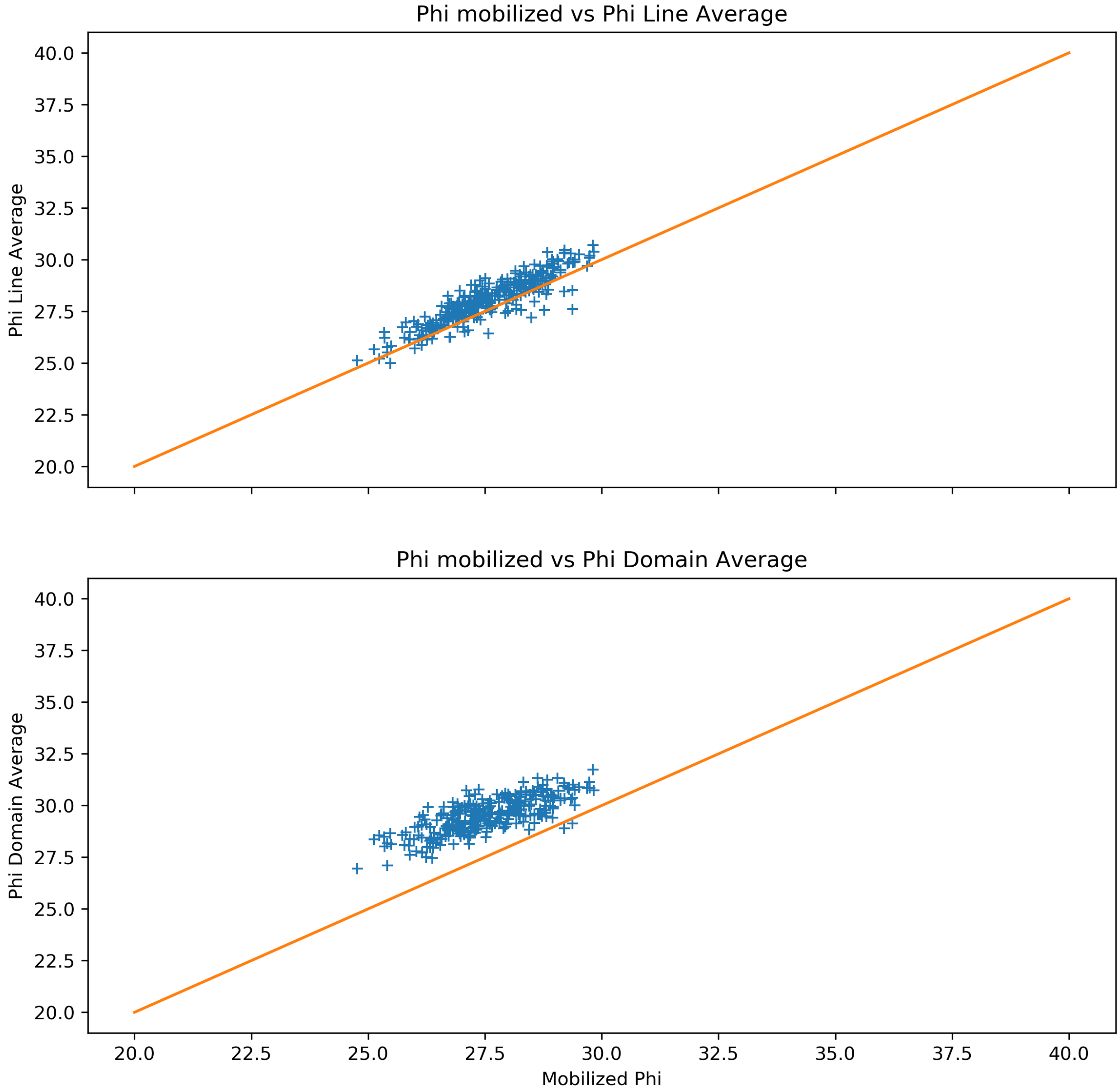


Figure 4.18: Results of 300 iterations showing that the line average shear strength is a better estimate of the soil sample friction angle than the domain average also for the case of a Mohr-Coulomb failure criteria

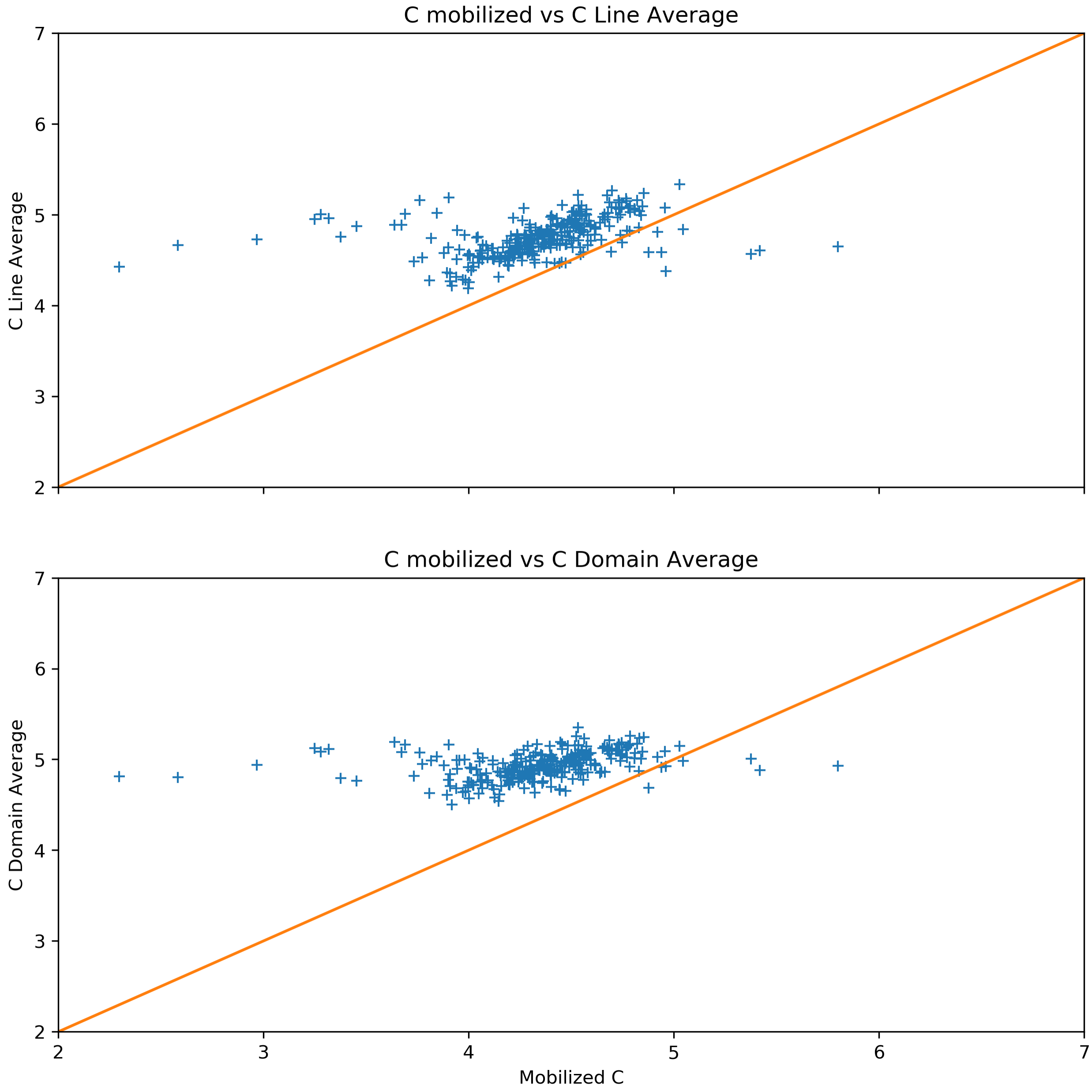
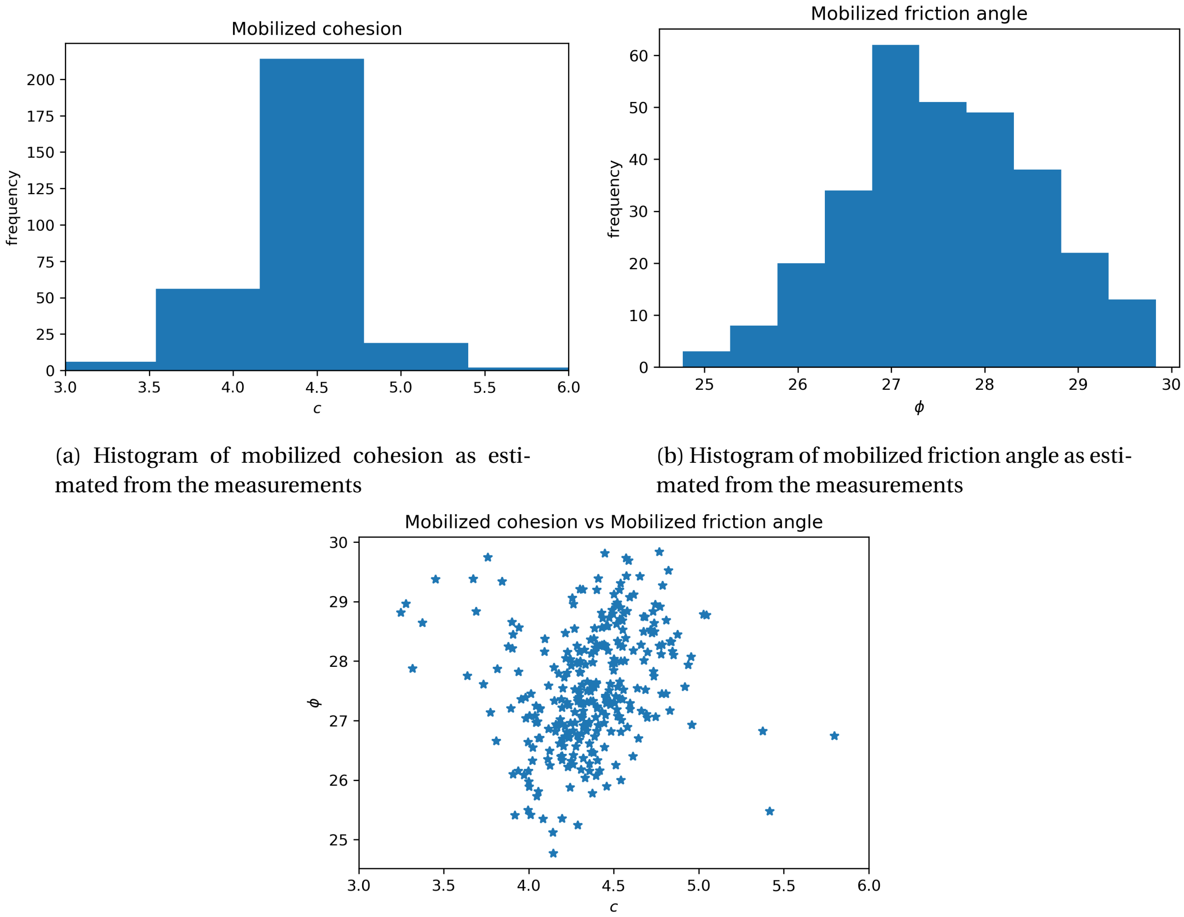


Figure 4.19: Results of 300 iterations showing that the line average cohesion estimate does correlate with the Mohr-Coulomb interpreted cohesion from the mobilized results of the biax test, but the results are noisy



(a) Histogram of mobilized cohesion as estimated from the measurements

(b) Histogram of mobilized friction angle as estimated from the measurements

(c) cross plot Mobilized cohesion vs Mobilized friction angle, some evidence of the input correlation can be seen in these results

Figure 4.20: Output mobilized / measured estimates of cohesion and friction angle for 300 spatially variable soil samples undergoing isotropically consolidated compression biaxial tests.

4.5 Slope Stability problem - Homogeneous isotropic soil - Zero variance

To check the base case of spatially invariant soil with known analytical solutions. This is to validate the input to the random field generation, the FEM mesh and calculation. The simulation is a elasto-plastic FEM simulation with Mohr-Couloumb material model using Plaxis undrained(C) behaviour. 15-Node triangular FEM elements are used. The slope is 5 meter high with a 2:1 gradient. The soil parameters are presented in Table 4.5. The resulting random field, or in this particular case a constant field, is shown in Figure 4.21 and the 2:1 slope geometry and Plaxis 2D mesh is shown in Figure 4.22.

Table 4.5: Soil parameters for Homogeneous isotropic soil

Soil model	Mohr Couloumb - Undrained(C)		
ooo Parameters	Input distribution μ	Output with variance reduction $CoV = \frac{\sigma}{\mu}$	Output without variance reduction θ
μ	20 kN/m^3	0	-
$CoV = \frac{\sigma}{\mu}$	10 MPa	0	-
σ	0.49	0	-

Running a Plaxis c-reduction calculation phase, result graph plotted in 4.23, on the uniform soil slope gives a Factor of Safety, $F_s = 1.14$. The corresponding F_s obtained by slope stability charts after Janbu [1968] is $F_s = 1.16$. Figure 4.24 shows the classical circular failure mechanism.

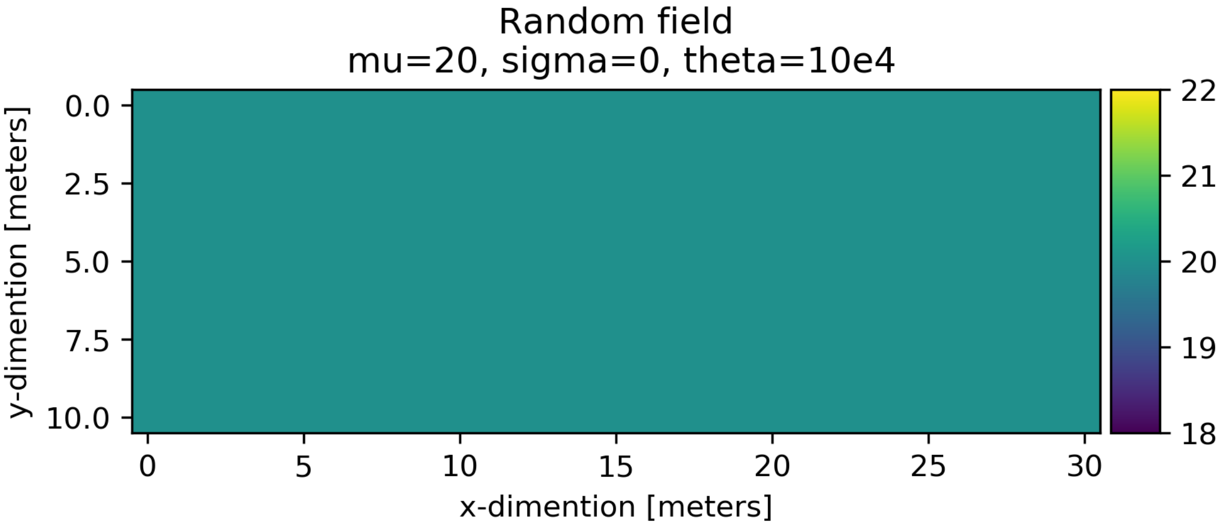


Figure 4.21: S_u Random field, in this particular case the soil strength is uniform

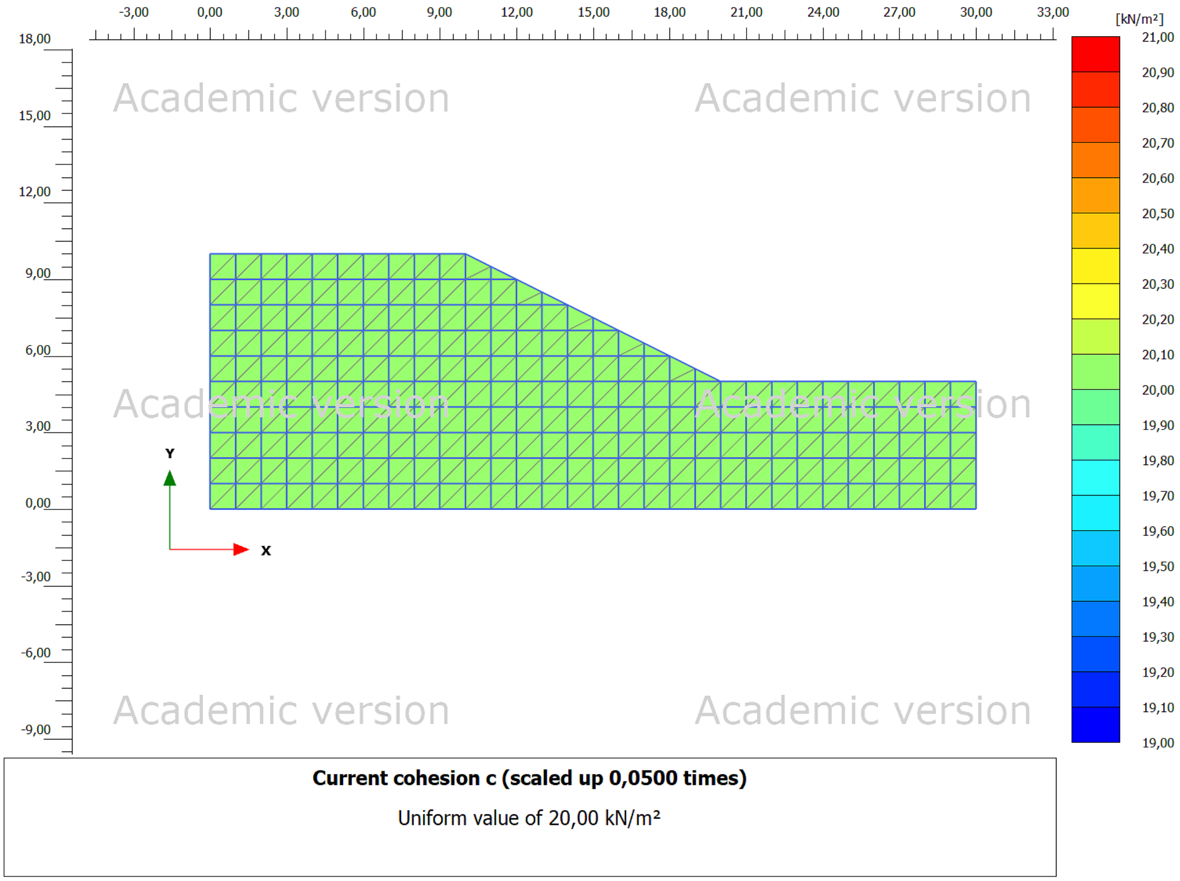


Figure 4.22: Slope geometry, with soil strength property from the random field mapped to the soil Plaxis soil elements and triangular FEM elements displayed

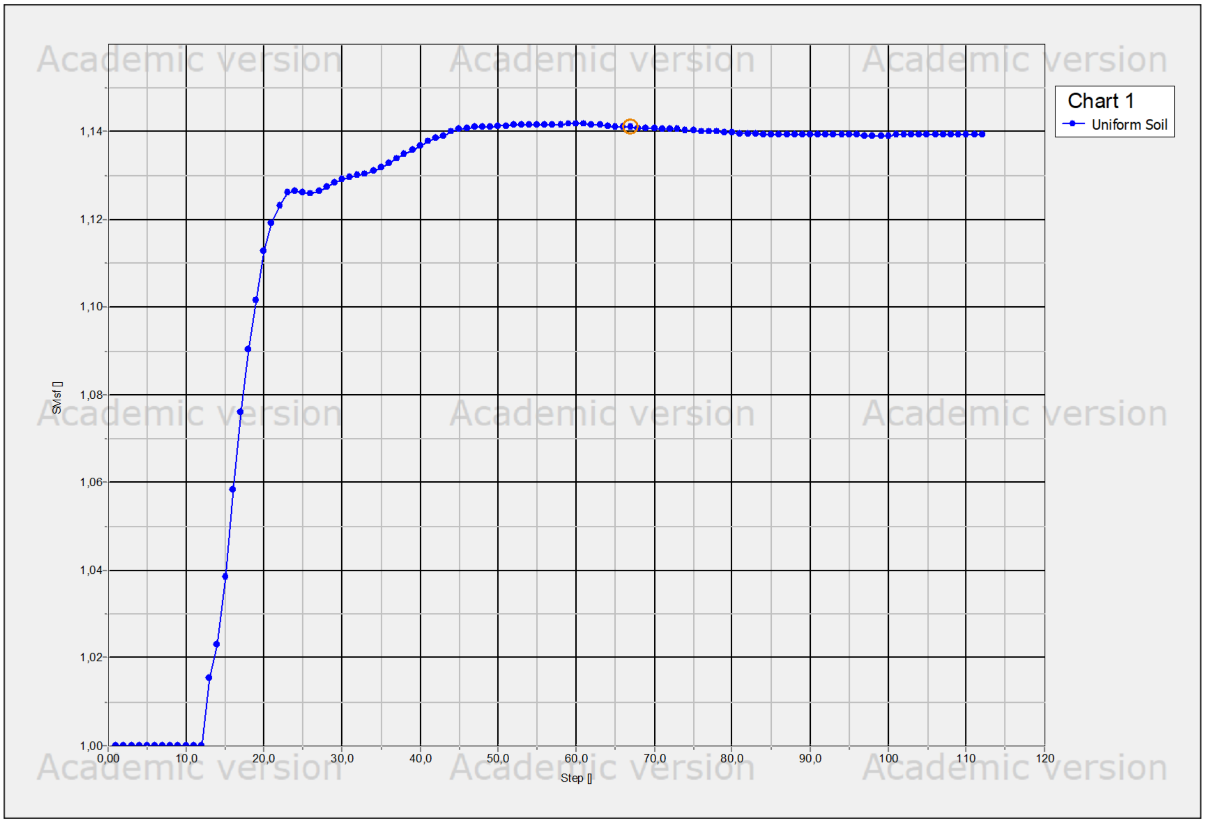
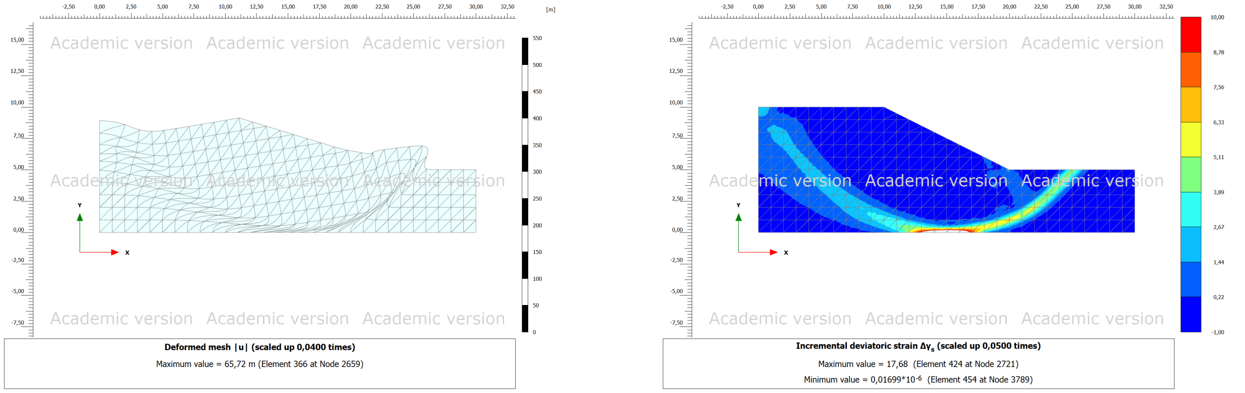


Figure 4.23: Plaxis Safety factor


Academic version
[m]
Y
X
Deformed mesh |u| (scaled up 0,0400 times)
Maximum value = 65,72 m (Element 366 at Node 2659)
Incremental deviatoric strain Δγs (scaled up 0,0500 times)
Maximum value = 17,68 (Element 424 at Node 2721)
Minimum value = 0,01699*10-6 (Element 454 at Node 3789)

(a) Deformed mesh showing the classic circular failure mechanism

(b) Shear band showing the classic circular failure mechanism

Figure 4.24: Failure mechanism for the homogeneous isotropic spatially invariant soil example

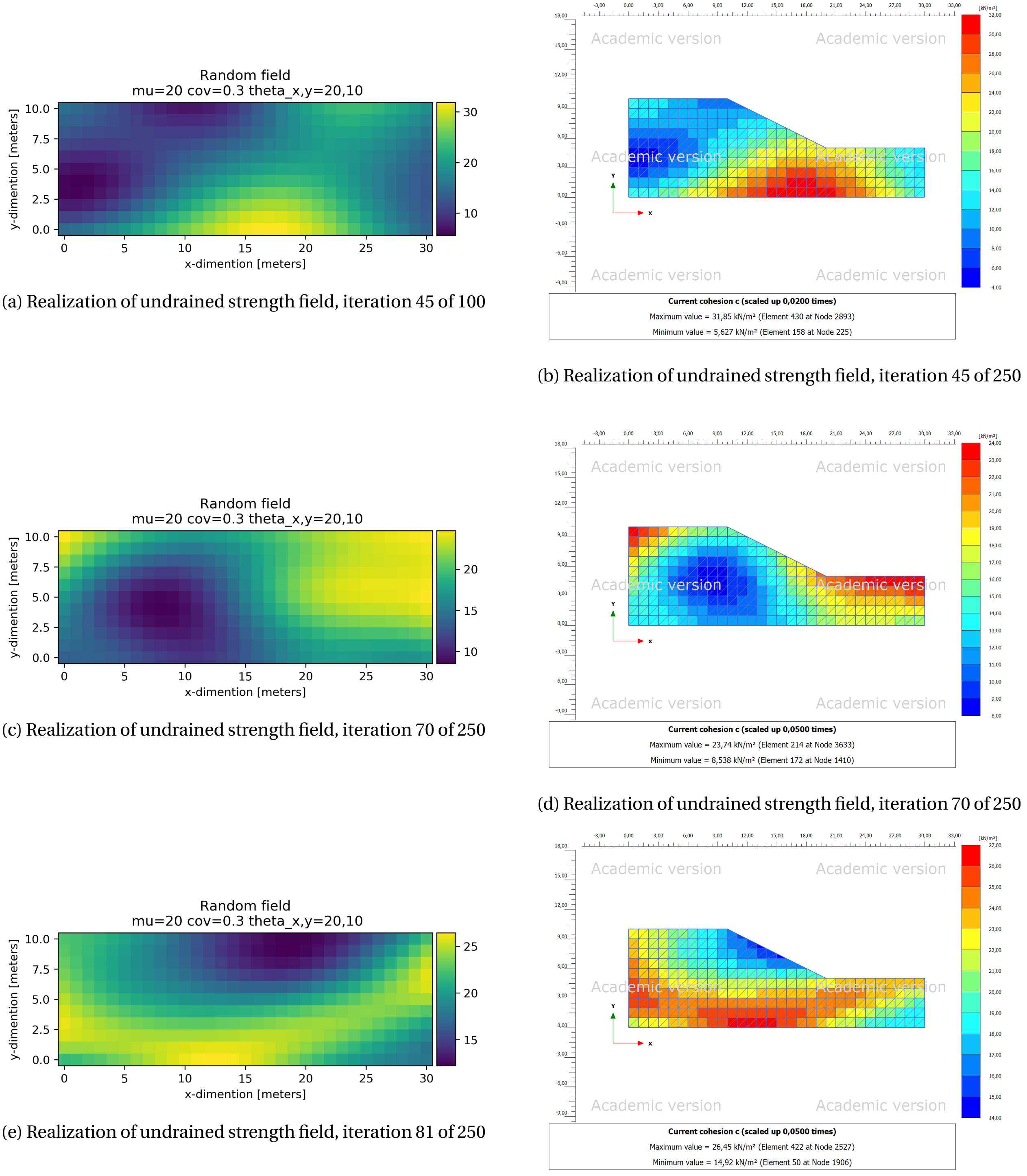
4.6 Slope stability problem - Homogeneous anisotropic soil - Scale of Fluctuation 20 and 10 meters, CoV=0.3 - Monte Carlo 250 realizations

Following are the results from 250 realizations and simulations using random field as undrained strength parameters. The simulations are elasto-plastic FEM simulation with Mohr-Coulomb material model using Plaxis undrained(C) behaviour. 15-Node triangular FEM elements are used. The slope is 5 meter high with a 2:1 gradient. The soil parameters are presented in Table 4.6. Examples of resulting random field is shown in 4.25.

Figure 4.26 shows the probability of failure of the slope vs iteration number. Taking the realizations through a $c-\phi$-reduction step to determine the factor of safety, gives the histogram in 4.27. The mean value of the factor of safety is around 1.1 with a standard deviation of 0.3. The mean value is a little less than that for the homogeneous isotropic soil of 1.14 and the analytical value of 1.16. Based on the shape of the histogram, assuming FOS is normally distributed gives, $P(FOS < 1) = 0.37$, about the same failure probability as the Monte Carlo estimate. More iterations in the simulation is needed to give a more robust result.

Table 4.6: Soil parameters for anisotropic soil

Soil model	Mohr Coulomb - Undrained(C)		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma unsat$	20 kN/m^3	0	-,-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, v	0.49	0	-
Undrained Shear Strength,S_u	20 kPa	0.3	20,10



(a) Realization of undrained strength field, iteration 45 of 100

(b) Realization of undrained strength field, iteration 45 of 250

(c) Realization of undrained strength field, iteration 70 of 250

(d) Realization of undrained strength field, iteration 70 of 250

(e) Realization of undrained strength field, iteration 81 of 250

(f) Realization of undrained strength field, iteration 81 of 250

Figure 4.25: Three different realizations of random fields created by SRM and their mapping to the FEM mesh grid.

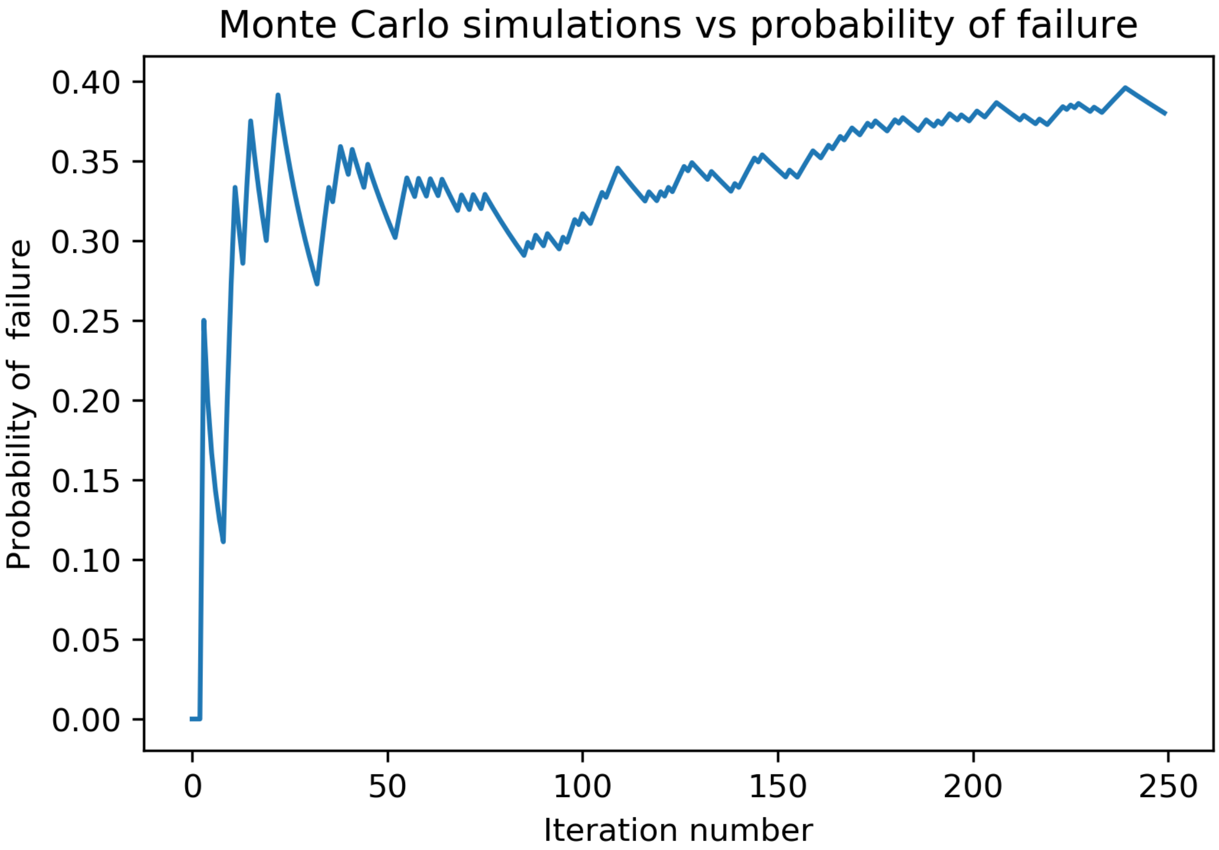


Figure 4.26: Iteration number vs probability of failure for the Monte Carlo RFEM slope stability problem. The failure probability converges towards a value of around 0.38. More iterations might be needed to see if the result is stable.

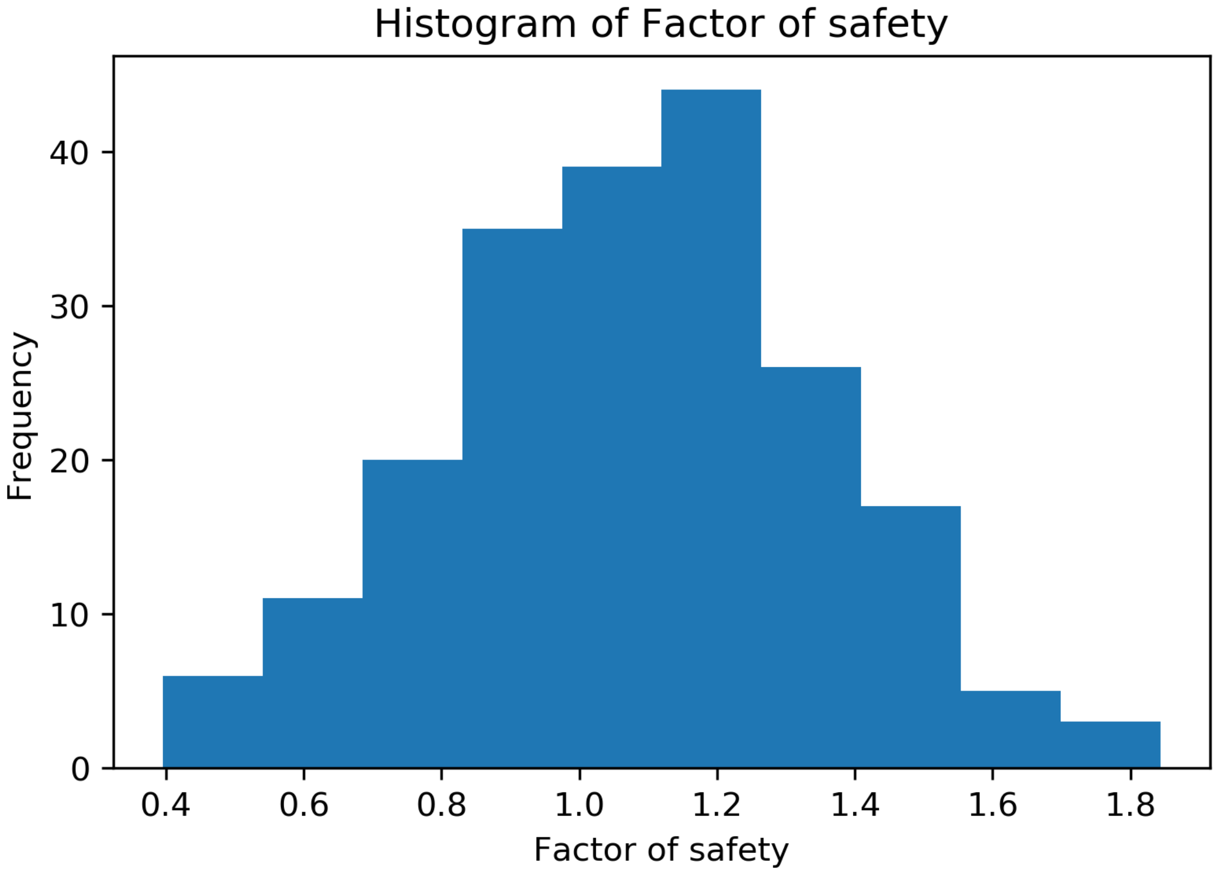
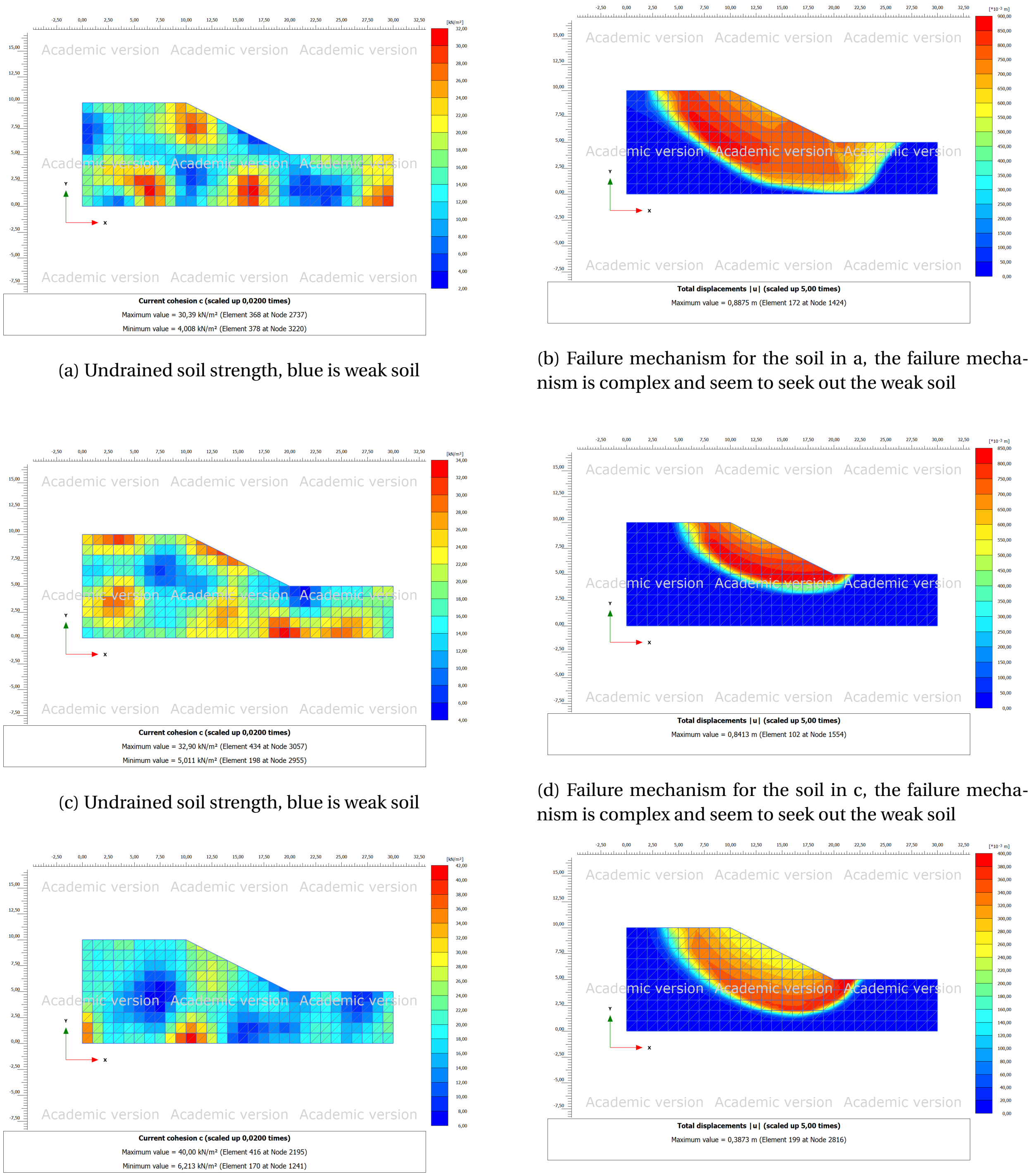


Figure 4.27: Histogram of the safety factor for the realizations in the Monte Carlo simulation

4.7 Slope stability problem - Scale of Fluctuation 5 meters, CoV=0.3 - Failure mechanism analysis

To show the effect of a complex failure mechanism in spatially varying soil a series of realization using isotropic SOF of 5 meters and a CoV of 0.3 and mean value of 20 was run. The results are presented in figure 4.28. Compared to figure 4.24 we see that for the spatially varying soil, the failure mechanism is not a given shape, and that it seems to seek out the weak soil.



(a) Undrained soil strength, blue is weak soil

(b) Failure mechanism for the soil in a, the failure mechanism is complex and seem to seek out the weak soil

(c) Undrained soil strength, blue is weak soil

(d) Failure mechanism for the soil in c, the failure mechanism is complex and seem to seek out the weak soil

(e) Undrained soil strength, blue is weak soil

(f) Failure mechanism for the soil in e, the failure mechanism is complex and seem to seek out the weak soil

Figure 4.28: Different realizations of random fields with isotropic SOF of 5 meters and a CoV of 0.3 and mean value of 20kPa

4.8 Bearing Capacity Problem - Homogeneous isotropic soil - CoV=0.0 - Known analytical solution

To illustrate the versatility of the implementation, the slope stability problem is changed to a bearing capacity problem by specifying a flat topology and adding a line load. This change is done with the change of 2 lines of code. To verify the validity, again a base case with an analytical solution is set up. Undrained bearing capacity for a smooth foundation with the known bearing capacity:

$$q = 5.14S_u \tag{4.1}$$

The simulation is a elasto-plastic FEM simulation with Mohr-Coulomb material model using Plaxis undrained(C) behaviour. 15-Node triangular FEM elements are used. A line load of 1000kPa and length of 9 meters is applied to the top of the soil. The soil parameters are presented in Table 4.7. Internal in the Plaxis code, ref step 6 in chapter 3.3, the 1000 kPa load is applied in increments, the fraction of the load applied is given in in the Plaxis variable called $\sum Mstage$. see Figure 4.29 for simulation result. The bearing capacity of the soil is given as:

$$q = 1000 * \sum MStage = 1000 * 0.103 = 103 \tag{4.2}$$

Equation 4.1 gives for $S_u = 20$, $q = 5.14S_u = 5.14 * 20 = 102.8$. The result of the simulation gives a good approximation.

The soil field, in this particular case a constant homogeneous field, is shown in Figure 4.30 and bearing capacity geometry, load and Plaxis 2D mesh is shown in Figure 4.31. The failure mechanism is illustrated in 4.32

Table 4.7: Soil parameters for anisotropic soil

Soil model	Mohr Coulomb - Undrained(C)		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x and θ_y
Unit weight, $\gamma_{sat} = \gamma unsat$	20 kN/m^3	0	-,-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, v	0.49	0	-
Undrained Shear Strength,S_u	20 kPa	0	0 and 0

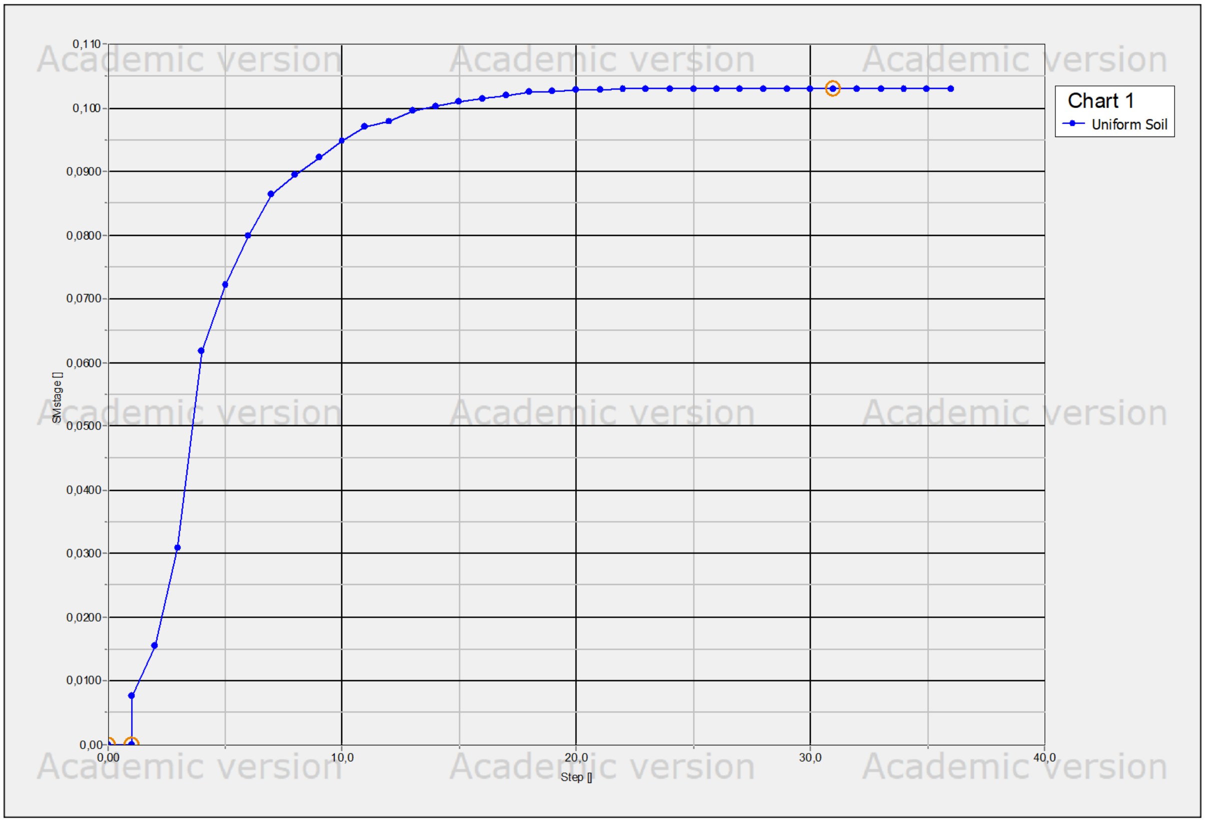


Figure 4.29: Bearing Capacity problem for uniform soil. Plaxis 2D Load step vs fraction of applied load. Failure occur at 0.103 times the applied load of 1000 kPa

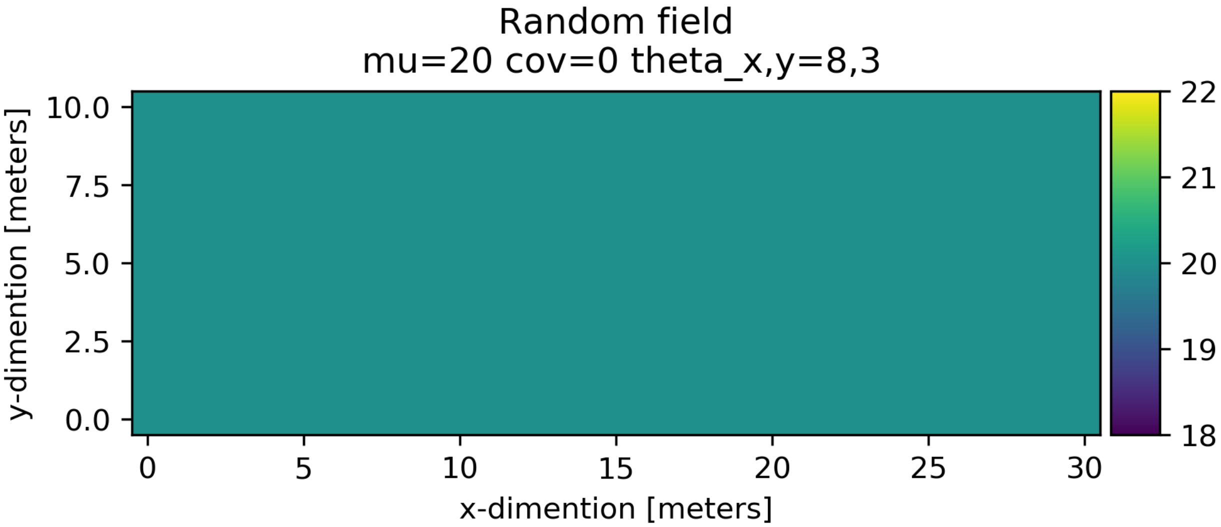


Figure 4.30: Uniform undrained soil shear strength of 20 kPa used in bearing capacity verification simulation

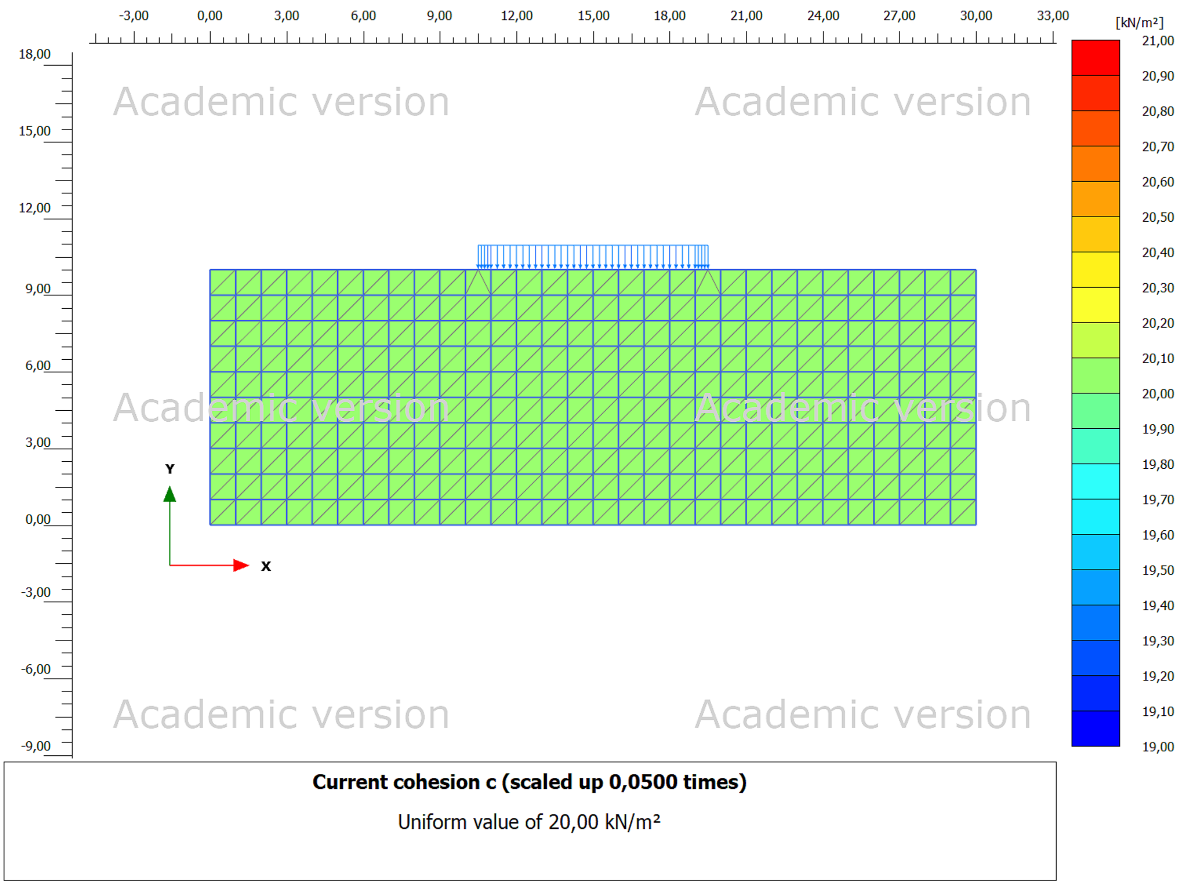


Figure 4.31: Bearing capacity problem geometry, load and FEM mesh.

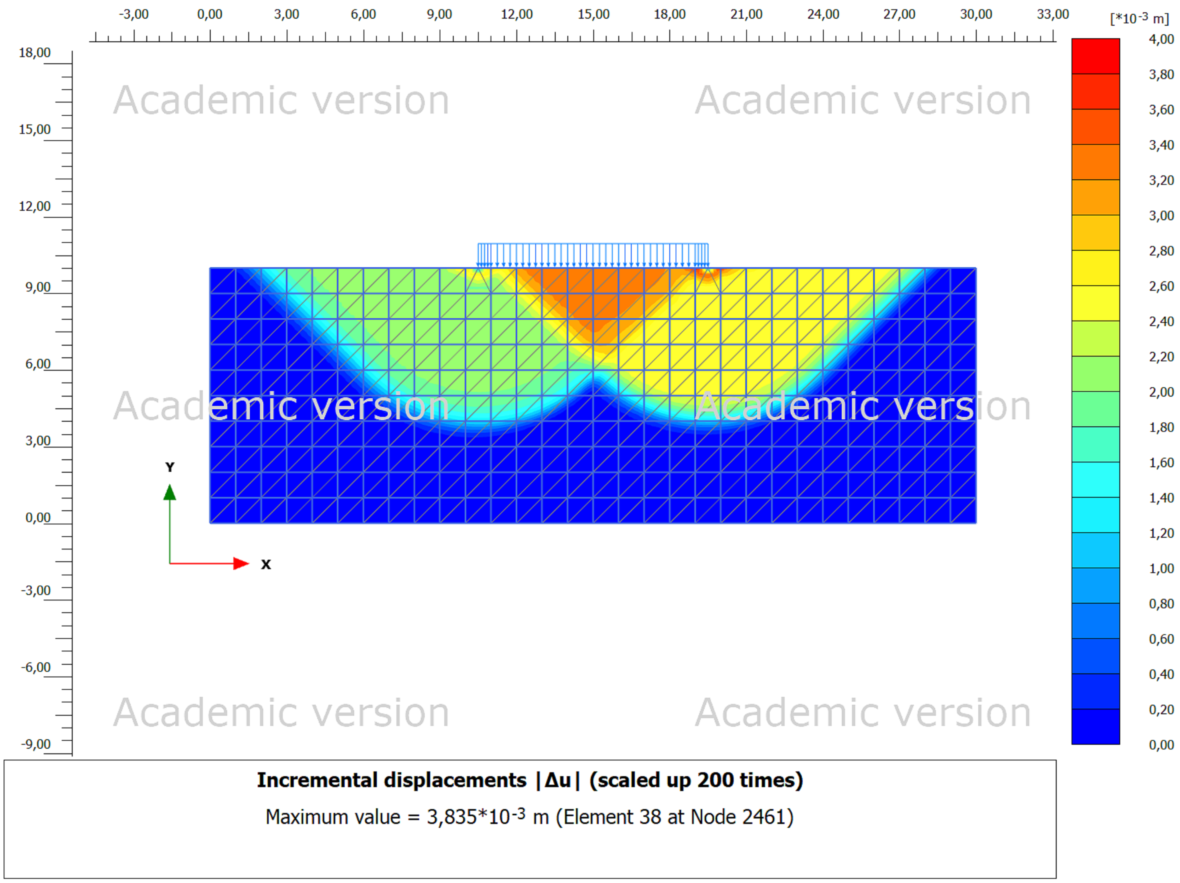


Figure 4.32: Illustration of failure mechanism for the bearing capacity verification simulation. Due to numerical approximations the failure seem to go to one side while the theoretical is symmetrical.

4.9 Bearing Capacity Problem - Homogeneous anisotropic soil - Scale of Fluctuation 8 and 3 meters, CoV=0.1, Monte Carlo 250 realizations

The bearing capacity problem is repeated with a random field input representing the soil undrained strength. The soil parameters are given in table 4.8. 250 iterations are run in a Monte Carlo simulation. The FEM parametrisation is the same as for the other problems, repeated here: Elasto-plastic FEM simulation with Mohr-Coulomb material model using Plaxis undrained(C) behaviour. 15-Node triangular FEM elements are used.

Table 4.8: Soil parameters for anisotropic soil

Soil model	Mohr Coulomb - Undrained(C)		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma unsat$	20 kN/m^3	0	-,-
Modulus of elasticity, E	10 MPa	0	-
Poissons ratio, v	0.49	0	-
Undrained Shear Strength,S_u	20 kPa	0.3	8,3

The resulting bearing capacities from the 250 realizations of the random finite element run is displayed in a histogram in Figure 4.33. Note that all bearing capacities is lower than that for uniform soil with constant undrained shear strength equal to the mean S_u in the RFEM run. The mean bearing capacity of the 250 iterations is around 95 kPa compared to the theoretical value of about 103 kPa.

Figure 4.34 shows three random realizations from the RFEM run and the corresponding variety in failure mechanism.

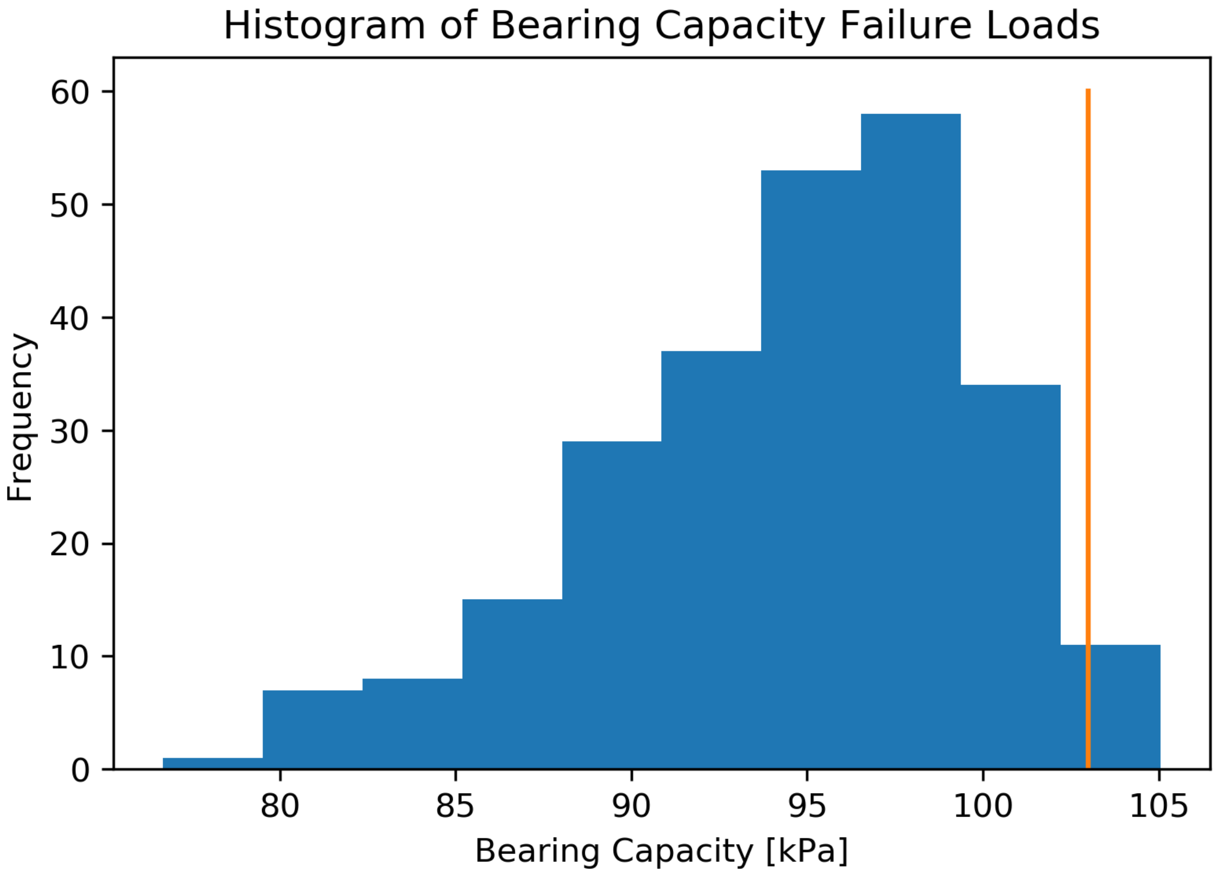
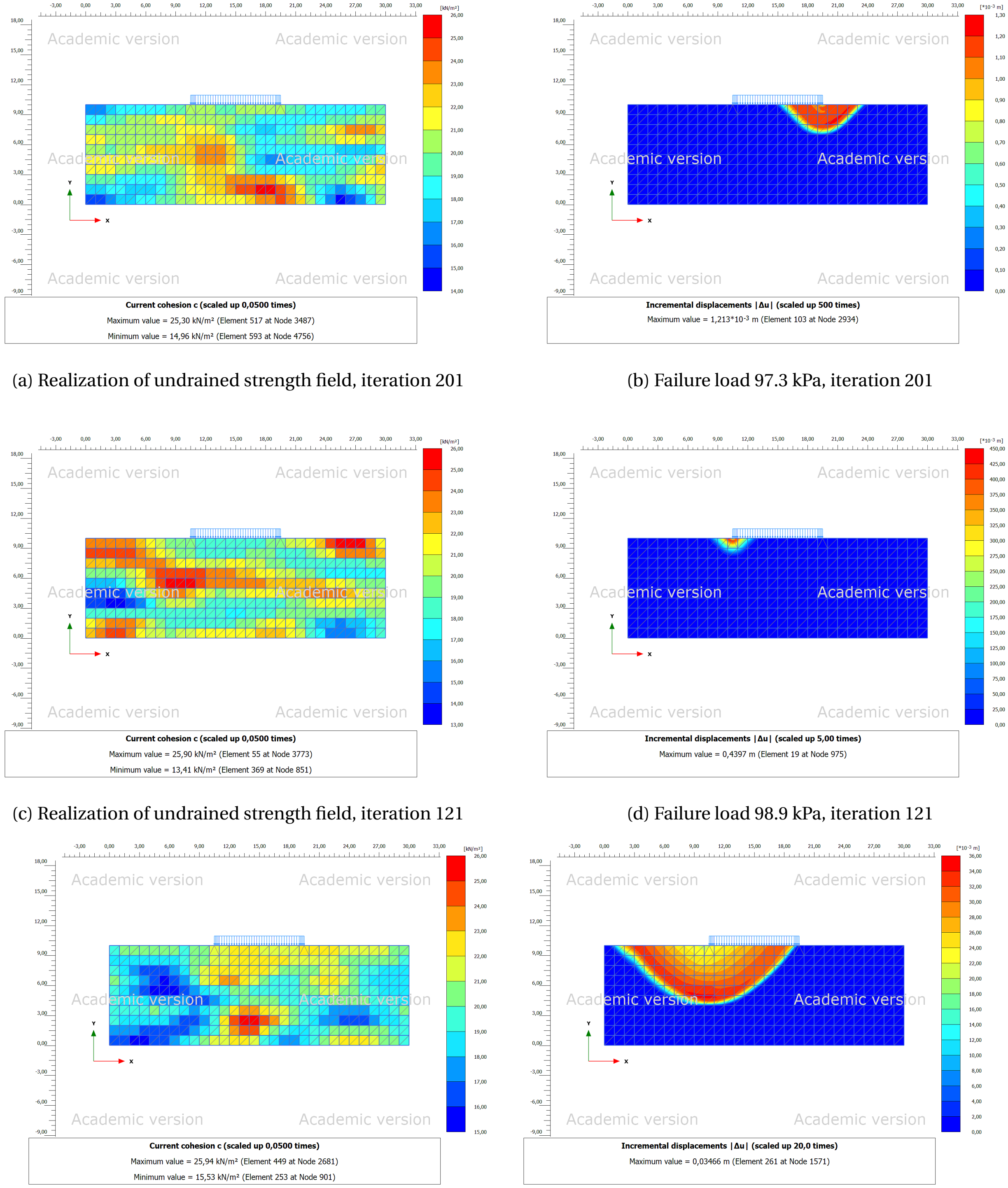


Figure 4.33: Histogram of Bering Capacities from 250 iterations of RFEM. The vertical line shows the theoretical value of 103 kPa



(a) Realization of undrained strength field, iteration 201

(b) Failure load 97.3 kPa, iteration 201

(c) Realization of undrained strength field, iteration 121

(d) Failure load 98.9 kPa, iteration 121

(e) Realization of undrained strength field, iteration 33

(f) Failure load 97.6 kPa, iteration 33

Figure 4.34: Three different realizations of random fields (Left) created by SRM, their FEM mesh and their corresponding failure mechanism (right) after incremental loading.

4.10 Oedometer test spatially varying soil - Modified Cam Clay soil model - fully coupled flow

To demonstrate more advanced features of the plaxis/RFEM script interface a constant rate of strain (CRS) oedometer test was simulated with a modified cam clay soil model and a fully coupled flow - deformation calculation. A sample of 6 centimeters in height and 12 centimeters in width was deformed by a line displacement at a constant rate of 1.0 %/hour for a total test duration of 0.416 days (10 hours) leading to a total deformation of 10 % (0.6 cm). The soil grid is 0.5 by 0.5 centimeters with on average two triangular FEM elements per soil cell. The sample is one way drained at the top of the sample. The pore pressure is recorded at the middle of the sample base. The displacement and the resulting force on the sample was recorded for each time step.

Studies have shown water content variations in samples after storage impacts the sample quality [Amundsen and Thakur, 2018]. Water content can be linked with void ratio. 300 simulations was run for two different values for CoV for the initial void ratio, that maybe could represent a disturbed or undisturbed sample. Figure 4.35 shows the void ratio at the end of the test for one realization. Figure 4.36 shows the ground water flow for the realization in 4.35, the ground water flow is predominantly upwards. The groundwater flow seem to be higher at the left side of the sample where the void ratio is lower. Figure 4.37 shows the pore pressure at the base of the sample at the last time step for one realization. The cross section of the pore pressure distribution resemble a parabola, fitting with the theoretical assumptions. There are small lateral variations in pore pressure.

Equation 4.3 was used for changing permeability. The initial permeability was constant and uniform:

$$log(\frac{k}{k_0}) = \frac{\Delta e}{c_k} \tag{4.3}$$

where k is the permeability, e is the void ratio and c_k is a constant chosen to be equal to 0.25.

Figure 4.38 and figure 4.39 shows the result from 300 realizations of the random field with parameters from table 4.9. A larger CoV gives a larger range in most interpreted parameters from the oedometer tests.

Table 4.9: Soil parameters for spatially varying soil in oedometer MCCM test

Soil model	Modified Cam Clay Model		
Statistical Soil Parameters	Mean μ	Coefficient of Variation $CoV = \frac{\sigma}{\mu}$	Scale of fluctuation θ_x, θ_y
Unit weight, $\gamma_{sat} = \gamma_{unsat}$	20 kN/m^3	0	-
λ	0.1	0	-
κ	0.02	0	-
permeability,$k_x = k_y$	0.1	-	
strength, M	1	0	-
preoverburden preasure, POP	100	0	-
void ratio, e	0.95	0.02 and 0.05	5,5 cm

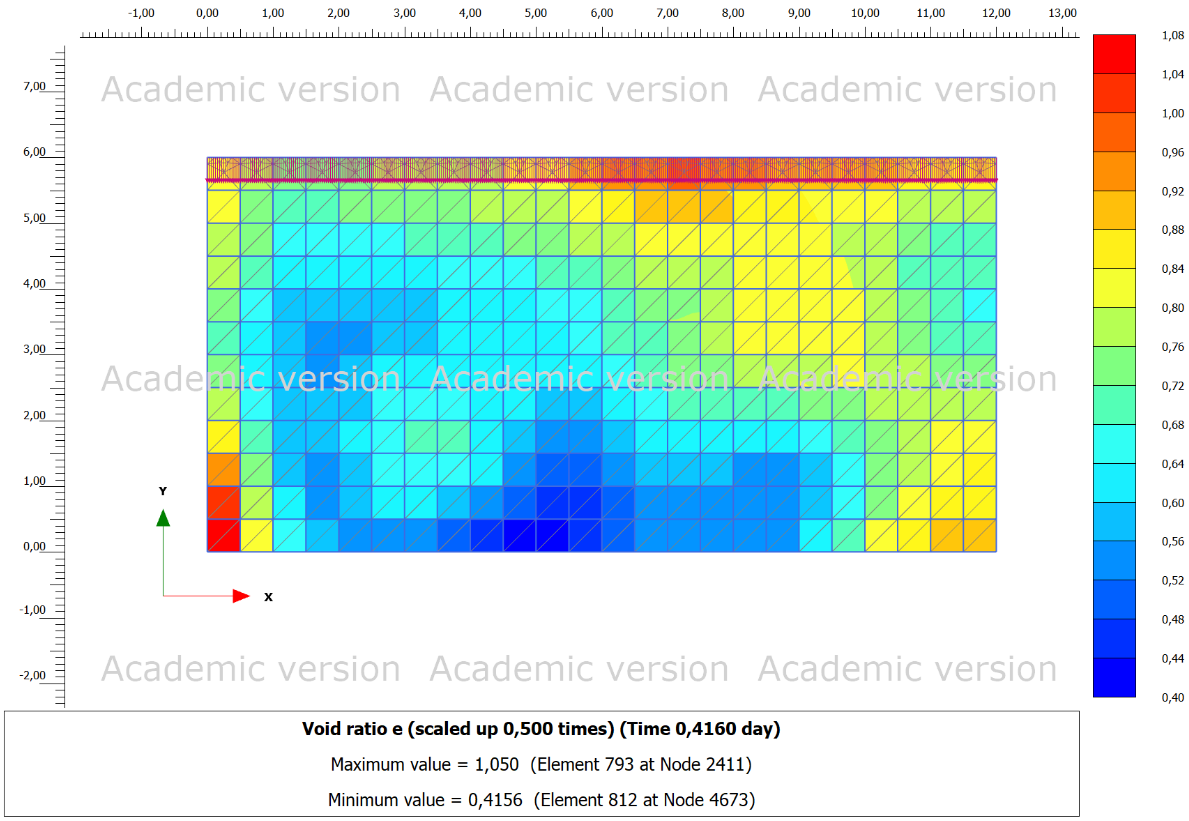
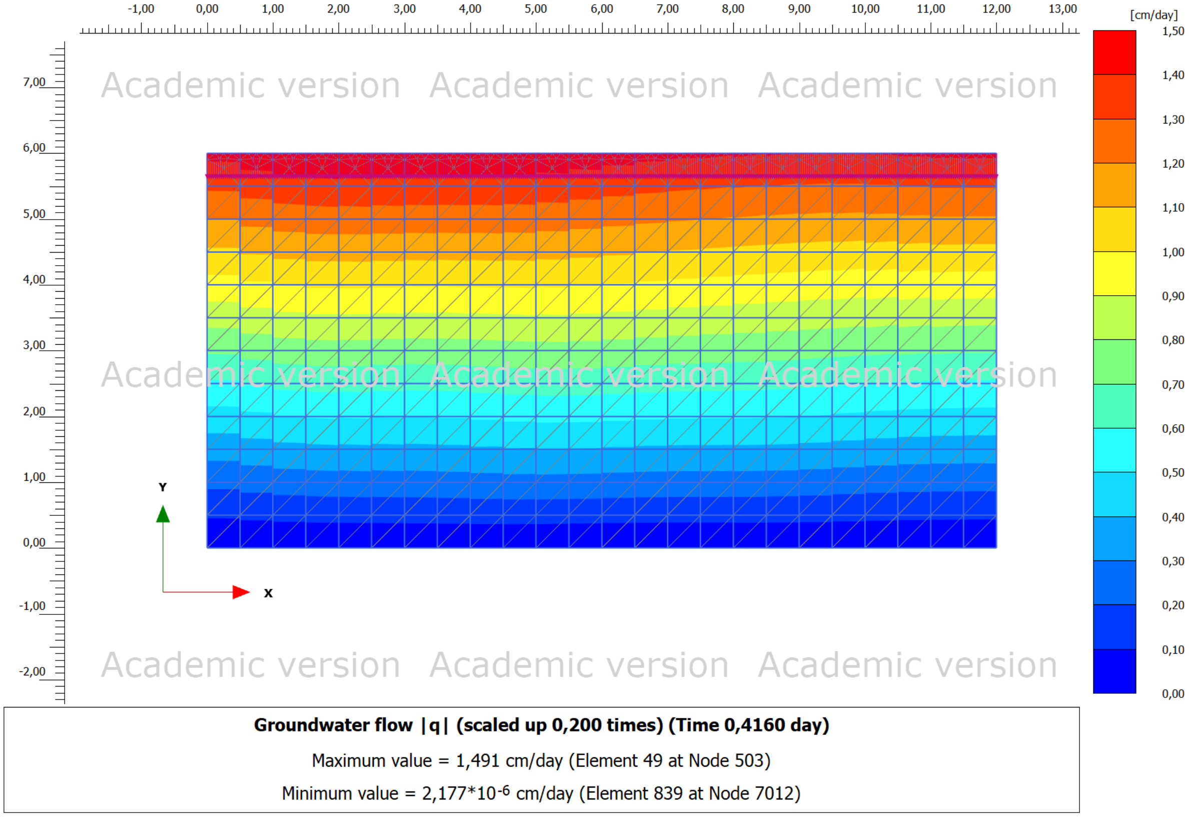


Figure 4.35: Oedometer test Void ratio at time = 10 hours

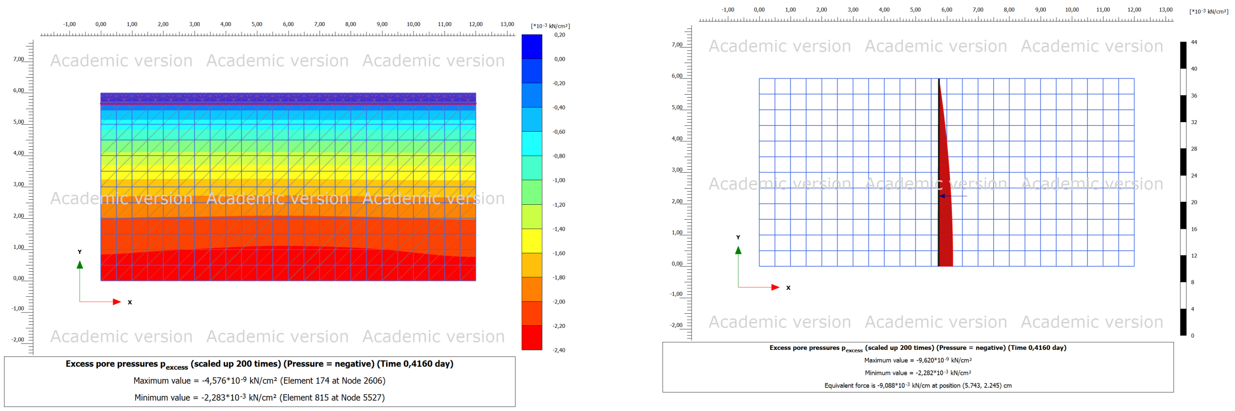


Groundwater flow |q| (scaled up 0,200 times) (Time 0,4160 day)

Maximum value = 1,491 cm/day (Element 49 at Node 503)

Minimum value = 2,177*10^{-6} cm/day (Element 839 at Node 7012)

Figure 4.36: Oedometer test ground water flow at time = 10 hours, the flow is much greater vertically than horizontally



Excess pore pressures p_{excess} (scaled up 200 times) (Pressure = negative) (Time 0,4160 day)

Maximum value = -4,576*10^{-9} kN/cm² (Element 174 at Node 2606)

Minimum value = -2,283*10^{-3} kN/cm² (Element 815 at Node 5527)

(a) Pore pressure distribution, the lateral variation is small

Excess pore pressures p_{excess} (scaled up 200 times) (Pressure = negative) (Time 0,4160 day)

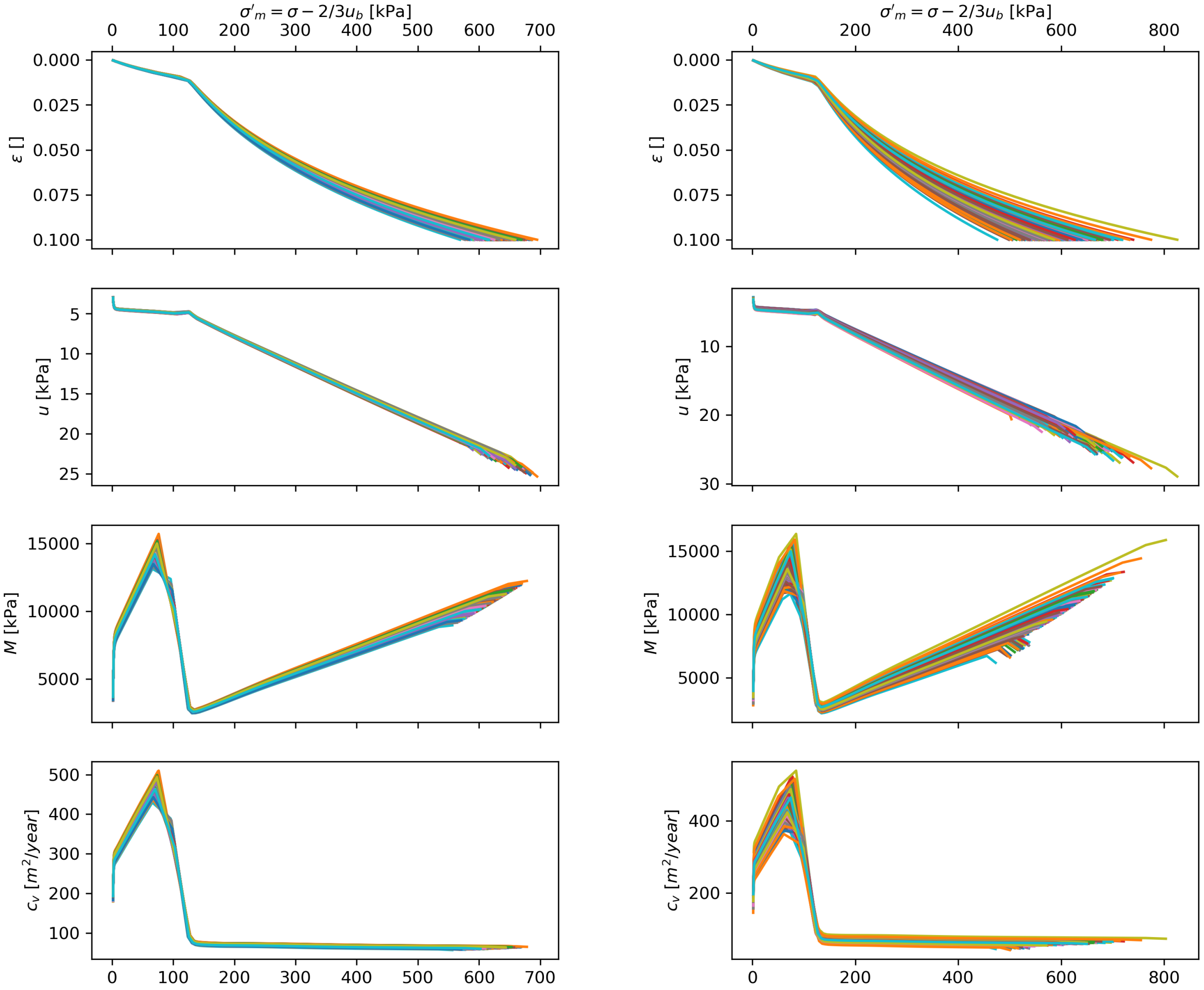
Maximum value = -9,620*10^{-9} kN/cm²

Minimum value = -2,282*10^{-3} kN/cm²

Equivalent force is -9,088*10^{-3} kN/cm at position (5.743, 2.245) cm

(b) Cross section of the pore pressure through the middle of the sample

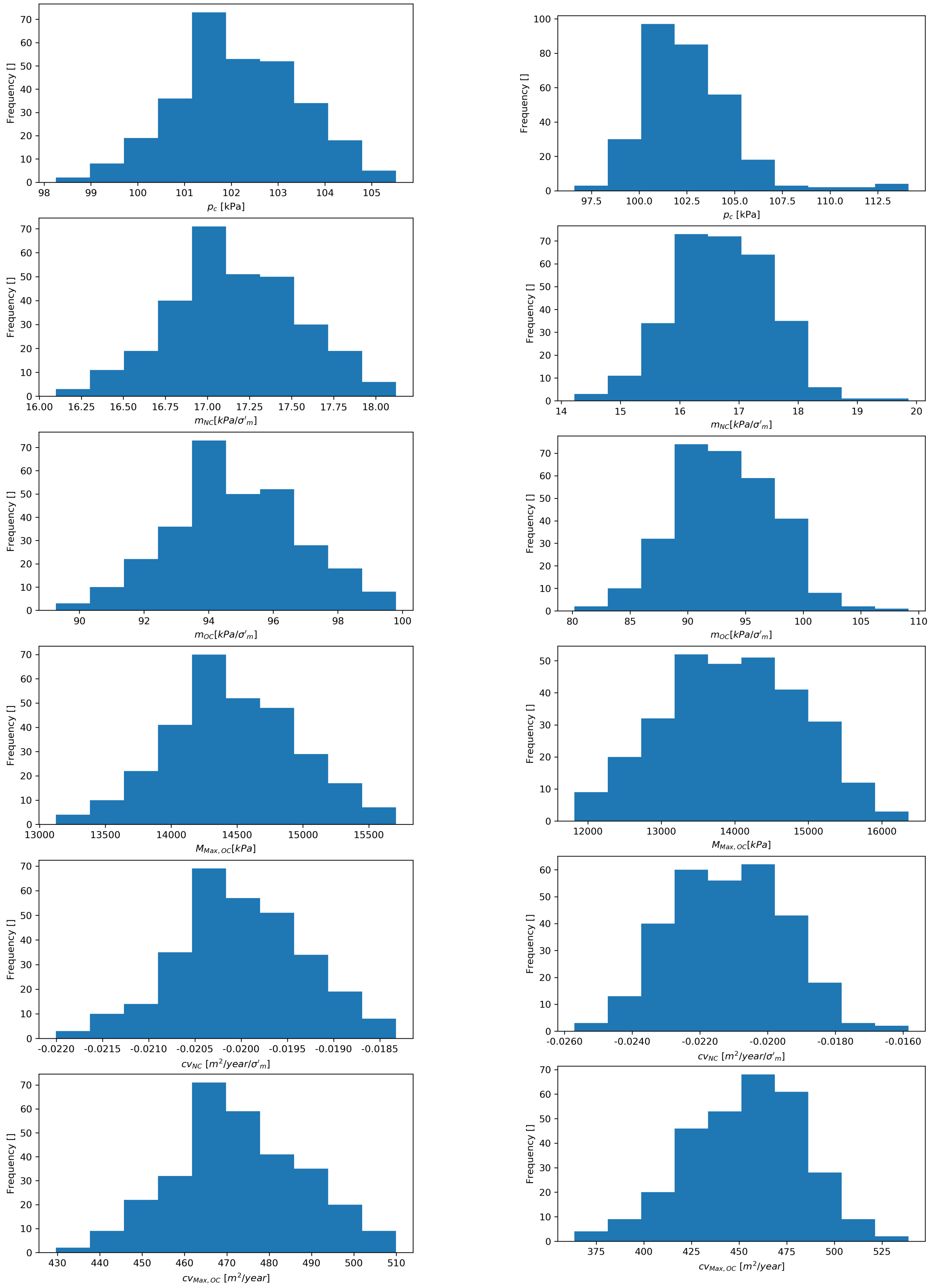
Figure 4.37: Pore pressure at time = 10 hours.



(a) CoV = 0.02

(b) CoV = 0.05

Figure 4.38: Oedometer plots for 300 realizations for two different coefficients of variations for a spatially varying void ratio. Notice different axis scales



(a) CoV = 0.02

(b) CoV = 0.05

Figure 4.39: Oedometer parameters for 300 realizations for two different coefficients of variations for a spatially varying void ratio. Notice different axis scales

Chapter 5

Summary and Recommendations for Further Work

This final chapter presents a summary of the results, a discussion of the findings and recommendations for further work.

5.1 Summary and Conclusions

In this project a method is implemented in Plaxis 2D using python API interface to run Random Finite Element simulations on geotechnical problems with advanced soilmodels and advanced analysis tools on spatially variable soils. The method is verified by controlling input parameters, fixing them to constrains giving known analytical solutions, by replicating well established laboratory experiments and successfully reproduce and expand on results from literature. The versatility of the implementation is demonstrated by verification on a slope stability and a bearing capacity problem.

5.2 Discussion

The major limitation of the implementation in its current form is its execution time per realization. The execution time for one realization of the problems described in this thesis is in the order of one minute, and the completion of a simulation with hundreds of realizations is in the order of hours. The processing speed improvement is needed for Monte Carlo simulations of rare events which demands thousands of realizations to gather the statistics needed for describing small probabilities with confidence. Rare events may also be the most interesting to study as they often could have the greatest consequences. Some speed up, like the actual Plaxis to python connection efficiency, is out of the users hands and must be handled by the Plaxis software developers, others can be done in more efficient implementation of the scripting.

The studied problems, except for the oedometer test which tested two inputs and the biaxial tests that tested two different failure criteria, did not change the input parametrization to see what effects the change in e.g the CoV or the scale of fluctuation of the random soil property field have on the results. The reason being that the main focus was on the implementation, and the implemented simulation results are adequate to prove the validity of the work done. The studies could easily be extended to study these effects, but are mostly reserved for specialized studies to isolate impacts of specific soil conditions.

There are multiple interesting observations that could be made from the results presented in this thesis. One of these studies could be the effect of soil sample disturbance on laboratory tests. The oedometer test presented could represent a disturbed and undisturbed sample, or represent uncertainty in sample acquisition. There is very little flow in the lateral direction of the oedometer test compared to the vertical direction. The vertical scale of fluctuation of permeability related parameters is the most influential in the oedometer simulation, which is also most prominent in earth samples due to sedimentation processes. But in a real loading situation the lateral flow is not restricted (like the impermeable steel ring zero lateral flow boundary condition of the oedometer test).

For the Mohr-Coloumb biaxial compression test, the slip surface cohesion and the mobilized cohesion measured in the experiments looks correlated but the results is noisy (does follow the 1:1 relation line but shifted to the left, indicating the mobilized cohesion is measured lower 4.19). The cohesion in the experiment is low given the high friction angle and effective stress level, and according to columbs law (equation 2.3) the friction angle term is dominating the shear strength. For the Tresca case the cohesion is dominating (friction angle equals zero). The mobilized strength of a sample is determined by the undrained strength (for the Tresca case), and the friction angle for the Mohr-Coloumb case (where friction angle greater than zero), of the slip surface (figures 4.8 and 4.18). The friction angle and the undrained shear strength is, as the names suggests, shear force resisting. Cohesion is a resistance against tension, the forces on the slip surface in the experiments are compressional shear, and never experience tensile forces so the cohesion is relevant only as a curve fitting parameter for the height of the Mohr-Coloumb failure line above the x-axis, impacting the ultimate failure stress state. More tests could be done to find at what ratio the different contributions dominate.

The geometry of the failure lines are mostly continuous curves with small undulations (figure 4.17), seeking weak soil but curvature and shape seems more related to mechanics and placement seems more determined by soil variability. More experiments are needed to study this under more soil conditions.

The load steps and tolerated error of the FEM calculation must be adequately low to allow for the needed precision when the sample approaches the failure surface.

The domain average cohesion (average of all sample node points) displays a narrower distribution than the slip surface average (figure 4.19). This is likely because there are more samples in the whole soil sample than in the slip surface alone. The slip surface intersects the random field and may omit peak or trough values that the domain average includes.

The estimation of the interpreted parameters from the experiments are automatically picked by an algorithm and could be cause of measurement uncertainty. Looking at the mobilized cohesion from the Mohr-Couloumb biaxial test, the range in the measured mobilized friction is greater than the range in the domain and line average (figure 4.19), suggesting that the estimate of the mobilized cohesion is prone to measurement noise. It still is able to pick up the pairwise positively correlated cohesion and friction angle input (figure 4.20c), but again the results are noisy. This also goes for the approximation to the failure surfaces, better more accurate ways exist to extract this. It is believed that the relative results presented, compared to each other for high number of experiments, is within statistical significance.

It is uncertain to what degree these implementations can be applied for use in production design in industry. The implementation, although quite simple, require a certain level of coding skill but the wide spread familiarity of engineers to FEM software and increased focus on programming and automation, and more awareness of the benefits of reliability based design and probabilistic methods suggests more solutions like the ones presented will appear in the near future.

5.3 Recommendations for Further Work

Recommendation for further work is given below. The focus of the list below is in improvement of the method implemented in this project work. Though the application of the method may be the most interesting...

The recommendations are classified by time as:

- Short-term further work
  - Stress testing by extending problem size, smaller earth elements and denser mesh. What is the limit of the program? And what size of problems can practically be analyzed?
  - Performance optimization.
- Medium-term further work
  - More advanced soil models with more random input parameters and correlated random fields. Is it possible to simulate strength softening or sensitivity?
  - More geometries and geotechnical problems like earth pressure problems. The implementation proposed can easily be extended with user input like excavation stages in a staged construction. Also possible to add and study forces on anchors, plates, sheet piles etc.
- Long-term further work
  - Performance optimization, parallelization. RFEM is a highly parallel process that can benefit by computing realizations in parallel for big performance gains and lower execution time. It is not believed to be straight forward to implement this, but maybe multiple instances of Plaxis can be started and run on different compute nodes.

Appendix A

Acronyms

CoV Coefficient of Variation

FEM Finite Element Method

FORM First Order Reliability Method

FOSM First Order Second Moment

RFEM Random Finite Element Method

SRM Spectral Representation Method

Appendix B

Source code

This appendix contains the source code to run one example of the RFEM Plaxis implementation.

B.1 Slope example

```python
import random
import numpy as np

def get_RGB_number(R, G, B):
    # get colour number from RGB using BIT LEFT SHIFT
    iB = B<<16 # left shift 16 bits for Blue
    iG = G<<8  # left shift  8 bits for Green
    iR = R     # left shift  0 bits for Red
    return iB + iG + iR

def normaliser(x,xmin,xmax):
    if xmax-xmin==0:
        return x
    return (x-xmin)/(xmax-xmin)

def create_geometry(s_i, g_i):
    """
    Takes the plaxis variables s_i, g_i and performs a series of operations
    to create a soil layer, assign a test material to soil, create a line load
    with dynamic multiplier
    """
    s_i.new()
    g_i.SoilContour.initializerectangular(0, 0, 10, 10)


```

```


#g_i.borehole(3)
#g_i.Soillayer_1.Zones[0].Bottom = -3
nx1=15
nx2=15
ny1=5
ny2=5
n=nx1+nx2

#topo=[10,10,10,10,10,10,10,10,10,10,9.5,9,8.5,8,7.5,7,6.5,6,5.5,5,5,5,5,5,5,5,5,5,5]
#interpoler topografi

H=5 #Hoeyde slope
D=2 #dybde
topoy = [D*H,D*H,D*H-H,D*H-H]
topox = [0,2*H,4*H,6*H]
#topo = np.interp(np.linspace(0,30,30),topox,topoy)
topo = np.interp(np.arange(31),topox,topoy)

print(topo)


m=ny1+ny2
matnum=0
tttrans=srm() #generate random field

sumin=np.min(tttrans)

sumax=np.max(tttrans)

for i in range(n):
    for j in range(m):
        if(j+1>=topo[i]):

            #g_i.polygon((0+i,0+j),(0+i,topo[i]),(1+i,topo[i+1]),(1+i,0+j))
            g_i.polygon((0+i,0+j),(0+i,topo[i]),(1+i,topo[i+1]),(1+i,0+j))







            su=tttrans[j,i]
            farge=int(normaliser(su,sumin,sumax)*255)
```

```
                material = g_i.soilmat("MaterialName", "Test"+str(i)+str(j), "SoilModel", "Mohr-Coulomb",
                                   "gammaUnsat", 20, "gammaSat", 20, "DrainageType","Undrained (C)",
                                    "Eref",10e3, "nu",0.49,  "cref",su,
                                      "Colour",get_RGB_number(farge, farge, farge)) #nu=poisson-ratio

                g_i.Soils[matnum].Material = material ########virker ikke saa bra med n!=m
                matnum += 1
                break


            g_i.polygon((0+i,0+j),(0+i,1+j),(1+i,1+j),(1+i,0+j))
#            material = g_i.soilmat("MaterialName", "Test"+str(i), "SoilModel", 1,
#                               "gammaUnsat", 17, "gammaSat", 20, "Gref", 2000)

            su=tttrans[j,i]
            farge=int(normaliser(su,sumin,sumax)*255)
            material = g_i.soilmat("MaterialName", "Test"+str(i)+str(j), "SoilModel", "Mohr-Coulomb",
                               "gammaUnsat", 20, "gammaSat", 20, "DrainageType","Undrained (C)",
                                "Eref",10e3, "nu",0.49,  "cref",su,
                                  "Colour",get_RGB_number(farge, farge, farge)) #nu=poisson-ratio

            g_i.Soils[matnum].Material = material ########virker ikke saa bra med n!=m
            matnum += 1

    g_i.gotostructures()
    #g_i.lineload((2.5, 0+m), (7.0, 0+m))
    #lineload_g = g_i.LineLoads[-1]
    #lineload_g.qy_start = -100


def simple_test_case(s_i, g_i):
    """
    Takes the plaxis variables s_i, g_i and performs a series of operations
    to create geometry features and generate the mesh
    """
    create_geometry(s_i, g_i)

    g_i.gotomesh()
    g_i.mesh(0.1)

    output_port = g_i.viewmesh()
    s, g = new_server(’localhost’, port=output_port, password=s_i.connection._password)

    return s, g
```

Listing B.1: Create geometry and soil model

```
from plxscripting.easy import *
import numpy as np
import matplotlib.pyplot as plt
from math import atan2


slopeStatus = np.zeros(1)

for n in range(len(slopeStatus)):

    s_o, g_o = simple_test_case(s_i, g_i)


    # Changes the mode and define phases
    g_i.gotostages()
    phase0_s = g_i.InitialPhase
    phase1_s = g_i.phase(phase0_s)
    phase1_s.Identification = ’phase1_iteration’ + str(n)

    # Activate all geometry
    g_i.Geometry.activate(phase1_s)

    g_i.calculate()
    g_i.view(phase1_s)
    print(phase1_s.Reached.SumMstage)
    slopeStatus[n]=float(str(phase1_s.Reached.SumMstage))
    #f=phase1_s.Reached.SumMstage
    #slopeStatus = np.append(slopeStatus,f)
    newest_plot = g_o.Plots[-1]
    newest_plot.ResultType = g_o.ResultTypes.Soil.C
    newest_plot.PlotType = ’shadings’
    newest_plot.Phase = g_o.Phases[-1]
    newest_plot.DrawFrame = False
    #image_wrapper is an object that can save the created
    #image or, if Pillow is installed, you can get the internal
    #Pillow.Image object and use that.
    image_wrapper = newest_plot.export(1600, 1200)
    try:
        from PIL import ImageFilter
        pil_image = image_wrapper.image
        new_image = pil_image.filter(ImageFilter.DETAIL)
        new_image.save("C:\\Users\\olewei\\Desktop\\testp"+str(time.strftime("%Y%m%d-%H%M%S"))+".png")
    except ImportError:
        #Just save if we don’t have Pillow
        image_wrapper.save("C:\\Users\\olewei\\Desktop\\testp"+str(time.strftime("%Y%m%d-%H%M%S"))+".png")
print(slopeStatus)
```

Listing B.2: Loop calculation

```
for it in range(145):
    phase2_s = g_i.phase(phase0_s)
    phase2_s.Identification = ’phase1_iteration’ + str(n+len(slopeStatus))



    nx1=15
    nx2=15
    ny1=5
    ny2=5
    n=nx1+nx2

    #topo=[10,10,10,10,10,10,10,10,10,10,9.5,9,8.5,8,7.5,7,6.5,6,5.5,5,5,5,5,5,5,5,5,5,5]
    #interpoler topografi

    H=5 #Hoeyde slope
    D=2 #dybde
    topoy = [D*H,D*H,D*H-H,D*H-H]
    topox = [0,2*H,4*H,6*H]
    #topo = np.interp(np.linspace(0,30,30),topox,topoy)
    topo = np.interp(np.arange(31),topox,topoy)

    print(topo)

    m=ny1+ny2
    matnum=0
    tttrans=srm() #generate random field
    sumin=np.min(tttrans)

    sumax=np.max(tttrans)
    g_i.gotosoil()

    for i in range(n):
            for j in range(m):
                if(j+1>=topo[i]):

                    #g_i.polygon((0+i,0+j),(0+i,topo[i]),(1+i,topo[i+1]),(1+i,0+j))
                    #g_i.polygon((0+i,0+j),(0+i,topo[i]),(1+i,topo[i+1]),(1+i,0+j))






                    su=tttrans[j,i]
                    #su = 100
                    farge=int(normaliser(su,sumin,sumax)*255)

                    g_i.Soils[matnum].Material.cref = su
                    #g_i.Soils[matnum].Material.Colour = get_RGB_number(farge, farge, farge)


                    matnum += 1
                    break


    #            g_i.polygon((0+i,0+j),(0+i,1+j),(1+i,1+j),(1+i,0+j))
    #            material = g_i.soilmat("MaterialName", "Test"+str(i), "SoilModel", 1,
    #                               "gammaUnsat", 17, "gammaSat", 20, "Gref", 2000)

                su=tttrans[j,i]
                #su = 100
                farge=int(normaliser(su,sumin,sumax)*255)

                g_i.Soils[matnum].Material.cref = su
                #g_i.Soils[matnum].Material.Colour = get_RGB_number(farge, farge, farge)
                matnum += 1

    g_i.gotostages()
    # Activate all geometry
    g_i.Geometry.activate(phase2_s)



    g_i.calculate()
    #g_i.view(phase2_s)

    slopeStatus = np.append(slopeStatus,float(str(phase2_s.Reached.SumMstage)))
    print(slopeStatus)
    print(it)
```

Listing B.3: Update soil parameters and loop

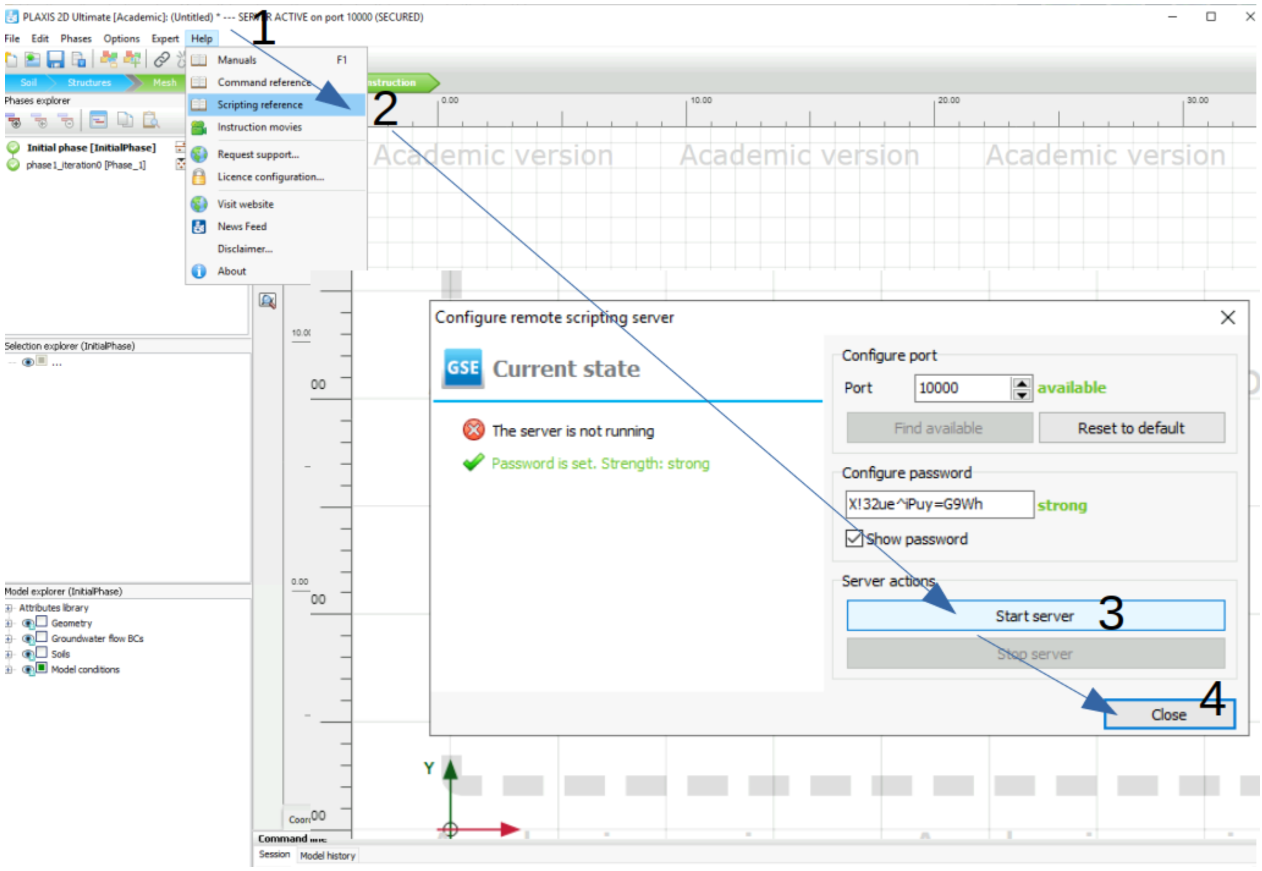
```
ppfail = np.array([])
for n,s in enumerate(slopeStatus):
    ss=slopeStatus[0:n+1]
    #print(ss)
    nonfail = np.count_nonzero(ss > .99)
    print("number of non failures ",nonfail," of ",len(ss)," iterations")
    pfail = (len(ss)-nonfail)/len(ss)
    print("probability of failure",pfail)
    ppfail = np.append(ppfail,pfail)
plt.figure()
plt.title('Monte Carlo simulations vs probability of failure')
plt.xlabel('Iteration number')
plt.ylabel('Probability of  failure')
plt.plot(ppfail)
plt.savefig("C:\\Users\\olewei\\Desktop\\pfailit"+str(time.strftime("%Y%m%d-%H%M%S"))+".png", dpi=300, bbox_inches='tight')
plt.show()
np.savetxt("C:\\Users\\olewei\\Desktop\\slopestatus.txt",slopeStatus)
```

Listing B.4: Results: graph of probability of failure

```
print(slopeStatus)
nonfail = np.count_nonzero(slopeStatus > .99)
print("number of non failures ",nonfail," of ",len(slopeStatus)," iterations")
pfail = (len(slopeStatus)-nonfail)/len(slopeStatus)
print("probability of failure",pfail)
```

Listing B.5: Results: probability of failure

B.2 Starting Python scripting interface - Jupyter notebook



Configure remote scripting server

Current state

The server is not running

Password is set. Strength: strong

Configure port

Port 10000 available

Configure password

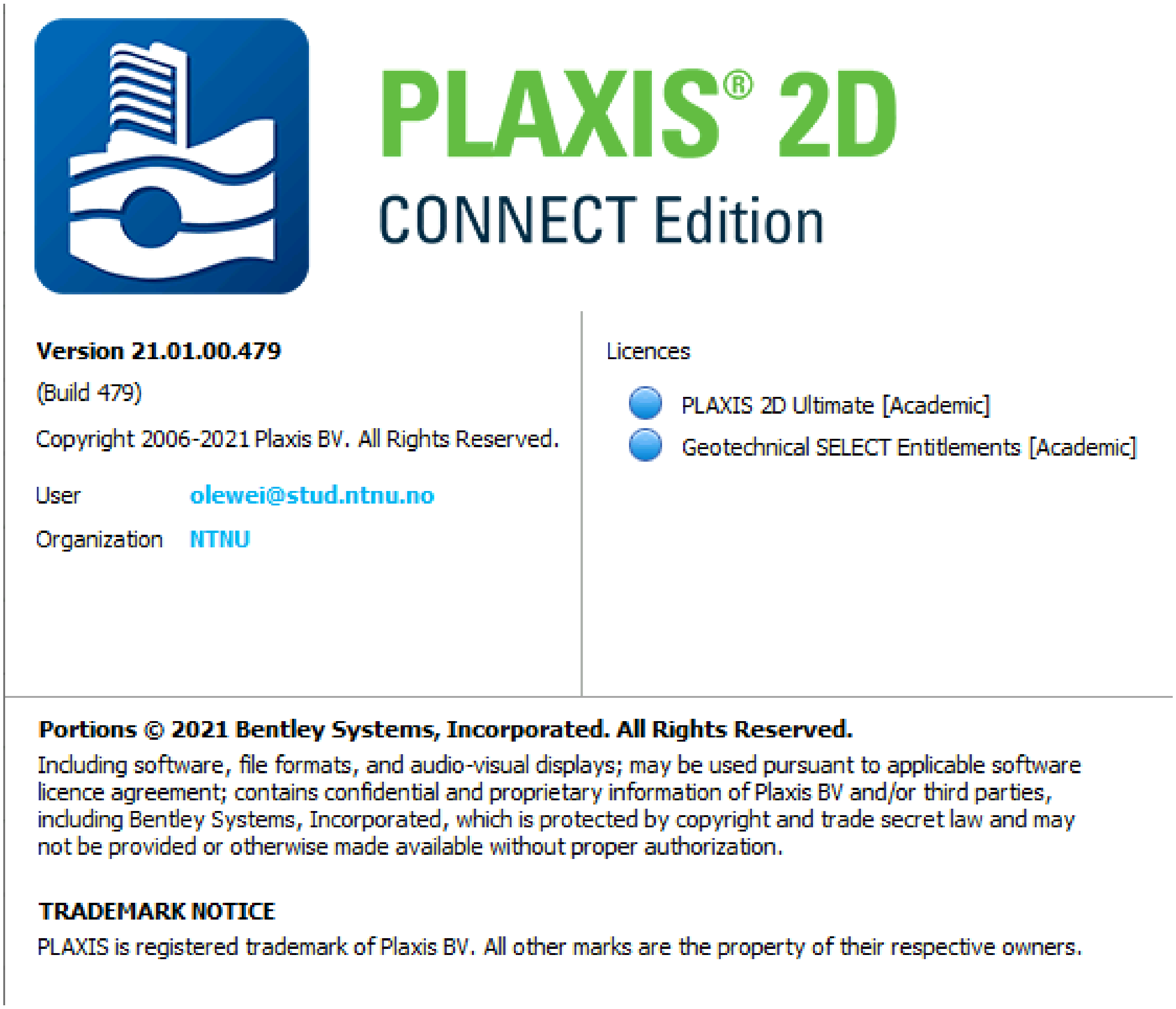
X!32ue^iPuy=G9Wh strong

☑ Show password

Server actions

Figure B.1: How to start the scripting interface in Plaxis 2D

B.3 Plaxis version information



Version 21.01.00.479

(Build 479)

Copyright 2006-2021 Plaxis BV. All Rights Reserved.

User	olewei@stud.ntnu.no
Organization	NTNU

Licences

- ● PLAXIS 2D Ultimate [Academic]
- ● Geotechnical SELECT Entitlements [Academic]

Portions © 2021 Bentley Systems, Incorporated. All Rights Reserved.

Including software, file formats, and audio-visual displays; may be used pursuant to applicable software licence agreement; contains confidential and proprietary information of Plaxis BV and/or third parties, including Bentley Systems, Incorporated, which is protected by copyright and trade secret law and may not be provided or otherwise made available without proper authorization.

TRADEMARK NOTICE

PLAXIS is registered trademark of Plaxis BV. All other marks are the property of their respective owners.

Figure B.2: Plaxis 2D version used in this project

Bibliography

H. A. Amundsen and V. K. S. Thakur. Storage duration effects on soft clay samples. 2018.

A. J. Bond, B. Schuppener, G. Scarpelli, T. L. Orr, S. Dimova, B. Nikolova, and A. V. Pinto. Eurocode 7: geotechnical design worked examples. In *Workshop "Eurocode*, volume 7, 2013.

R. Brinkgreve, W. Swolfs, E. Engin, D. Waterman, A. Chesaru, P. Bonnier, and V. Galavi. Plaxis 2d 2010. *User manual, Plaxis bv*, 2010.

J. Ching and K.-K. Phoon. Value of geotechnical site investigation in reliability-based design. *Advances in Structural Engineering*, 15(11):1935–1945, 2012.

J. Ching and K.-K. Phoon. Mobilized shear strength of spatially variable soils under simple stress states. *Structural Safety*, 41:20–28, 2013.

C. Coulomb. Test on the applications of the rules of maxima and minima to some problems of statics related to architecture. *Mem Math Phys*, 7:343–382, 1773.

G. Deodatis, M. Shinozuka, and A. Papageorgiou. Stochastic wave representation of seismic ground motion. ii: Simulation. *Journal of engineering mechanics*, 116(11):2381–2399, 1990.

G. A. Fenton, D. V. Griffiths, et al. *Risk assessment in geotechnical engineering*, volume 461. John Wiley & Sons New York, 2008.

D. Griffiths and G. A. Fenton. Seepage beneath water retaining structures founded on spatially random soil. *Geotechnique*, 43(4):577–587, 1993.

D. Griffiths and P. Lane. Slope stability analysis by finite elements. *Geotechnique*, 49(3):387–403, 1999.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett,

A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

N. Janbu. Slope stability computation. *Soil Mechanics and Foundation Engineering Report*, 1968.

K. Karlsrud and F. G. Hernandez-Martinez. Strength and deformation properties of norwegian clays from laboratory tests on high-quality block samples. *Canadian Geotechnical Journal*, 50 (12):1273–1293, 2013.

K. Krabbenhøft, A. Lymain, and J. Krabbenhøft. Optum g2 2016-user manual, 2016.

S. Nordal. Tba4116 geotechnical engineering advanced course. 2020.

M. Shinozuka and G. Deodatis. Simulation of multi-dimensional gaussian stochastic fields by spectral representation. 1996.

I. M. Smith, D. V. Griffiths, and L. Margetts. *Programming the finite element method*. John Wiley & Sons, 2013.

H. Tresca. On the yield of solids at high pressures. *Comptes Rendus Academie des Sciences*, 59: 754, 1864.

E. Vanmarcke. *Random fields: analysis and synthesis*. World scientific, 2010.

E. Vanmarcke and M. Grigoriu. Stochastic finite element analysis of simple beams. *Journal of engineering mechanics*, 109(5):1203–1214, 1983.

D. M. Wood. *Soil behaviour and critical state soil mechanics*. Cambridge university press, 1990.

NTNU

Norwegian University of
Science and Technology