Technische Universiteit Delft

Department of Geosciences and Engineering

An Automated System to Determine Constitutive Model Parameters from In Situ Tests

Author:
I.E. van Berkom

Thesis committee:

Prof. dr. K.G. Gavin	Geo-Engineering	TU Delft
Dr. ir. R.B.J. Brinkgreve	Geo-Engineering	TU Delft
Ir. H.J. Lengkeek	Hydraulic Engineering	TU Delft/Witteveen+Bos
Ir. A.K. de Jong	Geotechnical Engineering	Witteveen+Bos

A thesis submitted for the degree of Master of Science
at Delft University of Technology,
to be defended publicly on Tuesday March 24, 2020 at 16:00.

Abstract

In situ tests, such as the cone penetration test (CPT), are a popular tool for geotechnical engineers because they are an efficient and economical method for routine site characterisation, soil profiling and estimation of constitutive properties of soil. The main problem with interpreting in situ test results is the large amount of empiricism that engineers have to rely on. Furthermore, the rise in complexity of constitutive soil models have made soil interpretation from experimental data ever more challenging. Constitutive models in numerical analyses are used to simulate the stress-strain response of soils and each model is governed by a set of parameters that quantifies the mechanical behaviour of soil. A consequence for the increased complexity of a constitutive model is the increased number of parameters to be defined from a larger number of experimental tests. This is why parameter determination is still one of the most challenging tasks faced by geotechnical engineers.

The challenges in soil interpretation from experimental data have led to an increasing demand for a more efficient parameter determination system for performing more reliable numerical simulations in geotechnical engineering. This study explores the applicability of graphs to develop a generic system for the determination of constitutive model parameters from in situ test results. Graphs are mathematical structures used to model pairwise relations between objects in a network and benefit from their ability to visualise complex problems. In the same manner, the use of graphs in a parameter determination system could generate valid relations, or paths, between parameters in a network. This should give the user of the system, i.e., the geotechnical engineer, both insight in and control over the system. The proposed strategy aims to increase the confidence of derived parameters from in situ test results. It should eventually provide guidance for the user in selecting the right constitutive model and corresponding parameters for the considered application, in order to use the full potential of numerical analysis.

This study presents a proof of concept for an automated system to determine constitutive model parameters from in situ test results. Key aspects of the system are: *transparency* and *adaptability*. The system is kept transparent, since users are able to verify how available information (i.e., expertise by the engineer) is used by the system to arrive at a solution. The system is kept adaptable, since users can add their expertise into the system without having to make modifications to the system and since developers can easily expand the system in the future. This study illustrates how a system can automatically generate paths between parameters in a network (i.e., a graph), using the external database (e.g., a spreadsheet) as input by the geotechnical engineer. The focus of this system is on determining engineering parameters based on CPT data for coarse-grained soils. However, the universality of the system allows the system to be extended to a wider range of soils and in situ tests. Separate ongoing research efforts are devoted to further validating and tweaking of the system.

Keywords: parameter determination, constitutive modelling, in situ testing, soil characterisation, cone penetration test (CPT), graph, network analysis

Acknowledgements

This thesis was written to obtain my Master of Science degree in Geo-Engineering at Delft University of Technology (TU Delft), lasting from July 2019 to March 2020. This study marks the end of a very memorable time at TU Delft, which could not have been accomplished without the support of many wonderful people. Though, I would like to highlight a few individuals who have helped me pursue my goal of doing both interesting and relevant research in geotechnical engineering.

I would like to express my gratitude to my university supervisors: Ronald Brinkgreve and Ken Gavin, from the geotechnical engineering department, and Arny Lengkeek, from the hydraulic engineering department. Ronald Brinkgreve not only introduced me to this exciting topic but he encouraged me to do a subject I am really enthusiastic about. If you do not, it will be a very tough project. I now believe I can contribute a large part of the successful completion of my graduation thesis to this sound advice. Furthermore, Ken Gavin, who always gave me useful advice which kept me on the right path during my thesis project. Finally, Arny Lengkeek, who always had time to answer my questions. I admire Arny's dedication to his profession/field and appreciate his critical-thinking which helped me immensely during my research.

From Witteveen+Bos, I would like to thank Koen de Jong for his support and guidance as my daily supervisor. After guiding me during my previous internship at Witteveen+Bos, Koen responded with great enthusiasm to mentor me again during my master thesis. This study could not have been completed successfully without Koen's expertise in both fields of geotechnical engineering and software engineering, which I one day hope to match. Koen also introduced me to many dedicated engineers at Witteveen+Bos. I would like to give special thanks to Bart van Es for the on-going feedback he provided me during our meetings and his conceptual thinking ability that helped me structure my thesis.

I would like to address my special thanks to my friends from the Geo-Engineering master's programme, for their support and especially a wonderful two and a half years.

Last but not least, I would like to express my deepest gratitude for my family and Paul, for their infinite support during my entire TU Delft studies and especially during my graduation project in the last eight months.

I.E. (Ivanka) van Berkom
11 March 2020, The Hague

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem description

The advances made in hardware and software over the past thirty years have enabled a wider application of numerical methods in geotechnical engineering, both in research and development, and in practical engineering. These advances have enabled even junior engineers to solve complex soil problems that were the subject of research not so long ago. At the same time, constitutive models have developed considerably and have been implemented robustly in finite element codes, making it easy to perform numerical analysis (Schweiger, Fabris, Ausweger, & Hauser, 2018). However, the increasing complexity of constitutive models and the associated increase in the number of parameters have contributed to a gap between theory and practice in constitutive modelling.

The problem with oversimplifying soil behaviour

Constitutive models in numerical analyses are used to simulate the stress-strain response of soils. Each constitutive model is governed by a set of constitutive model parameters that quantifies the mechanical behaviour of soil. Researchers have developed numerous constitutive models, from simple to complex relationships. The linear elastic perfectly plastic Mohr-Coulomb model is widely used by geotechnical engineers in numerical analysis. The model is known for the simplest stress-strain relationship, its suitability for many practical applications and the limited number of constitutive model parameters. However, soil behaviour is more complex than predicted by the Mohr-Coulomb model. The Mohr-Coulomb model assumes linear elasticity until failure occurs. Soils exhibit nearly linear behaviour in the range of very small strains but non-linearity becomes dominant at increasing strain levels, even before the event of failure. This stiffness behaviour cannot be described by the Mohr-Coulomb model. The use of linear-elastic models in finite element analysis can underestimate movements in the soil with additional impact on supporting structures. When using the Mohr-Coulomb model in the case of excavations and retaining walls, not distinguishing between primary loading and unloading or reloading might lead to an unrealistic uplift of the retaining wall (Brinkgreve, 2005). To overcome the shortcomings of the Mohr-Coulomb model, more advanced constitutive models can be selected depending on the considered application. However, a consequence of the increased complexity of a constitutive model is that there is an increased number of material properties to be defined from a larger number of experimental tests (Wood, 2004). This explains the tendency of engineers to use less sophisticated constitutive models in numerical analysis which require fewer input parameters, as in the Mohr-Coulomb model.

Experimental testing to obtain engineering parameters

During site investigation, natural soils have to be characterised properly since soil characterisation can have large effects on projects, regarding safety, performance and economy (Mayne, 2006a). Soil interpretation is carried out through experimental testing: (*i*) *sampling-testing*, where samples are taken from the site and transported to the laboratory for testing under controlled conditions, and (*ii*) *in situ testing*, where probes are inserted into the ground with an applied pressure and the response of the soil is recorded (Graham, 2006). Sampling-testing benefits from the ability to directly obtain soil properties from experimental curves that can be used for analysis. However, the quality of such an analysis heavily depends on the quality of the obtained samples, since disturbance is always an issue. In situ tests, such as cone penetration tests, usually produce continuous soil records that provide a stratigraphy of the soil with minimal disturbance. They have become a popular tool since they are quick, reproducible and economical unlike sampling-testing. However, the main problem with interpreting in situ test results is the large amount of empiricism that engineers have to rely on. Engineering parameters cannot be obtained directly but have to be obtained through empirical relationships.

The challenging task of parameter determination

Constitutive model parameters mainly represent the stiffness and strength properties of the soil, which are of high interest in deformation and stability analyses in geotechnical problems. In order to derive these parameters, an appropriate parameter determination method is required that ensures a good understanding of the conditions under which the parameters are estimated. Parameters can be determined based on site investigation data derived from in situ tests (e.g., cone penetration test, standard penetration test), classification tests (e.g., Atterberg plasticity limits), laboratory tests (e.g., oedometer, triaxial) or geotechnical evidence (e.g., physical relationships, rules of thumbs, tables, charts). The outcome of each parameter depends on the validity and limitations of the selected method and interpretation is based on engineering judgement. This explains why parameter determination is still one of the most challenging tasks faced by geotechnical engineers. The challenges in soil interpretation from experimental data have led to an increasing demand for a more efficient parameter determination system that uses more advanced constitutive models for performing more reliable numerical simulations. *"There is little point in doing a refined analysis if the material properties cannot be identified clearly"* (Graham, 2006).

Artificial Neural Networks to determine parameters

The past two decades have shown an increased use of artificial neural networks (ANNs) in geotechnical applications (Goh & Kulhawy, 2003; Reale, Gavin, Librić, & Jurić-Kaćunić, 2018; Shahin, Jaksa, & Maier, 2001). An ANN is an artificial intelligence technique, in which a biologically inspired computational network mimic the behaviour of the human brain and nervous system. Without any prior knowledge, ANNs attempt to capture the relationship between input and output variables in a network. The main advantage is their ability to work with incomplete information. After ANN training, even incomplete data may produce an output. The most important drawback of ANNs is the incapability of explaining the behaviour of the network. ANNs are known as "black boxes". They are unable to explain *how* information is used to arrive at a solution.

A new approach to determine parameters

This study explores the applicability of graphs to develop a generic framework for the determination of constitutive model parameters from in situ test results. Graphs are mathematical structures used to model pairwise relations between objects in a network and benefit from their ability to visualise complex problems. The framework aims to increase the confidence of derived parameters from in situ test results in which the user, i.e., the geotechnical expert, is given a comprehensive understanding of the parameter determination process and is able to adapt information used by the system during this determination process. This way, the geotechnical engineer who is the expert user of the system, remains in control. Eventually, this framework aims at providing guidance for the user in selecting the right constitutive model and corresponding parameters for the considered application, in order to use the full potential of numerical analysis.

1.2 Aim, scope and objectives

The aim of this research is to elaborate a *transparent* and *adaptable* parameter determination framework that will increase the reliability of parameters derived from in situ tests using a graph-based approach. The system should be transparent, in the sense that it should be able to explain how available information is used to arrive at a solution, and adaptable, in the sense that users can incorporate their expertise into the system without making modifications to the system. This is regarded as a first step towards an automated parameter determination framework, while ensuring transparency and adaptability for the user; the geotechnical engineer.

The scope of this thesis is limited to deriving parameters that are of specific interest for most constitutive soil models, i.e., strength and stiffness parameters. The parameters are derived for coarse-grained soils (sands) based on the cone penetration test (CPT), however, due to the universality of the proposed framework, it may be extended to a wider range of soils and in situ tests. It should be emphasised that the final result is a proof of concept, demonstrating the viability of the system. An important distinction is made between verification and validation. Verification ensures that a system is built according to specifications (i.e., the system is built right), while validation ensures that the system meets the requirements of the user (i.e., the right system is built). The system is tested on a geotechnical case with fictitious data where the results are verified, however, validation lies beyond the scope of this thesis.

The six research objectives that must be achieved to meet the aim of this thesis are:

- *Generating paths.* Since more than one empirical correlation may be valid to determine a parameter, a parameter can be determined in multiple ways. The result is a network, i.e., a graph, where multiple paths lead to the same constitutive model parameter.

  How can a system be developed to generate valid paths between geotechnical parameters in a graph?

- *Performing calculations.* After generating paths, the system should be able to perform calculations for the derived parameters. This also includes the transfer of information along a path.

  How can the system be developed to allow for calculations performed for all parameters involved in a path?

- *Treating multiple parameter outcomes.* Since multiple empirical correlations may often be valid when determining a parameter, there may be multiple parameter outcomes for the same parameter.

  How does the system cope with multiple parameter outcomes?

- *Accounting for uncertainties.* Empirical correlations are often valid within a specified range and each correlation has its own limitations, which contribute to the uncertainty of the resulting parameter outcome. The more empirical correlations involved in a path, the higher the propagated uncertainty of the final parameter.

  How are uncertainties in geotechnical parameters accounted for in the system?

- *Enabling adaptability.* How can the system ensure adaptability for both the user and the future developer of the system?

  Users of the system, i.e., the geotechnical experts, must be able to incorporate their expertise into the system without having to modify the system itself. Additionally, the system should be adaptable for developers that wish to extend the system in the future, e.g. adding functionalities.

- *Ensuring transparency.* If a number of engineers are asked to calculate a geotechnical parameter for a well-known case, they will probably come up with as many different answers. In other words, there are multiple paths that may lead to the answer and the user must be able to verify the suggested paths to derive the answer.

  How can the system ensure transparency such that the user is able to verify the entire parameter determination procedure?

Chapter 2

Fundamentals of Parameter Determination

This chapter introduces the fundamentals of parameter determination in geotechnical engineering. Section 2.1 provides a review on soil interpretation by means of experimental testing which is essential to all site investigations. Section 2.2 covers the interpretation of CPT data, providing a review on the empirical relationships to determine parameters for soil profiling and to determine the state, strength and stiffness parameters in coarse-grained soils. At the end of each paragraph, a final selection is given of the empirical relationships used to generate the graph for the geotechnical case, presented in Section 4.2.

2.1 Experimental testing for soil interpretation

In all site investigations, an appropriate characterisation of soils is essential as it can have large effects on projects, regarding safety, performance and economy (Mayne, 2006a). Soils are interpreted by means of experimental testing, which can be divided into two categories (Graham, 2006): (*i*) *sampling-testing*, where samples are taken from the site and transported to the laboratory for testing, and (*ii*) *in situ testing*, where probes are inserted into the ground with an applied pressure and the response of the soil is recorded.

2.1.1 Sampling-testing

During sampling-testing, samples are transported from the site to the laboratory and stored until tests are carried out. Samples are trimmed to specific specimen sizes and tested under well-controlled conditions. This makes it straightforward to obtain parameters since they can be derived directly from experimental curves, which can be used in analysis immediately. However, the quality of such an analysis depends on how well the parameters are obtained from the samples. There are some limitations associated with sampling-testing. The main drawbacks associated with sampling-testing are:

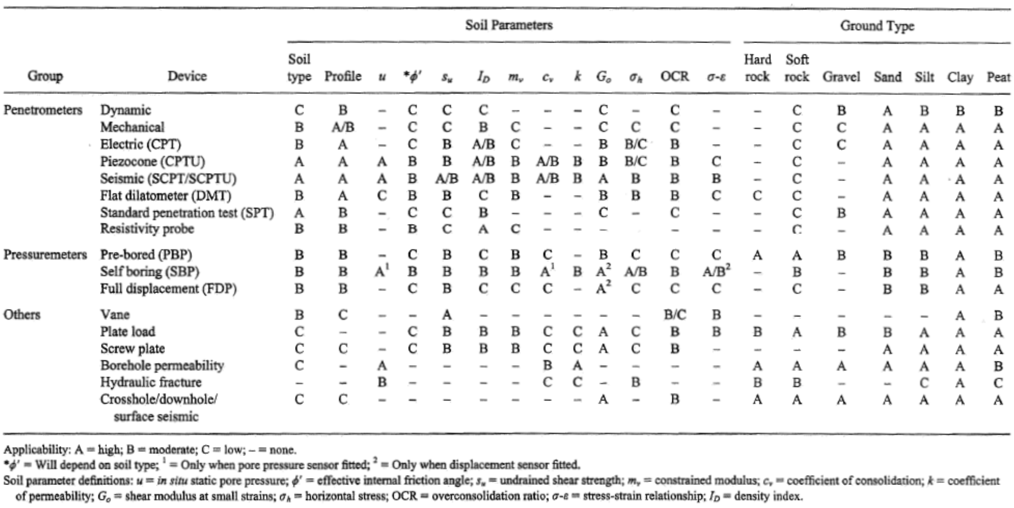
- *Sampling disturbance.* Samples may be disturbed during transportation and may not correspond with the in situ characteristics of the soil. Strength and stiffness parameters in particular, must be tested on "undisturbed samples" (Nhuan, 1981), i.e., minimal disturbance has been subjected to the soil in order for it to be used for all laboratory tests. Undisturbed sampling is especially an issue in cohesionless soils, but even in "high quality" undisturbed samples of cohesive soil, the impact of unavoidable sampling disturbance can be difficult to assess (Jamiolkowski, Ladd, Germaine, & Lancellotta, 1985).

- *Small volume of samples for testing.* Samples may not represent the soil profile as a whole (Graham, 2006); they only represent the soil at a specific location of the soil profile and may not include inhomogeneities that were initially present in situ.

- *Time-consuming and expensive.* Sampling-testing, which makes use of heavy and expensive devices, is more time-consuming and is likely to be more expensive compared with in situ testing (Mayne, 2005).

2.1.2 In situ testing

In situ testing enables testing of larger volumes of soil, hence, macrofabric effects on the soil properties are well-captured by in situ tests (Jamiolkowski et al., 1985). The soil is tested under its in situ conditions so that there is less disturbance than in sampling-testing. In situ tests, such as the cone penetration test, usually produce continuous records of the soil providing a full stratigraphy of the soil and classification of soil types. Cohesionless soils or highly fissured clays, are not able to provide undisturbed samples. Incorrect interpretations can have large impact on results of site investigation. For example, an assumption of normally-consolidated conditions of clean sands, which are very difficult to sample, will unavoidably underestimate the pile side friction or overestimate shallow footing settlements (Mayne, 2006a). These soils however are still suitable for in situ testing. In situ testing is faster than sampling-testing, making it a very economical tool in terms of cost-effectiveness. However, engineering parameters cannot be determined directly like in sampling-testing, but have to be determined empirically. Many authors have developed empirical relationships to interpret soil parameters from in situ tests, such as Been, Quiñonez, and Sancio (2010), Jamiolkowski, Lo Presti, and Manassero (2001) or Robertson (1995). Figure 2.1 illustrates the use of a number of in situ tests for different ground conditions, where the CPT, the piezocone penetration test (CPTU) and seismic penetration test (SCPT/SCPTU) show the highest applicability for most soil types. The interpretation of CPT data through empirical relationships is discussed in the next section.



		Soil Parameters													Ground Type						
Group	Device	Soil type	Profile	u	$*\phi'$	s_u	I_D	m_v	c_v	k	G_o	σ_h	OCR	σ-ε	Hard rock	Soft rock	Gravel	Sand	Silt	Clay	Peat
Penetrometers	Dynamic	C	B	–	C	C	C	–	–	–	C	–	C	–	–	C	B	A	B	B	B
	Mechanical	B	A/B	–	C	C	B	C	–	–	C	C	C	–	–	C	C	A	A	A	A
	Electric (CPT)	B	A	–	C	B	A/B	C	–	–	B	B/C	B	–	–	C	C	A	A	A	A
	Piezocone (CPTU)	A	A	A	B	B	A/B	B	A/B	B	B	B/C	B	C	–	C	–	A	A	A	A
	Seismic (SCPT/SCPTU)	A	A	A	B	A/B	A/B	B	A/B	B	A	B	B	B	–	C	–	A	A	A	A
	Flat dilatometer (DMT)	B	A	C	B	B	C	B	–	–	B	B	B	C	C	C	–	A	A	A	A
	Standard penetration test (SPT)	A	B	–	C	C	B	–	–	–	C	–	C	–	–	C	B	A	A	A	A
	Resistivity probe	B	B	–	B	C	A	C	–	–	–		–	–	–	C	–	A	A	A	A
Pressuremeters	Pre-bored (PBP)	B	B	–	C	B	C	B	C	–	B	C	C	C	A	A	B	B	B	A	B
	Self boring (SBP)	B	B	A[1]	B	B	B	B	A[1]	B	A[2]	A/B	B	A/B[2]	–	B	–	B	B	A	B
	Full displacement (FDP)	B	B	–	C	B	C	C	C	–	A[2]	C	C	C	–	C	–	B	B	A	A
Others	Vane	B	C	–	–	A	–	–	–	–	–	–	B/C	B	–	–	–	–	–	A	B
	Plate load	C	–	–	C	B	B	B	C	C	A	C	B	B	B	A	B	B	A	A	A
	Screw plate	C	C	–	C	B	B	B	C	C	A	C	B	–	–	–	–	A	A	A	A
	Borehole permeability	C	–	A	–	–	–	–	B	A	–	–	–	–	A	A	A	A	A	A	B
	Hydraulic fracture	–	–	B	–	–	–	–	C	C	–	B	–	–	B	B	–	–	C	A	C
	Crosshole/downhole/ surface seismic	C	C	–	–	–	–	–	–	–	A	–	B	–	A	A	A	A	A	A	A

Applicability: A = high; B = moderate; C = low; – = none.
*ϕ' = Will depend on soil type; [1] = Only when pore pressure sensor fitted; [2] = Only when displacement sensor fitted.
Soil parameter definitions: u = *in situ* static pore pressure; ϕ' = effective internal friction angle; s_u = undrained shear strength; m_v = constrained modulus; c_v = coefficient of consolidation; k = coefficient of permeability; G_o = shear modulus at small strains; σ_h = horizontal stress; OCR = overconsolidation ratio; σ-ε = stress-strain relationship; I_D = density index.

Figure 2.1: Applicability of in situ tests (Lunne et al., 1997).

2.2 Interpretation of CPT data

The cone penetration test (CPT) is now internationally recognised as an established, efficient and economical method for routine site characterisation, soil profiling and assessment of constitutive properties of geomaterials (Mayne, 2014; Schnaid, 2009). Although the first cone penetrometers were made by the Dutch engineer P. Barentsen in 1932, the CPT was patented in 1953 by Begemann who improved the Dutch static cone penetration test and who later proposed to classify soil from cone tip resistance and sleeve friction readings (Begemann, 1965). In the late 1970s, the CPT with pore pressure sensors, known as the piezocone penetration test (CPTU), was developed. The CPTU consists of a cone with a 60 degree apex angle, a cross-sectional area of 10 cm^2 and a friction sleeve area of 150 cm^2 (Figure 2.2). The cone is pushed into the ground at a constant rate of 2 cm/s and measurements of the cone tip resistance (q_c), sleeve resistance (f_s) and the penetration pore water pressure (u_2) are recorded by a single sounding. These three readings are obtained at depth intervals of 1 to 5 cm and are interpreted to profile the soil stratigraphy and estimate soil engineering parameters, related to stress history, shear strength and soil stiffness at the CPT location (Mayne, 2014).

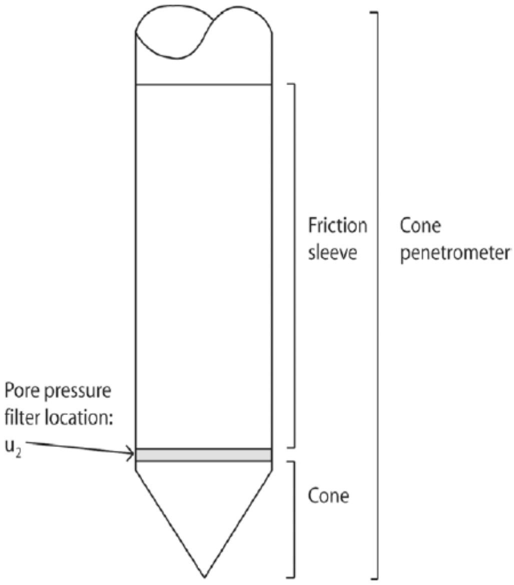


Figure 2.2: Schematic illustration of a cone penetrometer (Robertson, 2015).

2.2.1 Soil profiling

This section provides a review on empirical correlations to determine parameters required for soil profiling. It is worth mentioning that not all empirical correlations discussed in this section are applied when testing the system. Soil profiling is regarded as a pre-processing part of the system, since the system currently only focuses on coarse-grained soils. However, as mentioned earlier, the proposed strategy of the parameter determination system should allow the system to be extended to a wider range of soils. Therefore, it is still useful to include empirical correlations that can be beneficial for future development of the system.

2.2.1.1 Soil behavioural type

Since the arrival of the CPT, several authors have proposed empirical classification charts to describe the soil behaviour type for engineering applications (Douglas & Olsen, 1981; Jefferies & Davies, 1993; Robertson, 1990; Robertson et al., 1986; Senneset & Janbu, 1985). CPT classification charts provide guidance in determining the soil heterogeneity and typically involve a few basic concepts (Schnaid, 2009): (*i*) cone tip resistances are relatively higher in sands and decrease with increasing fines content, (*ii*) sleeve resistances are relatively lower in sands and increase with increasing fines content, and (*iii*) pore water pressures are lower in sand, caused by high permeability, and increase with increasing fines content. These charts only are not able to predict the soil type based on grain size distribution, but can only be used as a guide to describe the soil behaviour type. Due to the empirical nature of CPT-based soil classification, interpretation methods have evolved as new and larger data bases have been collected and evaluated (Kulhawy & Mayne, 1990). Since the system focuses on sands, the soil behavioural type is not included in the final analysis.

Douglas and Olsen (1981) developed the first soil classification chart using electric CPT data. The chart (Figure 2.3) uses q_c versus F_r to classify soil into zones described by the Unified Soil Classification System (USCS), a soil classification system standardised in ASTM D2487-85 (ASTM, 1989). USCS classifies soils into two groups, coarse-grained soils (sand and gravel) and fine-grained soils (clay and silt), based on their physical characteristics: grain-size distribution and plasticity (Atterberg) limits. The chart shows that high readings in cone tip resistance and low readings in friction ratio are typically found in sands. Conversely, low readings in cone tip resistance and high readings in friction ratio are typically found in clays. Organic soils (e.g. peat) tend to show low readings in cone tip resistance and very high readings in friction ratio, whereas sensitive soils tend to show low readings in both cone tip resistance and friction ratio

Robertson et al. (1986) suggested a chart based on the soil behaviour type (SBT), since the cone reacts to the in situ mechanical behaviour of the soil (e.g. strength, stiffness, compressibility and drainage) rather than the physical characteristics (e.g. grain-size distribution and plasticity). The chart uses CPTU data, where the cone tip resistance is corrected for pore water pressures acting on the geometry of the cone, known as "unequal area effects" (Campanella, Gillespie, & Robertson, 1982; Jamiolkowski et al., 1985):

$$q_t = q_c + u_2(1 - a) \tag{2.1}$$

in which q_t is the corrected cone tip resistance, q_c is the measured cone tip resistance, u_2 is the pore water pressure measured between the cone and the friction sleeve and a is the cone area ratio, typically around 0.8. The relationship between the cone tip resistance and the sleeve friction is described by:

$$R_f = \frac{f_s}{q_t} \cdot 100\% \tag{2.2}$$

in which R_f is the friction ratio, f_s is the sleeve friction and q_t is the corrected cone tip resistance. Typically, readings in q_t and u_2 are used to predict mechanical or consolidation properties while f_s readings are mainly used to classify soil and estimate the unit soil weight γ Robertson et al. (1986) also added a chart using the pore pressure ratio, defined as:

$$B_q = \frac{u_2 - u_0}{q_t - \sigma_v} \tag{2.3}$$

in which B_q is the pore pressure ratio, u_2 is the measured pore water pressure between the cone and the friction sleeve, u_0 is the in situ pore water pressure, q_t is the corrected cone tip resistance and σ_v is the total overburden stress. The charts separate the soil types into twelve SBT zones, as shown in Figure 2.4.

Robertson (1990) updated the SBT charts after Wroth (1988) recommended to express classification charts using normalised parameters to account for overburden stress effects:

$$Q_t = \frac{q_t - \sigma_{v0}}{\sigma'_{v0}} \tag{2.4}$$

$$F_r = \frac{f_s}{q_t - \sigma_{v0}} \tag{2.5}$$

$$B_q = \frac{u_2 - u_0}{q_t - \sigma_v} \tag{2.6}$$

in which Q_t is the normalised cone tip resistance, F_r is the normalised friction ratio, B_q is the pore pressure ratio and σ'_{v0} is the effective overburden stress ($= \sigma_v - u_0$). The chart based on F_r is considered more reliable than the chart based on B_q, since lack of repeatability in readings of measured pore water pressure (u_2) can occur due to loss of saturation. This particularly occurs at CPT locations onshore with deep water tables or in very stiff soils (Robertson, 2016). The SBT charts are shown in Figure 2.5.

Jefferies and Davies (1991) modified the normalised SBT chart by combining the pore pressure ratio with the normalised cone tip resistance, $Q_t(1-B_q)$, as illustrated in Figure 2.6. Soft sensitive soils can cause unreliable results if $B_q > 1$. To overcome this problem, Jefferies and Been (2006) updated the chart by using the parameter $Q_t(1 - B_q) + 1$. Jefferies and Davies (1993) also suggested that Q_t and F_r can be combined to describe the SBT index, I_c, as a value to express the radius of the concentric circles representing the boundaries between the normalised soil behaviour type (SBTn) zones in the Q_t-F_r chart, later modified by Robertson and Wride (1998) to apply to the Robertson (1990) Q_t-F_r chart:

$$I_c = [(3.47 - log(Q_t))^2 + (log(F_r) + 1.22)^2]^{0.5} \tag{2.7}$$

Robertson (2009) modified the normalised cone tip resistance with a variable stress exponent, n, to use it in the SBTn chart by Robertson (1990):

$$Q_{tn} = [(q_t - \sigma_v)/p_a](p_a/\sigma'_{v0})^n \tag{2.8}$$

in which Q_{tn} is the modified normalised cone tip resistance, $(q_t - \sigma_v)/p_a$ is the dimensionless net cone tip resistance, $(p_a/\sigma'_{v0})^n)$ is the stress normalisation factor, p_a is the atmospheric reference pressure and n is the stress exponent:

$$n = 0.381 \cdot I_c + 0.05(\sigma'_{v0}/p_a) - 0.15 \tag{2.9}$$

with $n \leq 1$ and I_c also determined by Q_{tn}. The updated SBTn chart is shown in Figure 2.7.

Robertson (2010) proposed an updated SBT chart using non-normalised CPT results:

$$I_{SBT} = [(3.47 - log_{10}(\frac{q_c}{p_a}))^2 + (log_{10}(R_f) + 1.22)^2]^{0.5} \tag{2.10}$$

in which I_{SBT} is the non-normalised SBT index, q_c is the cone tip resistance (or corrected cone tip resistance, q_t). I_{SBT} is generally less reliable than I_c, however, the difference is often negligible if the overburden effective stress ranges from 50 to 150 kPa. Since the overburden effective stress is often unknown, it is easier to use non-normalised CPT results in practice. The chart is shown in Figure 2.8.

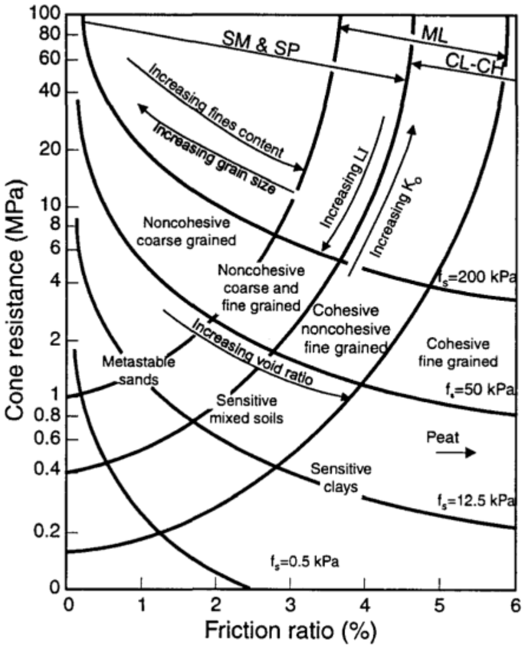


Figure 2.3: Soil classification chart based on electric CPT data (Douglas & Olsen, 1981).

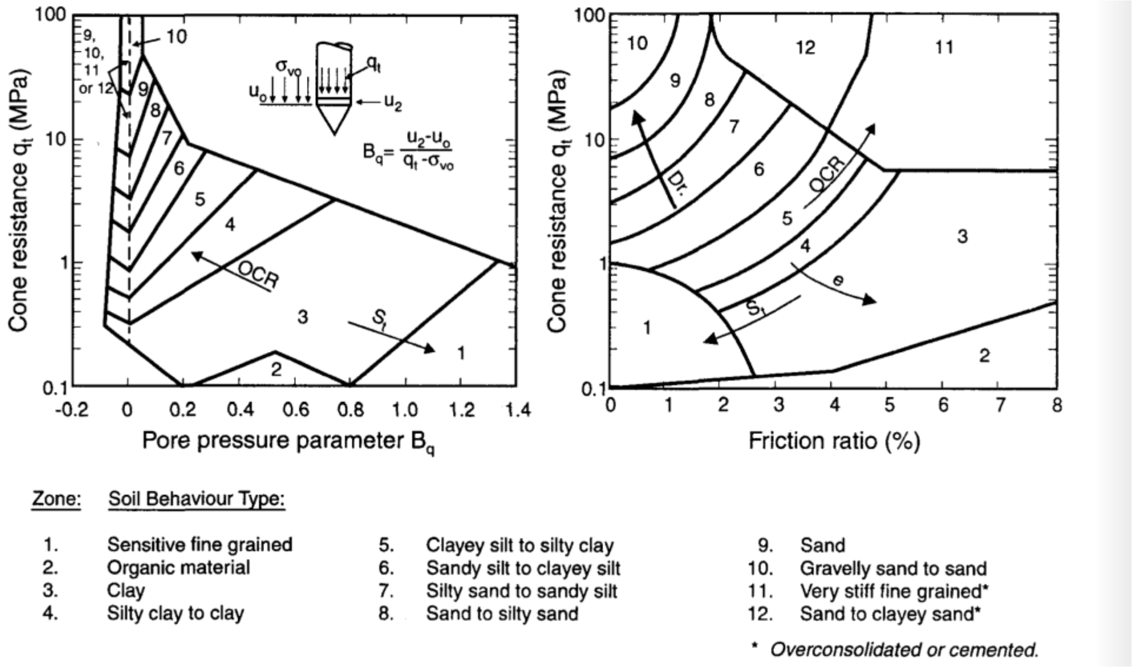


Figure 2.4: Soil behavioural type (SBT) classification system based on CPTU data (Robertson et al., 1986).

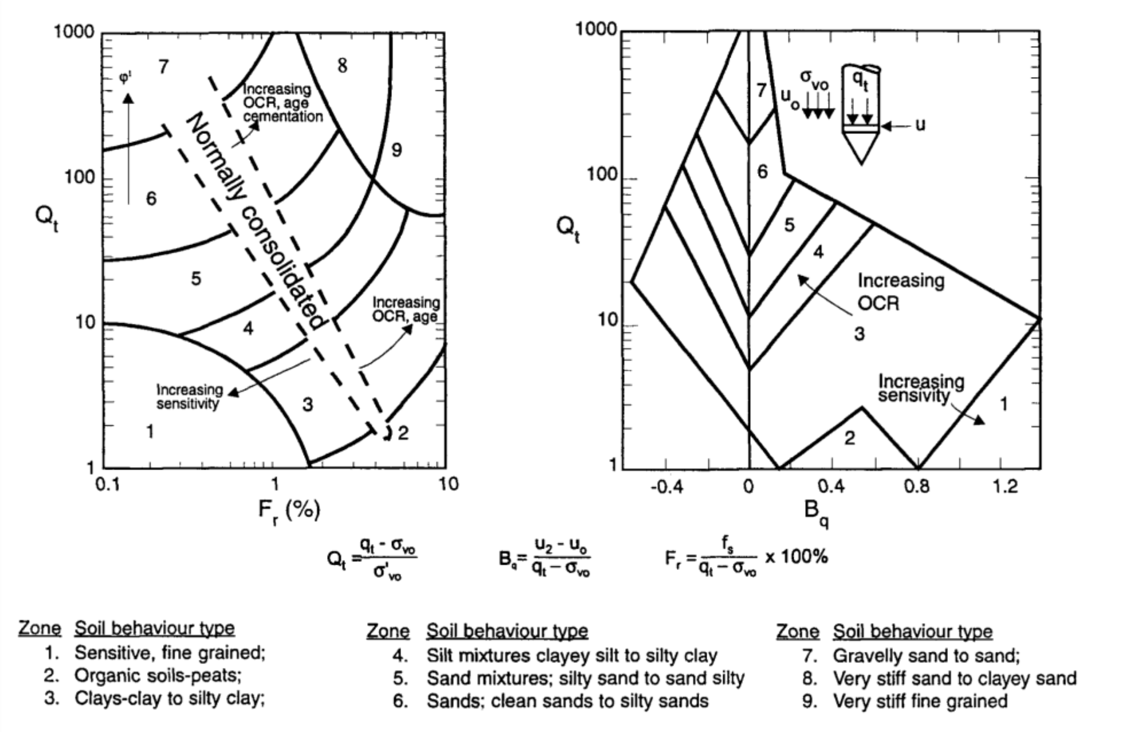


Figure 2.5: Soil behavioural type (SBT) classification chart based on normalised CPTU data (Robertson, 1990).

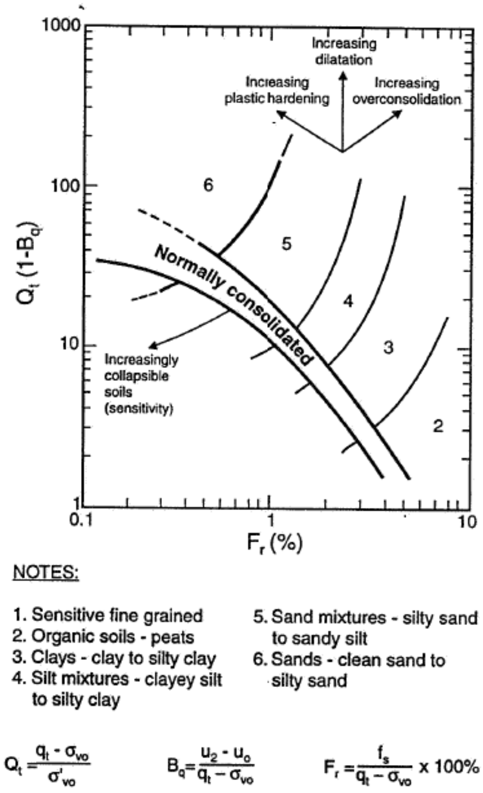


Figure 2.6: Modified normalised soil behavioural type (SBT) classification chart based on CPTU data (Jefferies & Davies, 1991).

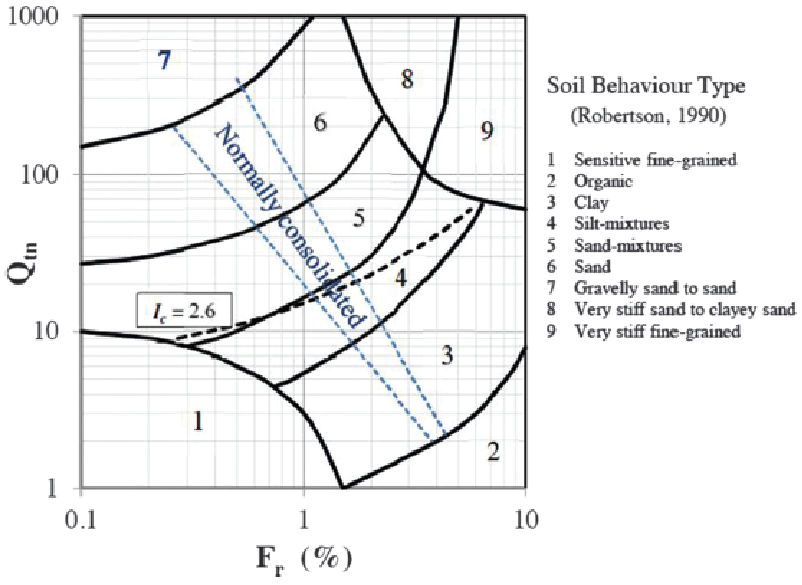
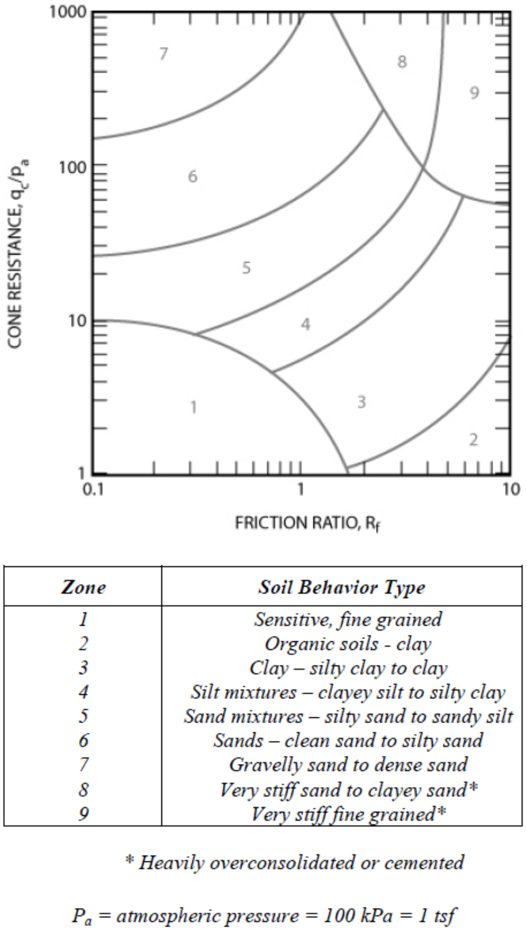


Figure 2.7: Soil behavioural type (SBT) classification chart by Robertson (1990), updated by Robertson (2009).



Zone	*Soil Behavior Type*
1	*Sensitive, fine grained*
2	*Organic soils - clay*
3	*Clay – silty clay to clay*
4	*Silt mixtures – clayey silt to silty clay*
5	*Sand mixtures – silty sand to sandy silt*
6	*Sands – clean sand to silty sand*
7	*Gravelly sand to dense sand*
8	*Very stiff sand to clayey sand**
9	*Very stiff fine grained**

** Heavily overconsolidated or cemented*

P_a = atmospheric pressure = 100 kPa = 1 tsf

Figure 2.8: Updated soil behavioural type (SBT) classification chart using non-normalised CPT data (Robertson, 2010).

2.2.1.2 Soil unit weight

The total soil unit weight, γ_t, is an important parameter since CPTU measurements involve the evaluation of total and effective overburden stresses ($\sigma'_{v0} = \sigma_{v0} - u_0$) and it is required for the calculation of many other subsequent parameters. The following empirical correlations apply for most soil types. To prove the viability of the system, it is unnecessary to include all empirical correlations, therefore only one is selected for the final analysis.

Lunne et al. (1997) proposed a way to estimate the total soil unit weight by using the zones described in the SBTn chart by Robertson et al. (1986) which relate to the discrete unit weight values given in Table 2.1. This method provides reasonable estimates for the soil unit weight, however, due to variations in soil density it is more preferred to use a continuous function.

Table 2.1: Approximate soil unit weight (γ_t) based on the soil behavioural type classification system in Figure 2.4 (Lunne et al., 1997).

Zone	Approximate unit weight (kN/m^3)
1	17.5
2	12.5
3	17.5
4	18.0
5	18.0
6	18.0
7	18.5
8	19.0
9	19.5
10	20.0
11	20.5
12	19.0

Mayne (2007) found a relationship between the total unit weight and the shear wave velocity, V_s, in case of seismic piezocone (SCPTU) data:

$$\gamma_t = 8.32log_{10}(V_s) - 1.61log_{10}(z) \tag{2.11}$$

in which V_s is in m/s and depth z is in m. This relationship provides reasonable and reliable estimates since the use of SCPTU provides four measurements with depth: q_t, f_s, u_2 and V_s (Mayne, 2014).

Robertson and Cabal (2010) suggested a continuous relationship to estimate the total soil unit weight based on only CPT measurements, q_t (or q_c) and f_s. The chart is shown in Figure 2.10, where the soil unit weight is a function of the corrected cone tip resistance, q_t, and the friction ratio, R_f, using the following equation:

$$\gamma_t = \gamma_w \cdot [0.27log_{10}(R_f) + 0.36log_{10}(q_t/p_a) + 1.236] \tag{2.12}$$

The equation is limited to sands and clays where $\gamma_t > 15$ kN/m^3 and also assumes that the soil unit weight increases with increasing friction ratio for all soil types, while typically the opposite is found in soft clays and organic soils (Lengkeek et al., 2018).

Mayne, Peuchen, and Bouwmeester (2010) proposed an equation to estimate the total unit weight for a wider range of soils including clays and silts based on only f_s and σ'_{v0}:

$$\gamma_t = 1.95 \cdot (\sigma'_{v0}/p_a)^{0.06} \cdot (f_s/p_a)^{0.06} \tag{2.13}$$

in which σ'_{v0}, p_a and f_s are given in kN/m^2. The equation provides reasonable estimates of the soil unit weight without any reliance on q_t or u_2, however, the equation requires a first estimate of the overburden effective stress. The equation is applicable for $\gamma_t > 12$ kN/m^3.

Mayne and Peuchen (2012) recommended a relationship to directly obtain the soil unit weight based on the measured sleeve friction, f_s, only:

$$\gamma_t = 26 - \frac{14}{1 + [0.5log_{10}(f_s + 1)]^2} \tag{2.14}$$

in which f_s is in kN/m^2. Of the three measured CPTU readings, it is commonly known that the sleeve friction suffers from a lack of reliability. However, Figure 2.9 shows that the sleeve friction f_s ranges between 1 kPa and 1000 kPa whereas the soil unit weight γ_t ranges between 12 and 23 kN/m^3. This implies that an accurate measurement of the sleeve friction is not required since the variance of the estimated soil unit weight is expected to be approximately 1.5 kN/m^3 (Mayne, 2014). The relationship provides reasonable soil unit weight estimates for sands, silts and clays, but is less suitable for organic soils, as seen in Figure 2.9.

Lengkeek et al. (2018) proposed a relationship to predict the total soil unit weight from CPT data which can be extended to soft and organic soils, typically found in Dutch soils. The relationship is given by:

$$\gamma_t = \gamma_{t,ref} - \beta \cdot \frac{log_{10}(q_{t,ref}/q_t)}{log_{10}(R_{f,ref}/R_f)} \tag{2.15}$$

in which $\gamma_{t,ref}$ is the reference unit weight at which q_t is independent of R_f, $q_{t,ref}$ is the reference corrected cone tip resistance at which γ_t is independent of R_f, $R_{f,ref}$ is the reference friction ratio located at the peak of all lines of equal unit weight and β represents the inclination of the equal unit weight lines. The total unit weight decreases as the friction ratio increases if q_t<$q_{t,ref}$. The proposed parameters for the equation are listed in Table 2.2. According to Lengkeek et al. (2018), the proposed relationship provides more reliable soil unit weight estimates since no iterative procedure is required and the application domain is extended to soft and organic soils, as shown in Figure 2.11.

Final selection. Considering that the system is applied on sands and that accurate sleeve friction estimates are not necessary, Equation 2.14 by Mayne and Peuchen (2012) is selected to estimate the total soil unit weight for the final analysis.

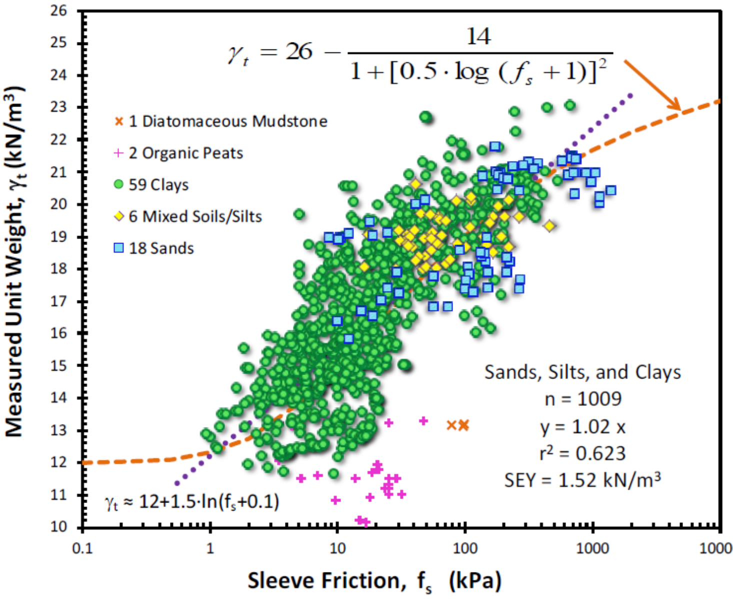


Figure 2.9: Proposed relationship to estimate the total soil unit weight based on the sleeve friction (Mayne, 2014).

Table 2.2: Proposed parameters for Equation 2.15 (Lengkeek et al., 2018).

Parameter	Proposed values
$\gamma_{t,ref}$ [kN/m^3]	19.0
$q_{t,ref}$ [-]	5.0
$R_{f,ref}$ [-]	30.0
β	[-] 4.1

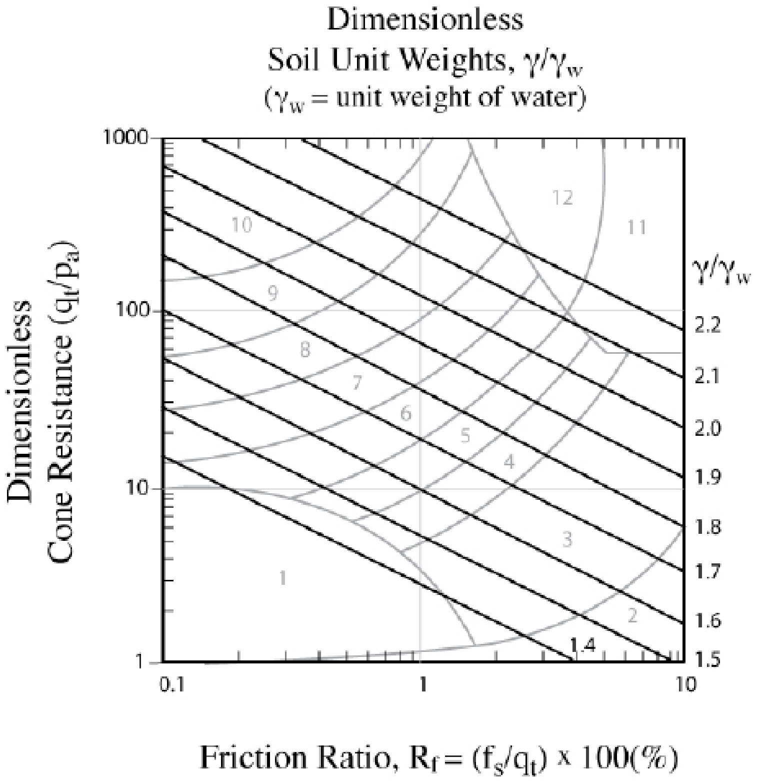


Figure 2.10: Proposed relationship to estimate the total soil unit weight based on the cone tip resistance and the friction ratio (Robertson & Cabal, 2010).

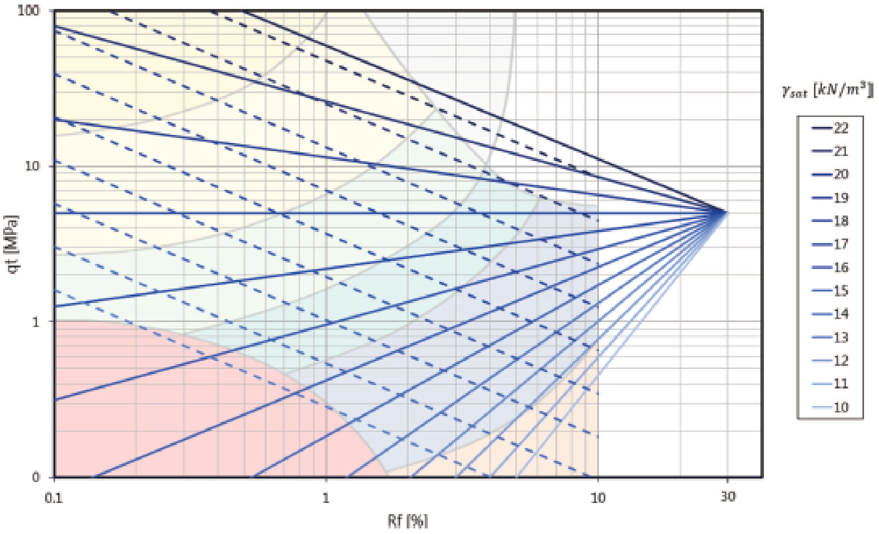


Figure 2.11: Proposed relationship to estimate the total soil unit weight based on CPT measurements, extended to soft and organic soils (Lengkeek et al., 2018).

2.2.2 Interpretation in coarse-grained soils

The framework focuses on determining strength and stiffness parameters in coarse-grained soils which are required in most constitutive models, such as the Hardening Soil model. While some of these parameters can be used directly in constitutive models as input (e.g., peak friction angle and dilatancy angle), other parameters may first have to be converted to a reference stress level (e.g., stiffness moduli). In other words, some soil parameters may also serve as constitutive model parameters. Although the main focus lies on parameter determination in coarse-grained soils, the proposed strategy may be extended to a wider range of soil types.

2.2.2.1 State parameters

Relative density (D_r)

The relative density, D_r, is often used as an intermediate soil parameter and is defined as:

$$D_r = \frac{e_{max} - e}{e_{max} - e_{min}} \tag{2.16}$$

in which e_{max} is the maximum void ratio, e_{min} is the minimum void ratio and e is the in situ void ratio. Although the relative density has long been used to express the compactibility of sands, it is becoming more common to use the state parameter related to the critical-state line, Ψ, due to its application to critical state soil mechanics and ability to understand problems related to soil liquefaction (Jefferies & Been, 2006; Robertson, 2009, 2010).

Lunne and Christoffersen (1983) suggested to estimate the relative density for young silica sands using:

$$D_r(\%) = (1/2.91) ln \frac{q_c}{60(\sigma'_{v0})^{0.7}} \tag{2.17}$$

Jamiolkowski et al. (1985) proposed the following expression to estimate the relative density:

$$D_r(\%) = -98 + 66 log_{10} \frac{q_c}{(\sigma'_{v0})^{0.5}} \tag{2.18}$$

in which q_c and σ'_{v0} are given in t/m^2. This equation can also be expressed more conveniently as (Kulhawy & Mayne, 1990):

$$D_r(\%) = 68[log_{10}(q_{t1}) - 1] \tag{2.19}$$

in which $q_{t1} = (q_t/p_a)/(\sigma'_{v0}/p_a)^{0.5}$ is the normalised cone tip resistance, similar in magnitude to Q_{tn} (Robertson, 2009).

Kulhawy and Mayne (1990) recommended the evaluation of the relative density using the normalised cone tip resistance and taking the stress history into account:

$$D_r(\%) = 100\sqrt{\left(\frac{q_{t1}}{305 \cdot OCR^{0.2}}\right)} \tag{2.20}$$

in which OCR is the overconsolidation ratio. The relationship applies to normally-consolidated (NC) to overconsolidated (OC) quartz to silica sands, as shown in Figure 2.12.

Final selection. All empirical correlations for the relative density proposed by the abovementioned authors are selected for the final analysis, which are: Equation 2.17, Equation 2.19 and Equation 2.20.

Overconsolidation ratio (OCR) and in situ stress ratio (K_0)

Evaluating the stress history for coarse-grained soils is a challenging task, as undisturbed samples of sands and silts are difficult to obtain and determination of the yield stress is difficult due to the flat oedometric $e - log(\sigma'_v)$-curve (Mayne, 2007). The overconsolidation ratio, OCR, is defined as the ratio between the maximum past effective consolidation stress and the present effective overburden stress: $OCR = \sigma'_p/\sigma'_{v0}$. The in situ stress ratio, K_0, is defined as the ratio between the horizontal effective stress and the vertical effective stress: $K_0 = \sigma'_{h0}/\sigma'_{v0}$.

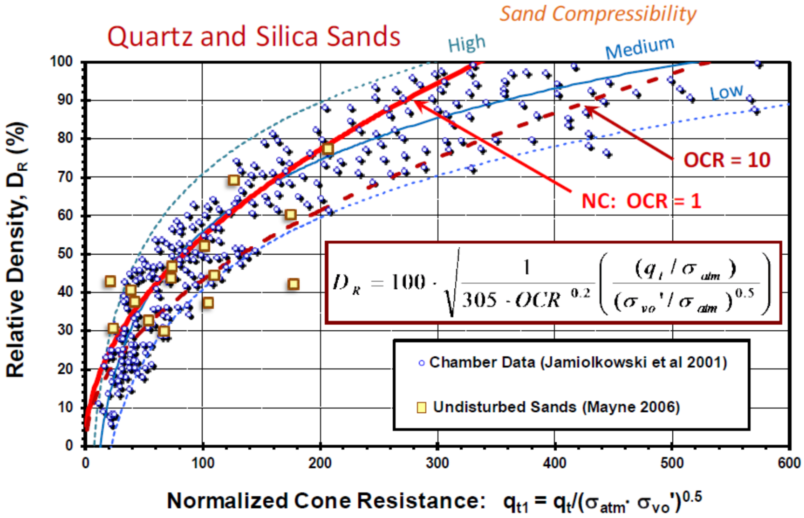


Figure 2.12: Estimation of relative density from CPT data using Equation 2.20 (Kulhawy & Mayne, 1990) from Mayne (2014).

Mayne (2005) presented a relationship to evaluate the overconsolidation ratio, OCR:

$$OCR = \left[\frac{0.192 \cdot (q_t/p_a)^{0.22}}{(1 - sin\phi') \cdot (\sigma'_{v0}/p_a)^{0.31}} \right]^{\left(\frac{1}{sin\phi' - 0.27} \right)} \tag{2.21}$$

in which ϕ'_p is the effective peak angle of internal friction. The in situ stress ratio, K_0 can then be evaluated using the following expression as derived from laboratory tests on sands (Mayne, 2007):

$$K_0 = 0.192 \cdot (q_t/p_a)^{0.22} \cdot (p_a/\sigma'_{v0})^{0.31} \cdot OCR^{0.27} \tag{2.22}$$

Mayne (2009) suggested a power law expression as a first-order estimate for the effective yield stress:

$$\sigma'_p = 0.33(q_t - \sigma_{v0})^{m'} (p_a/100)^{1-m'} \tag{2.23}$$

in which the exponent m' increases with decreasing mean particle size: $m' \approx 0.72$ in clean quartz to silica sands, $m' \approx 0.8$ in silty sands and $m' \approx 0.85$ in silts (Mayne, 2013). For young and uncemented soils, m' can also be assessed directly based on the soil behaviour type index, I_c:

$$m' = 1 - \frac{0.28}{1 + (I_c/2.65)^{25}} \tag{2.24}$$

The overconsolidation stress, OCR, can then be calculated by:

$$OCR = \frac{\sigma'_p}{\sigma'_{v0}} = \frac{0.33(q_t - \sigma_{v0})^{m'} (p_a/100)^{1-m'}}{\sigma'_{v0}} \tag{2.25}$$

Mayne and Kulhawy (1982) proposed the following relationship for estimating the in situ stress ratio in uncemented sands:

$$K_0 = (1 - sin\phi'_p) \cdot OCR^{sin\phi'_p} \tag{2.26}$$

Final selection. For the final analysis only calculation of OCR is considered since it is required for the calculation of the relative density D_r using Equation 2.20 by Kulhawy and Mayne (1990). The empirical correlation selected for calculating OCR is Equation 2.25 with $m' = 0.7$ (Mayne, 2013).

2.2.2.2 Strength parameters

Peak friction angle (ϕ'_p) and dilatancy angle (ψ_p)

The shear strength of coarse-grained soils is commonly described by the peak friction angle, ϕ'_p. The peak friction angle is composed of two components: (i) the ultimate constant volume (or critical-state) friction angle, ϕ_{cv}, which is a function of angularity, grading and mineralogy (Stroud, 1988) and (ii) the peak angle of dilation, ψ_p, which is a measure for the soil to change volume during shearing related to the packing of particles and ambient stress level (i.e., dilatancy angle increases with increasing relative density and reduces with increasing stress level). Together, the two components form the peak friction angle:

$$\phi'_p \approx \phi'_{cs} + \psi_p \tag{2.27}$$

Robertson and Campanella (1983) described a relationship to estimate the peak friction angle, ϕ'_p, from large CPT calibration chamber tests which were not corrected for boundary size effects and can therefore be conveniently expressed as (Mayne, 2006b):

$$\phi'_p = arctan[0.1 + 0.38log(q_t/\sigma'_{v0})] \tag{2.28}$$

in which ϕ'_p is in degrees. Figure 2.13 shows that this relationship can results in an overestimation of the peak friction angle for medium to high normalised cone tip stresses where q_t/σ'_{v0}>60. For lower normalised cone tip stresses, the relationship results in an underestimation of the peak friction angle.

Kulhawy and Mayne (1990) suggested the following relationship to derive the peak friction angle, ϕ'_p, based on CPT calibration chamber data:

$$\phi'_p = 17.6 + 11.0log(q_{t1}) \tag{2.29}$$

in which ϕ'_p is in degrees. This relationship mainly applies to quartz-silica sands.

Bolton (1986) presented the following equation to estimate the peak angle of dilatancy, ψ_p:

$$\psi_p = \phi'_p - \phi'_{cs} = m[D_r(Q - ln(p') - R] \tag{2.30}$$

in which p' is the mean effective stress, R is a fitting coefficient found equal to 1 for the evaluated test data and conditions, Q is a soil mineralogy and compressibility coefficient ranging from 10 for silica sands to 7 for calcareous sands (Jamiolkowski et al., 1985) and m is a stress path dependent parameter taken as 5 for plane strain conditions and 3 for triaxial conditions (Jamiolkowski, 2001). The equation allows evaluation of the friction angle with different mineralogy and/or grain size distribution once the cone tip resistance, q_c, and the mean effective stress, p', are known.

Brinkgreve, Engin, and Engin (2010) presented formulas to derive the Hardening Soil Small (HSS) model parameters for sand, derived by regression analysis on collected soil data, e.g. from Jefferies and Been (2006). The formulas for evaluating the peak friction angle and the dilatancy angle are:

$$\phi'_p = 28 + 12.5 \cdot D_r/100 \tag{2.31}$$

$$\psi_p = -2 + 12.5 \cdot D_r/100 \tag{2.32}$$

with ϕ'_p and ψ_p in degrees. These equations apply for drained conditions (Brinkgreve et al., 2010).

Final selection. For the final analysis, all empirical correlations mentioned above for the peak friction angle and the maximum dilatancy angle, are used. These are: Equation 2.28, Equation 2.29, Equation 2.30, Equation 2.31 and Equation 2.32.

2.2.2.3 Stiffness parameters

Secant (E_{50}), unloading-reloading (E_{ur}) and oedometric (E_{oed}) moduli

Estimating stiffness parameters for sands using in situ test data is of great importance since undisturbed sand samples, which are required for sampling-testing, are difficult to obtain. In addition, deriving reliable stiffness parameters from in situ tests is difficult since stiffness varies with effective stress levels and stress history, and boundary conditions (e.g., stress levels, drainage

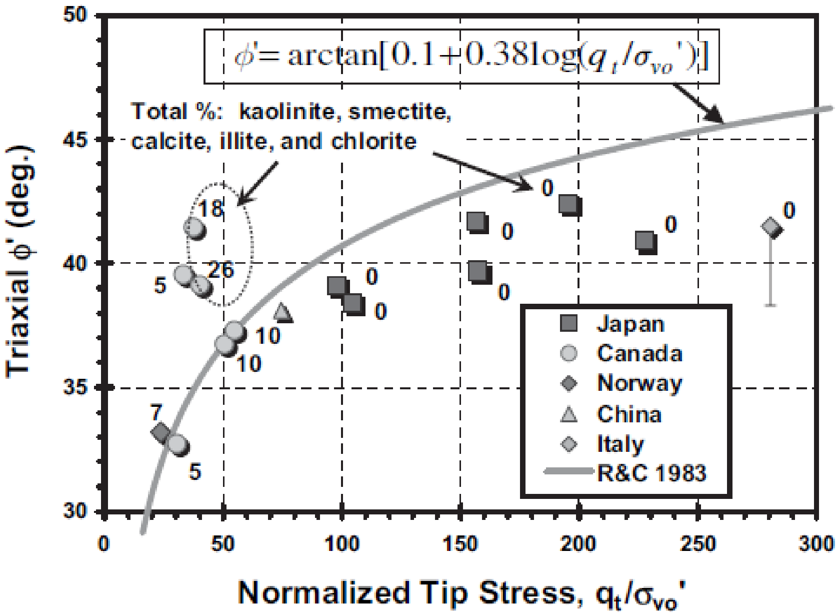


Figure 2.13: Estimation of the peak friction angle from CPT data using Equation 2.28 (Robertson & Campanella, 1983) from Mayne (2006b).

and direction of loading) cannot be controlled (Lunne et al., 1997). The elastic parameters, Young's modulus (E) and shear modulus (G), together represent the stiffness of the soil:

$$E = 2G(1+\nu) \tag{2.33}$$

in which ν is the Poisson's ratio.

Brinkgreve et al. (2010); Lengkeek (2003) presented formulas to derive the reference secant modulus, E_{50}^{ref}, the reference unloading-reloading stiffness, E_{ur}^{ref} and the reference oedometric stiffness, E_{oed}^{ref}, based on the relative density D_r as quartz sand increases linearly with the relative density:

$$E_{50}^{ref} = 60,000 \cdot D_r \tag{2.34}$$

$$E_{ur}^{ref} = 180,000 \cdot D_r \tag{2.35}$$

$$E_{oed}^{ref} = 60,000 \cdot D_r \tag{2.36}$$

in which E_{50}^{ref}, E_{oed}^{ref} and E_{ur}^{ref} are in kN/m^2. These parameters represent the reference stiffness in a stress path at a reference stress level of p_a = 100 kN/m^2.

Vermeer (2000) proposed a relationship for the stress-dependent oedometric stiffness of sands:

$$E_{oed}^{ref} = 3 \cdot q_c \sqrt{\frac{p_a}{\sigma'_{v0}}} \tag{2.37}$$

in which E_{oed}^{ref} is in kN/m^2. This relationship mainly applies for normally consolidated sands.

Final selection. For the final analysis all above-mentioned empirical correlations are selected for deriving the stiffness moduli.

2.3 Summary

In this chapter, the fundamentals of parameter determination in geotechnical engineering was discussed. A brief review was provided on experimental testing for soil interpretation: *sampling-testing* and *in situ testing*. Furthermore, a review was provided on parameter determination based on CPT data, in which the empirical correlations were presented to determine soil profiling parameters and to determine strength, state and stiffness parameters in coarse-grained soils. This study focuses on parameter determination in coarse-grained soils using CPT-based empirical relationships, however, the proposed strategy of the system allows it to be extended to other soil types and in situ tests.

Chapter 3

Graphs as Determination Method

This chapter elaborates the methodology of the parameter determination system. Section 3.1 first introduces some basic concepts and definitions of graph theory and network analysis, which are used to base the system on. Section 3.2 presents an overview of the conceptual framework for the parameter determination system. Section 3.3 describes how objects are defined in the parameter determination system, such as: nodes, edges and paths, and explains how this terminology differs from the terminology according to graph theory. Section 3.4 demonstrates how the objects in the system should be defined to generate valid paths between parameters in the system. Section 3.5 explains how the system accounts for uncertainties of parameters, in which a clarification is given why existing graph algorithms cannot be applied to the parameter determination system, and the proposed strategy to account for uncertainties is described. Finally, Section 3.6 demonstrates how the definition the objects in the system should be extended to allow for calculations to be performed for parameters involved in a path.

3.1 Introduction to graph theory and network analysis

3.1.1 Graphs

Graph theory is a branch of discrete mathematics that studies the relationship between objects in a network. One example of a network is a *social network*. In a social network, relationships between people are modelled. Modelling data as a network can increase insight into what entities or nodes are important, such as broadcasters in a network. Another example is a *transportation network*, in which the connectivity between locations are modelled, such as roads or flight paths connecting them. Such a network may be used to optimise transportation between cities. A network is described by two sets of objects: *nodes* and *edges*, which together form a network. A network is a mathematical term of a *graph*. Nodes and edges can have associated metadata, i.e., properties. An example is shown in Figure 3.1. Two friends, John and Paul, met in 2012. John and Paul are represented by the nodes, with metadata stored in a key-value pair as 'id' and 'age'. The friendship is represented as the line between the two nodes and may also have metadata such as 'date' representing the date on which they first met.

There are different types of graphs: *undirected* graphs and *directed* graphs. An undirected graph is comprised of edges without any inherent direction associated with them. An example is a metro network with bi-directional transportation. This is generally drawn as a line with no arrows between two nodes. On the other hand, a one-directional network may be found in a one-way traffic city centre. This network is a *directed* graph, since there is an inherent direction associated with the graph. Graphs can also be characterised by the presence of weights. A *weighted graph* is a graph where edges have a weight, e.g., distance, cost or time. In an *unweighted graph*, the edges do not have a weight but simply represent the link between a pair of nodes. The different types of graphs are illustrated in Figure 3.2.

3.1.2 Mathematical expression of graphs

Graphs can be expressed mathematically using: (*i*) *adjacency matrices* and (*ii*) *adjacency lists*, which are digital representations of a graph. An adjacency matrix is a square matrix in which the relation is stored from each node to every other node in a graph. An example is shown in Figure 3.3 in which a weighted directed graph with four nodes is depicted that may represent city locations forming a one-directional network. This configuration might be found in a one-way traffic city centre. The nodes are connected by directed edges with a weight. The weights may represent the time it costs to travel from one node to another node. This can be expressed digitally in an adjacency matrix. For example, the link from Node A to Node B is marked by the number "3", which represents the weight of the directed edge from Node A to Node B in the graph. A value of zero implies there is no relation from the start node to the end node. For example, Node D has no incoming edge from Node A and therefore the entry in the first row and the fourth column of the adjacency matrix is equal to zero. Although adjacency matrices are very convenient to work with, as computations can easily be performed on matrices, they use a substantial amount of memory by storing all zeros in the matrix as well (Singh & Sharma, 2012). An adjacency list combines the adjacency matrix with edge lists. For example, in Figure 3.3, Node A has two end nodes, Node B and Node C, with weights "3" and "1", respectively. Adjacency lists allow to store graphs more compact than adjacency matrices by eliminating the zeros of a matrix, reducing the waste of space (Chakraborty, Dutta, Mondal, & Nath, 2018).

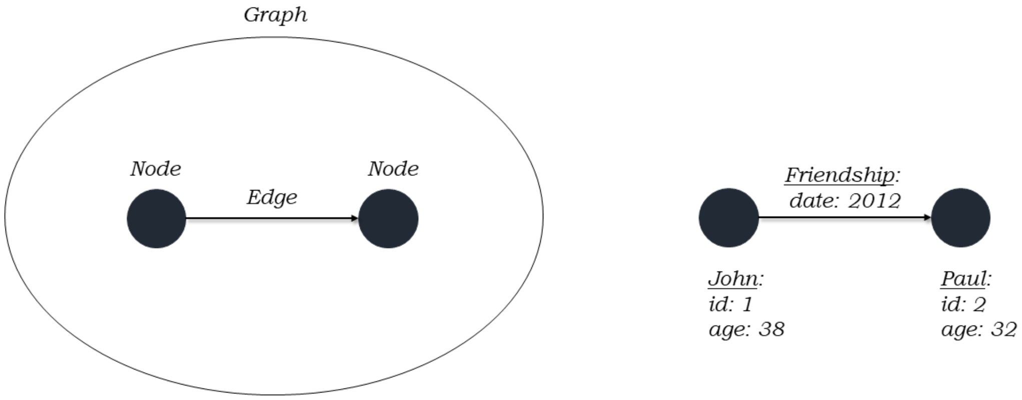


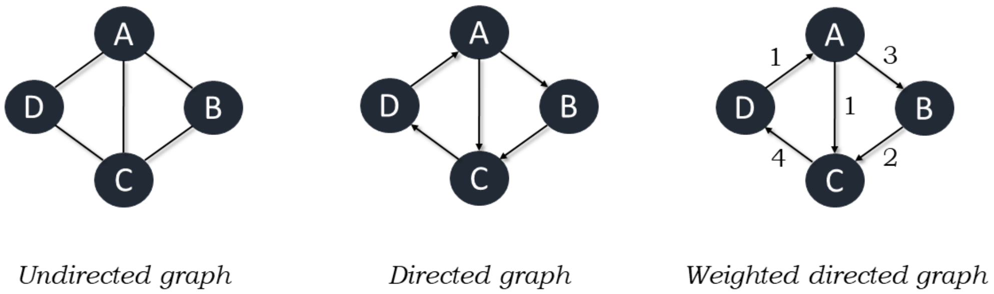
Figure 3.1: Construction of a graph.

3.1.3 Dijkstra's algorithm

The problem to find the shortest path in a network from a particular node to another node is a well-known problem and widely used in many applications, such as route navigation and computer network routing (Shu-Xi, 2012). Different types of graph algorithms exist that solve the shortest-path problem. One of best-known algorithms was developed by Edsger Dijkstra (1959), who invented an algorithm to find the lowest cost path, i.e., the shortest path, between a pair of nodes. For each node in a graph, a label is assigned that represents the minimal distance from the source node to all other nodes in a graph. Step by step, Dijkstra's algorithm attempts to improve these distance values assigned to the nodes by decreasing them. If all nodes have been visited, the algorithm terminates. The algorithm works as follows

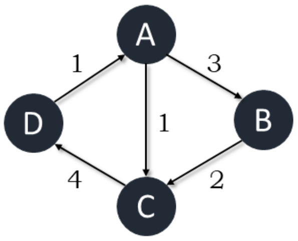
1. Assign tentative distances to all nodes of the graph: the source node is set to *zero* and all other nodes are set to *infinity*, i.e., the distance from the source node to every other node is yet unknown.

2. For each node, identify whether it has been visited: all nodes are marked as *unvisited* and the source node is marked as *current*.



Undirected graph *Directed graph* *Weighted directed graph*

Figure 3.2: Graph types



	A	B	C	D
A	0	3	1	0
B	0	0	2	0
C	0	0	0	4
D	1	0	0	0

Adjacency matrix

A	B,3	C,1
B	C,2	
C	D,4	
D	A,1	

Adjacency list

Figure 3.3: An example of a weighted directed graph and its mathematical representations.

3. For each unvisited neighbour of the current node, calculate its tentative distance. If this distance is less than the original distance assigned to the node, the distance is replaced with the newly obtained distance.

4. After having considered all neighbours of the current node, the current node is marked as visited and removed from the unvisited set.

5. Step 3 and Step 4 are repeated until all nodes of the graph have been visited. The algorithm is finished.

3.2 Conceptual framework

In the parameter determination framework an algorithm must automatically generate valid correlation paths, linking in situ test data (e.g., CPT measurements) via intermediate parameters (e.g., relative density) with constitutive model parameters (Brinkgreve, 2019). The concept is illustrated schematically in Figure 3.4. A distinction is made between *source parameters* and *derived parameters*. Source parameters have fixed values which should be specified by the user, such as CPT measurements (e.g., q_c, q_t and u_2) and the water table. Derived parameters are calculated by the system such as intermediate parameters and destination parameters. Derived parameters can also be used in other correlations. For the universality of the system, the term *method* is used instead of formula or correlation since parameters might also be derived based on tables or charts from literature. The key to finding valid paths is by defining the objects in the framework (e.g., methods) in a generic way such that it allows them to be connected with each other. The parameter determination process is deterministic, i.e., given a specific input the system should always produce the same output. Since in many situations more than one method can be selected as more correlations may be valid, different paths can lead to the same parameter. These paths can have different lengths involving different correlations and therefore often different parameters. It should be mentioned that the system is developed for determining parameters on a specific point in a soil body.

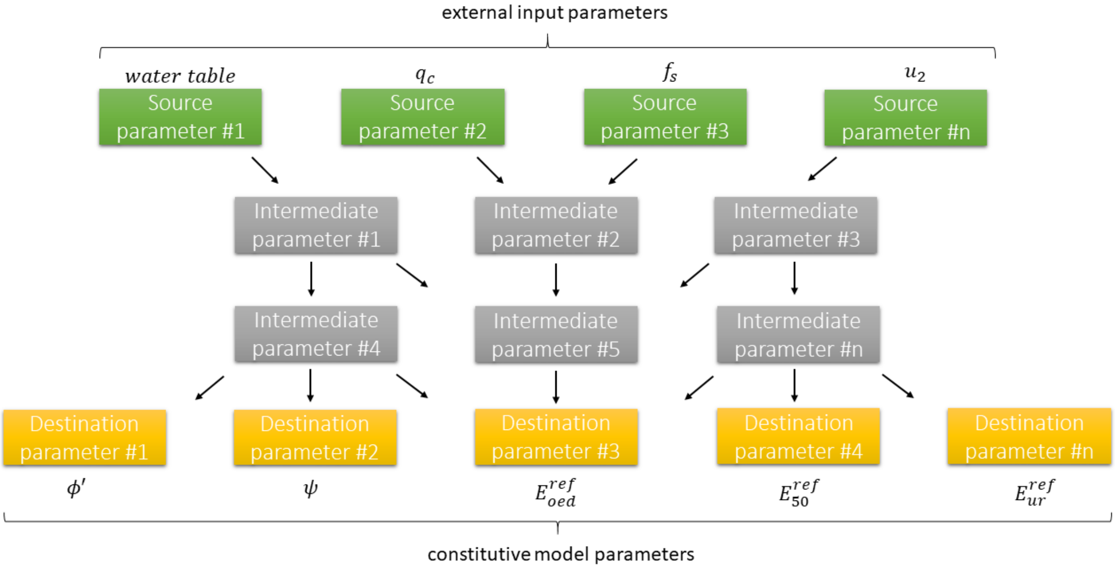


Figure 3.4: Schematic representation of the parameter determination framework.

3.3 Terminology in the framework

3.3.1 Nodes and edges

Consider the graph in Figure 3.5a. The graph describes two source nodes, Parameter A and Parameter B, and a destination node, Parameter C. Parameter A and Parameter B are used as input for the method to derive Parameter C. The method has an additional attribute containing the formula: C = A + B. This graph is defined by two types of nodes: parameters and methods, connected by edges that do not contain any additional metadata. An edge simply represents the links (relationship) between a pair of nodes, i.e., method and a parameter. An alternative way of defining the objects in a graph is shown in Figure 3.5b. Here, the graph is composed of one type of node, representing the parameters, and two edges, representing the links between the parameters. The edges now have an additional attribute, namely the method to determine Parameter C. The problem with this definition of edges is that methods are not uniquely defined. The edges might have to be grouped to indicate they both represent the method to derive C. This problem is even more noticeable in the next example, in Figure 3.6, where two methods exist to determine Parameter C. Figure 3.6a shows that Parameter C can be derived by Method C1 and Method C2. There are two *paths* leading to Parameter C, resulting in two possible outcomes for Parameter C. The graph in Figure 3.6b, where parameters are defined as nodes and methods are assigned to the edges as an attribute, shows there are four directed edges from the source parameters to the destination parameter. Unlike the graph in Figure 3.6a, where the two paths can be recognised, it is impossible to recognise the two paths in this graph (Figure 3.6b). For this reason, the parameter determination framework distinguishes two types of nodes (i.e., parameters and methods) and the edges simply represent the relationship between parameters and methods since they do not have any associated metadata.

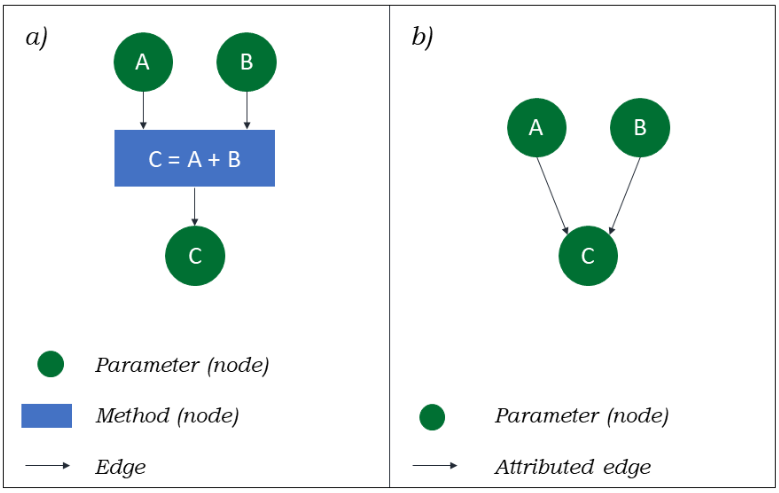


Figure 3.5: Two graphs with different definitions of objects: a) parameters and methods as nodes and b) parameters as nodes.

3.3.2 Paths

By definition, paths in graph theory do not allow branching. However, the parameter determination framework includes methods which are usually multivariable formulas. A multivariable formula involves multiple parameters (variables) and therefore methods can have multiple incoming edges. These edges form *one path*, which is in conflict with graph theory where they are defined as *separate paths*. For example, since the method in Figure 3.5 requires both input parameters, Parameter A and Parameter B, to calculate Parameter C, there is one path that leads to Parameter C. Existing graph algorithms would define the path from Parameter A to Parameter C and the path from Parameter B to Parameter C as separate paths. This will be clear later on from the first results of the first test case.

Branching paths

Figure 3.7 illustrates an example of a graph in a parameter determination framework, named Test Case I. Please note that this example does not correspond with the true situation, but it is used as an example to explain how paths are defined in a parameter determination framework. The graph consists of two types of nodes, where the blue boxes represent the *methods* and the green boxes represent the *parameters*. The goal is to determine destination parameter, ϕ', which is the peak friction angle. According to the graph, there are four possible ways (i.e., paths) to calculate parameter ϕ'. These paths are indicated with four different colours on the duplicated graph, right from the original graph in Figure 3.7. Each of these paths involve different methods

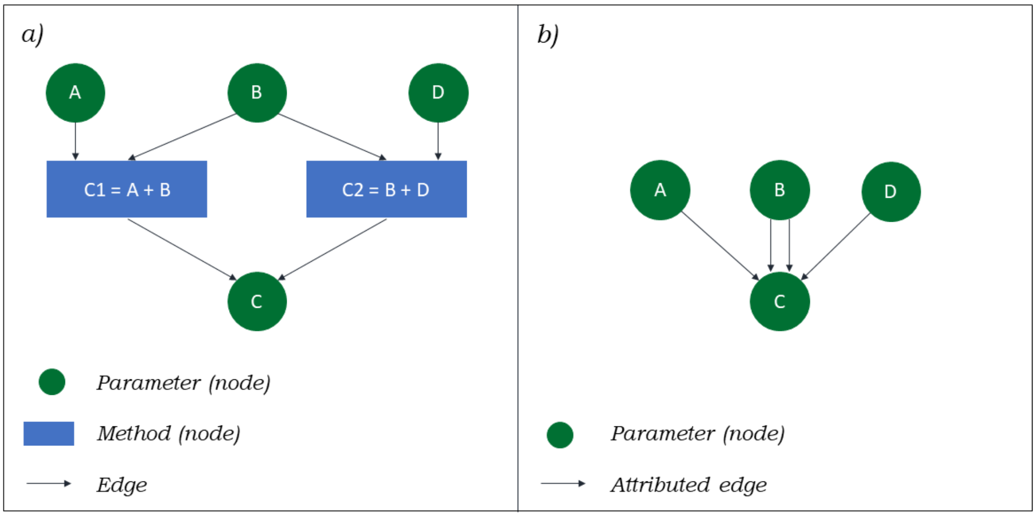


Figure 3.6: Two graphs with different definitions of objects, in which two methods exist to derive the source parameter: a) parameters and methods as nodes and b) parameters as nodes.

and therefore different parameters. Paths can end up at different sets of source parameters. The source parameters are indicated by the circles.

Establishing the external database

Paths are naturally constructed reversely. For example, a user of the system would calculate parameter ϕ' by starting with the evaluation and determination of valid methods to determine ϕ', evaluating the input parameters in each of these methods, evaluating and selecting valid methods required for each input parameter involved in these methods and so on. For the final parameter determination framework, this procedure is continued until no unknowns are left. Since these relationships are established prior to use of the system; the parameter determination procedure is deterministic. The paths terminate at the source parameters, which in this case are: q_c, u_2, σ'_{v0} and OCR. This graph uses a drawn arrow to indicate the link between a parameter and a method in which the parameter is used as input. A method with two incoming edges implies *both* parameters are needed as input, which is implicitly defined in the formula of the method. A dashed arrow indicates the link between a method and the resulting output parameter. Note that a method is always characterised by one outgoing edge, since it represents a formula to determine the resulting parameter. A parameter with multiple incoming edges implies that there are multiple paths to calculate this parameter, resulting in different parameter outcomes.

To construct paths appropriately, two external databases should be established: one consisting of the methods and one consisting of the parameters. These are all the nodes involved in a graph. It should be emphasised that the responsibility of correctly establishing the external databases lies with the user of the system. The user, i.e., the geotechnical engineer, is expected to have the required knowledge to implement the parameters and methods correctly. Circular referencing, which should always be avoided, will result in closed loops that may cause the system to crash. Circular referencing can be prevented by (i) correctly implementing the external database, in which the *user* has the responsibility like in this study, or by (ii) adding a functionality to the system which detects loops and resolves them (e.g., by indicating the invalidity of a certain path).

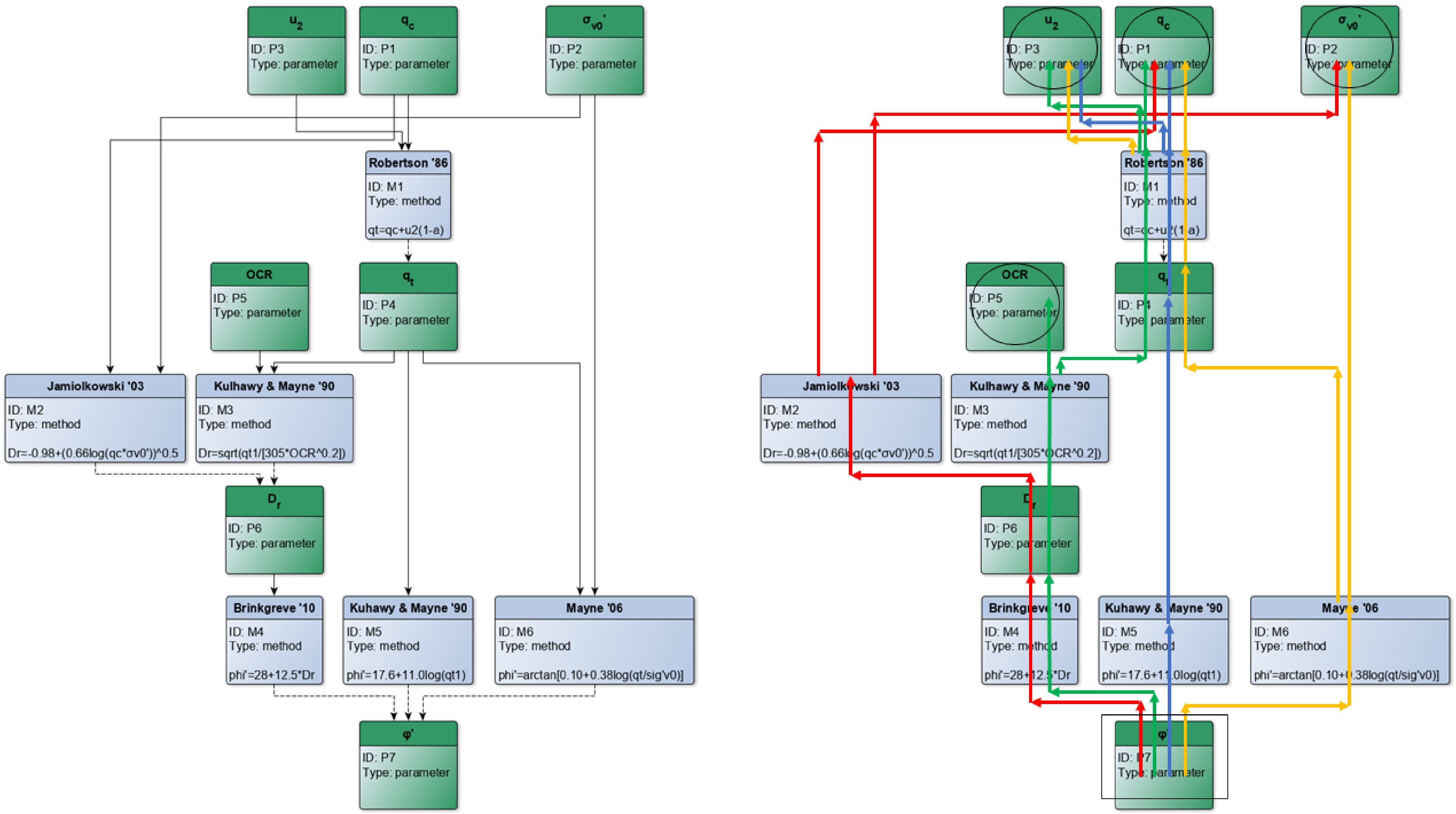


Figure 3.7: Test Case I. A directed graph with two types of nodes: methods, indicated in blue, and parameters, indicated in green. The source parameters and the destination parameter are indicated by the circles and the square, respectively. The right graph presents four paths that lead to the destination parameter, ψ', indicated by the four different colours.

3.4 Generating a path

The aim is to automatically generate edges (links) between the nodes which result in a path. A simplified version of the graph from Figure 3.7 is shown in Figure 3.8. The key to generating a path is by defining the relevant objects in a system. The graph shows there are two types of nodes: a method and a parameter. For both types of nodes we define an abstract object, i.e., an abstract method and an abstract parameter. Both of them are given metadata (properties) which are characteristic. Each type of method/parameter should inherit the properties as specified by their abstract method/parameter. Properties in this system are referred to as *attributes*. If a type of an abstract object, e.g., a specific formula or a specific parameter, is imported into the system by the user from an external database, it becomes an *instance* of the abstract object after the system implements it. An instance of an abstract object is also referred to as a *concrete object*. For the sake of simplification, formulas are not shown in this graph. They are irrelevant at this stage since we are only interested in connecting edges between parameters and not performing any calculations yet. All parameters and methods are "known" by the system since they are provided by the user. Relationships between the nodes are implicitly defined by the methods, since the methods define the input and output parameters.

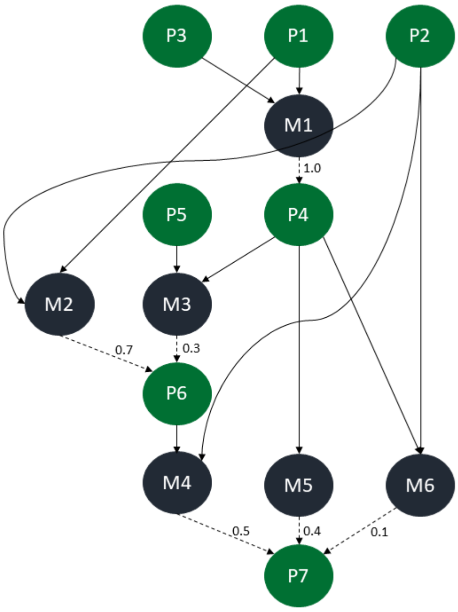


Figure 3.8: Simplified graph of Test Case I.

3.4.1 Definition of a method

Consider method M1 in Figure 3.8. M1 is linked with two input parameters, P1 and P3, and linked with its resulting output parameter, P4. M1 has a weight of 1.0, since there is no alternative method to determine P4. The abstract object for a method is defined by the following attributes: ID; input parameter(s); output parameter; weight. The implementation for method M1, for example, would be defined by the attributes: M1; P1, P3; P4; 1.0. The abstract object and implementations of a method are given in Table 3.1.

Methods can either be theory-based or empirical-based (i.e., empirical correlations) and have a weight which indicate the suitability of the formula. A theory-based formula is always given a weight of 1.0 since it is based on a physical relationship. On the other hand, the weight of an empirical correlation can never have a weight of 1.0 since it is an approximation. The suitability of an empirical correlation depends its validity and limitations. If more methods exist to determine the same parameter, i.e., there are multiple existing paths, paths can be prioritised by adopting a mathematical procedure such as a weighted average using the weights of the methods. For example, in Figure 3.8, M4, M5 and M6 have an assumed weight of 0.5, 0.4 and 0.1, respectively, implying

Table 3.1: Abstract object and concrete objects (after implementation) of a method.

Abstract object	*Concrete objects*					
ID	M1	M2	M3	M4	M5	M6
Input parameter(s)	P1, P3	P1, P2	P4, P5	P2, P6	P4	P2, P4
Output parameter	P4	P6	P6	P7	P7	P7
Weight	1.0	0.7	0.3	0.5	0.4	0.1

that M4 would have the highest suitability and M6 would have the lowest suitability. Note that these weights are fictitious. The weights are assigned to the outgoing edges. The edge between an input parameter and a method is always characterised by a weight of 1.0 and are not shown on the graph. The weighting procedure for empirical correlations requires extensive study and lies beyond the scope of this thesis since the purpose of the thesis is to focus on the logic of the system, which is developed in such a way that it can be improved and extended.

An abstract method is defined by the following attributes:

- **uid**
  Definition: This defines the unique identity of the method as specified by the user from an external database (i.e., external attribute).

- **parameters_in**
  Definition: This defines the input parameter(s) of the method as specified by the user from an external database (i.e., external attribute).

- **parameter_out**
  Definition: This defines the output parameter as specified by the user from an external database (i.e., external attribute).

- **weight**
  Definition: This defines the weight of the method. In case of a theoretical formula, the weight is equal to 1.0. In case of an empirical formula, the weight is determined based on the validity and limitations of the formula. The weight is specified by the user from an external data base (i.e., external attribute).

3.4.2 Definition of a parameter

Consider parameter P4 in Figure 3.8. P4 is linked with methods M1, M3, M5 and M6. However, this information does not need to be stored in the object of a parameter, since parameters are *unaware* of their relationship with methods. On the other hand, methods *know* what their related input parameter(s) and output parameter are, since it is defined in the formula. The abstract object and implementations of a parameter are given in Table 3.2.

Table 3.2: Abstract object and concrete objects (after implementation) of a parameter.

Abstract object	*Concrete objects*						
ID	P1	P2	P3	P4	P5	P6	P7
Weight	1.0	1.0	1.0	1.0	1.0	1.0	1.0

An abstract parameter is defined by the following attributes:

- **uid**
  Definition: This defines the unique identity of the parameter as specified by the user from an external database (i.e., external attribute).

- **weight**
  Definition: This defines the weight of the parameter which is simply the relationship of the parameter with the method. This weight is always equal to one and therefore it is defined within the system (i.e., internal attribute).

3.4.3 Definition of a connector

In order to generate a link between a method and its output parameter, an additional abstract object is created: the *connector*. The connector collects all methods and parameters from the external database and imports them into the system as dictionaries: one containing the methods and one containing the parameters. A dictionary is Python's implementation of a data structure, consisting of a list of attribute-value pairs. The connector evaluates the two dictionaries to find valid links between a method and a parameter, i.e., each method is assigned to its related parameter. This allows for the connectivity between methods and parameters to generate paths.

The abstract connector is defined by the following attributes:

- **methoddct**
  Definition: This defines the dictionary of the methods as imported from the external database containing the methods.

- **parameterdct**
  Definition: This defines the dictionary of the parameters as imported from the external database containing the methods.

- **matrix**
  Definition: This defines the adjacency matrix representing the weighted relationships (edges) between all nodes. The weights are stored in the individual instances of a method and a parameter (i.e., concrete methods and concrete parameters).

3.5 Accounting for uncertainties

Uncertainties in parameters can be caused by the natural randomness of the soil properties, i.e., aleatory uncertainties, and by a lack of information and errors in measurement and calculation, i.e., epistemic uncertainties (Lunne et al., 1997). These uncertainties are not considered when testing the system on a geotechnical example, since it is assumed that the source parameters are obtained with 100% accuracy. Any uncertainty of a parameter derived by the system is assumed to be caused only by the uncertainty of an empirical correlation. Empirical correlations are approximate relationships, based on average soil properties from different soil types, and are usually valid within a specified range or valid for certain soil types. The validity and limitations of empirical correlations should be incorporated by the system and contribute the uncertainty of a parameter.

3.5.1 Problem with existing graph algorithms

One may suggest that the more empirical correlations involved in a path, the higher the propagated uncertainty of the destination parameter since every empirical formula carries a certain degree of uncertainty. Therefore, one may assume that the shortest path leads to the parameter with the lowest uncertainty (or highest accuracy). As mentioned in the previous section, existing graph algorithms including Dijkstra's shortest path algorithm are only applicable for paths without branching, i.e., *single-source paths*. Graph algorithms would unjustly consider these edges as separate paths, as shown in Table 3.3. Ten paths are identified while in reality there are four paths. There is little point in finding the shortest path by Dijkstra's algorithm in a graph with multiple-source paths. The shortest weighted paths of the graph in (Figure 3.9b) using Dijkstra's algorithm are given in Table 3.4.

To overcome this problem, the graph may be simplified by assuming each method only has one unknown variable, i.e., a method now only has one incoming edge from a parameter node, as shown in Figure 3.9. By assuming that parameter nodes P2, P3 and P5 are "known", multivariable formulas are converted to singlevariable formulas, i.e., multiple-source paths are converted to single-source paths. Due to this simplification, graph algorithms such as Dijkstra's algorithm may be applied to find the shortest (weighted) path between two nodes in a graph. The shortest weighted paths, also referred to as the least cost paths, of the graph with single-source paths (Figure 3.9b) found by Dijkstra's algorithm are given in Table 3.4.

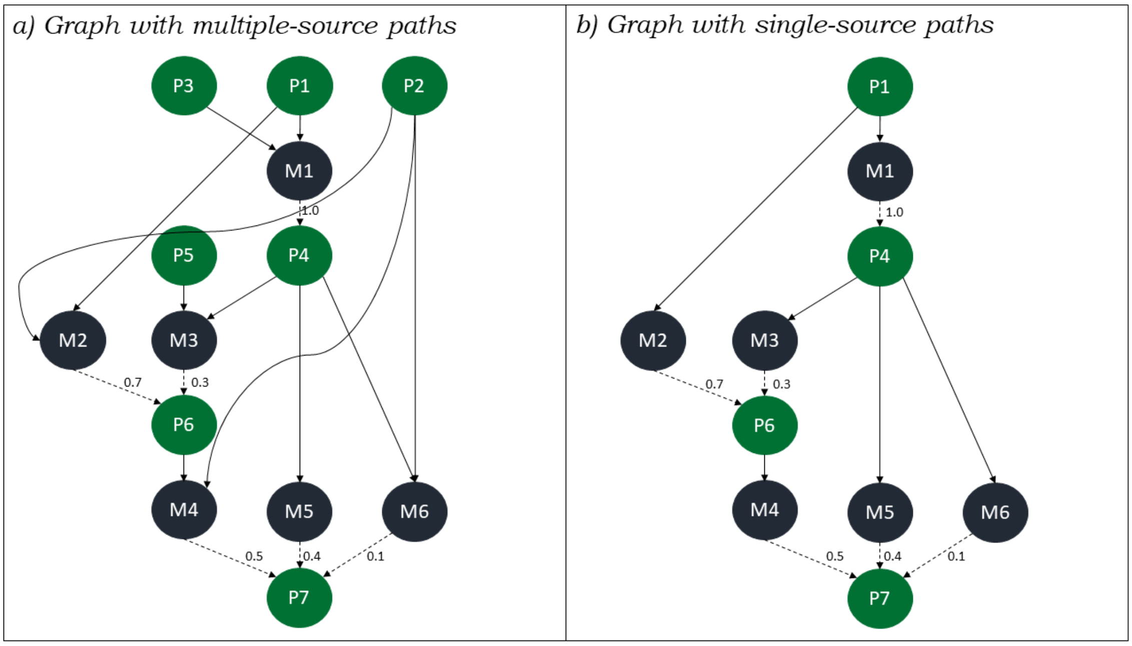
There are many concerns with this strategy. Since existing graph algorithms are unable to cope with multivariable formulas, they are unable to incorporate uncertainties of individual parameters involved in a formula. Developing the system so that it is capable of making these assumptions for each formula will overly complicate the logic of the system since objects are not uniquely defined in the system. This also has an effect on the transparency and adaptability of the system. When comparing the graph in Figure 3.9a with the graph in Figure 3.9b, it is clear that the aforementioned graph shows all relations between the objects whereas in the latter mentioned some of these relations are "hidden" in the objects, reducing the transparency of the system. Adding a new method, such as an additional empirical correlation, to the graph will cause more difficulties since the involved parameters of the method might not exist as objects in the system which in turn reduces the adaptability system.

Table 3.3: All paths from destination parameter, P7, to source parameters: P1, P2, P3 and P5.

Path (#)	Destination parameter	Source parameters					
1	**P7**	M4	P6	M2	**P1**		
2	**P7**	M6	P4	M1	**P1**		
3	**P7**	M5	P4	M1	**P1**		
4	**P7**	M4	P6	M3	P4	M1	**P1**
1	**P7**	M6	P4	M1	**P3**		
2	**P7**	M5	P4	M1	**P3**		
3	**P7**	M4	P6	M3	P4	M1	**P3**
1	**P7**	M6	**P2**				
2	**P7**	M4	P6	M2	**P2**		
1	**P7**	M4	P6	M4	**P5**		

Table 3.4: The shortest paths from destination parameter, P7, to source parameters: P1, P2, P3 and P5.

Path (#)	Destination parameter	Source parameter					
1	**P7**	M4	P6	M3	P4	M1	**P1**
2	**P7**	M4	P6	M3	P1	M1	**P3**
3	**P7**	M4	P6	M2	**P2**		
4	**P7**	M4	P6	M3	**P5**		



a) Graph with multiple-source paths

b) Graph with single-source paths

Figure 3.9: Simplification of a graph with multiple-source paths into a graph with single-source paths.

3.5.2 Proposed strategy

The problem with using the length of a path to define the suitability of a path is that individual uncertainties of the parameters involved in a multiple-source path are not taken into account. It is suggested to determine the accuracy of a multiple-source path by calculating the accuracy of each parameter involved in a path. The calculated accuracy is stored as an attribute of each derived parameter. Uncertainties propagate along the path and the resulting accuracy of the destination parameter defines the accuracy of the path. Hence, the quality of a path is measured by the accuracy of the destination parameter and not by the length of a path as described by Dijkstra's algorithm since the length does not incorporate the individual uncertainties of the parameters involved in a multiple-source path. The term accuracy is chosen to keep the system generic. This attribute can be modified by the user when the user intends to use the coefficient of variation or standard deviation for example.

Source parameters in this system are assumed to have no aleatory or epistemic uncertainties, and therefore receive an *accuracy* of 1.0. Derived parameters are determined by methods. A method carries a weight for the suitability of the formula but also involves input parameter(s) which have an accuracy. The accuracy of a derived parameter is calculated by multiplying the weight of the method with the dyadic product of the accuracy of the input parameters.

The proposed strategy to cope with multiple-source paths is by defining the abstract objects for a parameter and a method in a generic and abstract manner such that:

1. Each method is able to connect with its related input parameter(s) and output parameter.
2. Calculations can be performed for each derived parameter.
3. Information can be transferred along the path.

Hence, for each derived parameter involved in a path, the value and the uncertainty is calculated, transferred to the next method, incorporated in the calculation for the next parameter and so on. Finally, all paths are visualised on the graph where each path shows the selected parameters and methods to arrive at each destination parameter. After calculating the value and accuracy for each parameter, the parameter receives additional attributes for the resulting value and accuracy.

3.6 Performing calculations

In order to obtain the value and the accuracy for each derived parameter involved in a path, the abstract method and the abstract parameter, earlier defined in Table 3.1 and Table 3.2, require additional attributes to enable these calculations. The system distinguishes external from internal attributes as shown in Table 3.5. An extensive overview of the parameter determination architecture is given in pseudocode in Figure 3.11, showing how the abstract objects are defined and connected with each other.

Table 3.5: Overview of the definition of objects in which a distinction is made between external (i.e., user input) and internal attributes (i.e., derived by the system).

Method		**Parameter**	
Attribute	External/Internal	Attribute	External/Internal
ID	External	ID	External
Input parameter(s)	External	Symbol	External
Output parameter	External	Value[1]	External/Internal
Author	External	Accuracy[1]	External/Internal
Formula	External	Weight	External
Accformula	External	Methods	Internal
Weight	External		
Output value	Internal		
Output accuracy	Internal		
Connector	Internal		

[1] The value and accuracy of source parameters (e.g., measured CPT data) are fixed and should be specified by the user, externally. The value and accuracy of derived parameters are calculated by the system, internally.

3.6.1 Final definition of a method

The abstract method receives six additional attributes for the author, the formula, the formula for the accuracy, the weight of the method, the output value, the output accuracy and the connector. The final definition of an abstract method is described as follows. Each concrete method inherits the attributes from the abstract method.

- **uid**
  Definition: This defines the unique identity of the method as specified by the user from an external database (i.e., external attribute).
  Example: m1

- **parameters_in**
  Definition: This defines the input parameter(s) of the method as specified by the user from an external database (i.e., external attribute).
  Example: u_2, q_c, a

- **parameter_out**
  Definition: This defines the output parameter as specified by the user from an external database (i.e., external attribute).
  Example: q_t

- **author**
  Definition: This defines the name of the formula, which is usually the author of an empirical correlation followed by the year of publication. In case of a theoretical formula, it is suggested to incorporate the resulting parameter into the name. The author is specified by the user from an external database (i.e., external attribute).
  Example: Robertson1986

- **formula**
  Definition: This defines the formula to calculate the value of the output parameter and is specified by the user from an external database (i.e., external attribute).
  Example: $q_c + u_2 \cdot (1 - a)$

- **accformula**
  Definition: This defines the formula to calculate the accuracy of the output parameter by multiplying the accuracy of the input parameter(s) with each other. Note that the final accuracy is determined by multiplying this number with the weight of the method. The formula for the accuracy is specified by the user from an external database (i.e., external attribute).
  Example: $q_c \cdot u_2 \cdot a$

- **weight**
  Definition: This defines the weight of the method. In case of a theoretical formula, the weight is equal to 1.0. In case of an empirical formula, the weight is determined based on the limitations and validity of the formula. Note that these weights are fictitious numbers and do not compare with reality. To improve the weighting procedure of an empirical correlation, extensive study is required. The weight is specified by the user from an external data base (i.e., external attribute). All methods that are empirical correlations have been assigned a weight of 0.60.

- **output_value**
  Definition: This defines the value of the output parameter. If the output value does not yet exist, the function "calculate" is called which calculates the value of the output parameter using *formula* with the value of the input parameter(s) provided that these values already exist. This attribute is defined within the system (i.e., internal attribute).

- **output_accuracy**
  Definition: This defines the accuracy of the output parameter. If the output accuracy does not yet exist, the function "calculate" is called which calculates the accuracy of the output parameter using *accformula* with the accuracy of the input parameter(s) provided that these values already exist. This attribute is defined within the system (i.e., internal attribute).

- **connector**
  Definition: This attribute refers to the abstract object Connector which enables the connectivity between a method and a parameter. This abstract object (*i*) collects all methods and parameters given by the user in an external database and imports them as a dictionary, (*ii*) connects methods with parameters provided that the output of the method is equal to the parameter, (*iii*) generates a graph using the Python package graphviz (Gansner, 2011) and (*iv*) creates the adjacency matrix of the graph. This attribute is defined within the system (i.e., internal attribute).

3.6.2 Final definition of a parameter

The abstract parameter receives five additional attributes for the symbol, the unit, the weight, the value, the accuracy and the related methods. The final definition of an abstract parameter is described as follows. Each concrete parameter inherits the attributes from the abstract parameter.

- **uid**
  Definition: This defines the unique identity of the parameter as specified by the user from an external database (i.e., external attribute).
  Example: p1

- **symbol**
  Definition: This defines the symbol of the parameter as specified by the user from an external database (i.e., external attribute).
  Example: q_c

- **unit**
  Definition: This defines the unit of the parameter as specified by the user from an external database (i.e., external attribute). Parameters in the system should be in SI base units and should be used consistently throughout the system. Therefore, a constant unit is chosen for force and length which are kN and m. If an empirical correlation uses different units, the correlation should be adjusted (scaled) to the units of the parameters in the system, to keep the system consistent.
  Example: kN/m^2

- **definition**
  Definition: This defines the definition of the parameter as specified by the user from an external database (i.e., external attribute).
  Example: cone tip resistance

- **weight**
  Definition: This defines the weight of the parameter which is simply the relationship of the parameter with the method. This weight is always equal to one and therefore it is defined within the system (i.e., internal attribute).

- **value**
  Definition: This defines the output value of the parameter. In case of a source parameter, the value is specified by the user (external attribute). In case of a derived parameter, the output value is calculated by the system (i.e., internal attribute).

- **accuracy**
  Definition: This defines the output accuracy of the parameter. In case of a source parameter, the accuracy is specified by the user (i.e., external attribute). In case of a derived parameter, the output accuracy is calculated by the system (internal attribute).

- **methods**
  Definition: This defines the related methods of the parameter assigned by the abstract object Connector. If no methods are detected, the parameter is a source parameter and an empty list is returned. If methods are detected, a list of the methods is returned. This attribute is defined within the system (i.e., internal attribute).

3.7 Summary

This chapter elaborated the proposed strategy for a parameter determination system using graphs, which are used in this study to increase insight into the parameter determination process for the advanced user of the system, such as a geotechnical engineer. To summarise, an overview is shown in Figure 3.10. Using an external database (*input*) of methods and parameters, such as a spreadsheet which can be established by the user of the system, the parameter determination system collects and processes the imported data from the external database (*processing*), to generate the resulting graph (*output*). In the parameter determination system, three abstract objects are defined: Method, Parameter and Connector. The imported methods and parameters become concrete objects when these abstract objects are implemented. The Connector generates edges between methods and parameters that are linked, resulting in a path.

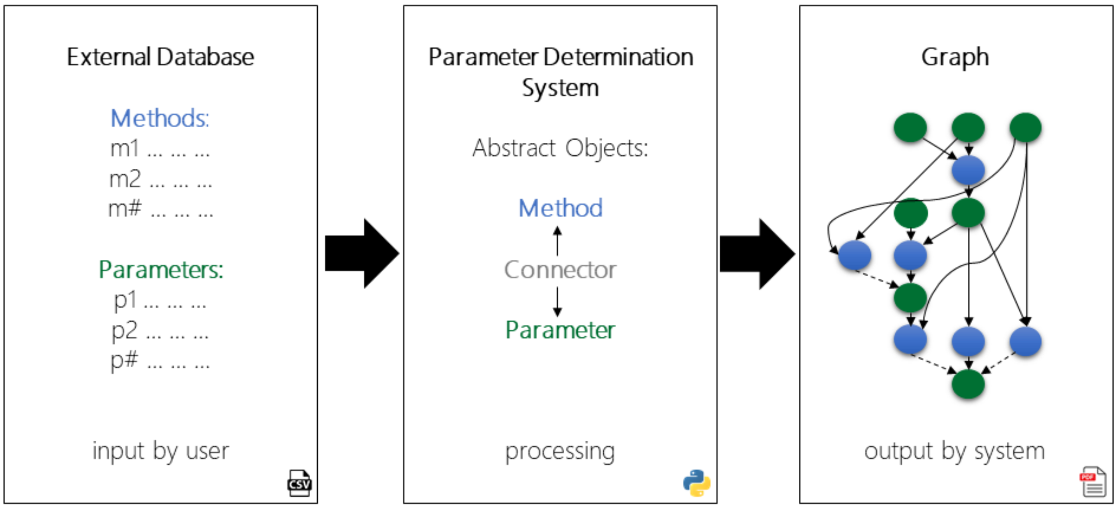
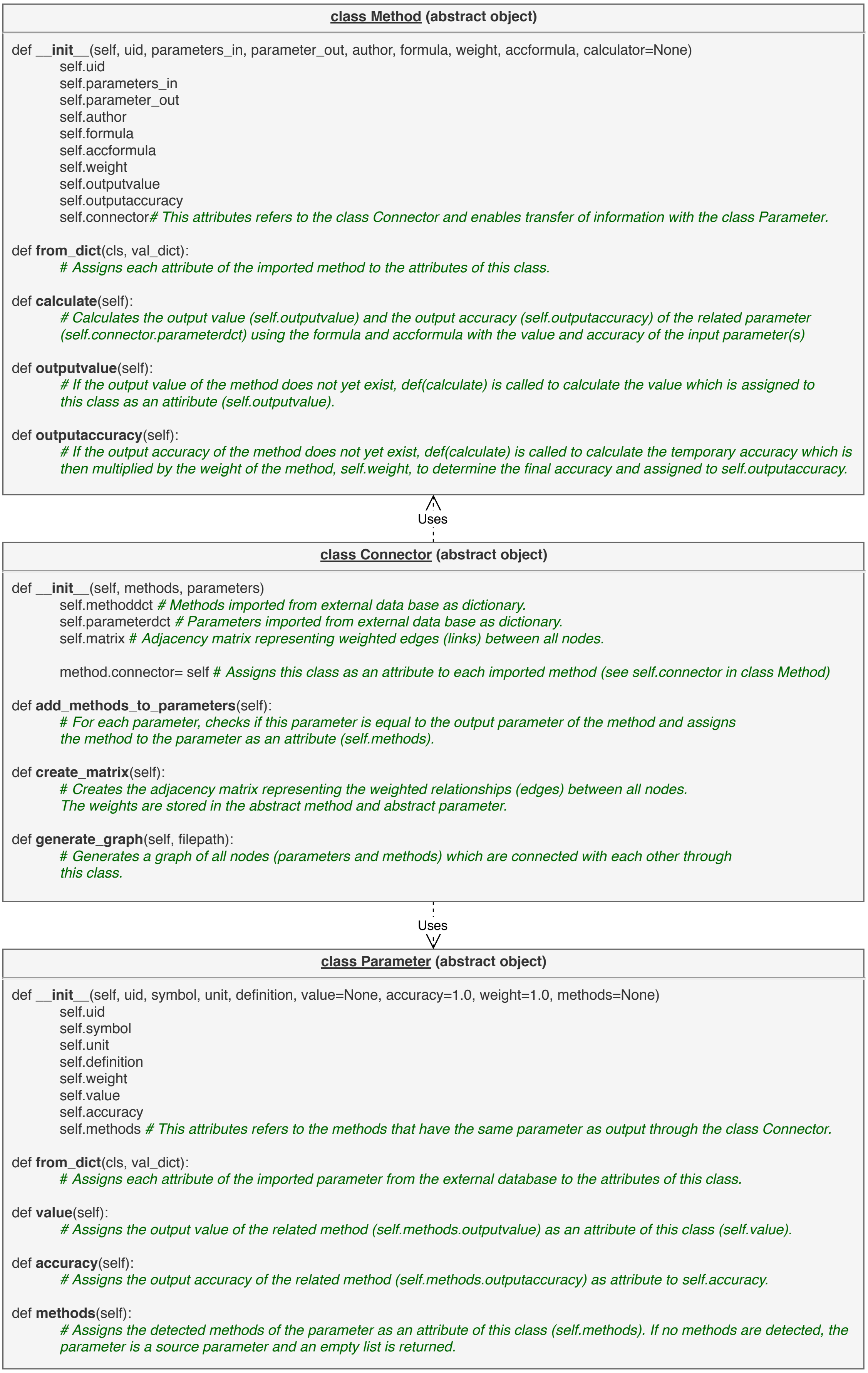


Figure 3.10: Conceptual overview of the parameter determination framework.



```
class Method (abstract object)

def __init__(self, uid, parameters_in, parameter_out, author, formula, weight, accformula, calculator=None)
        self.uid
        self.parameters_in
        self.parameter_out
        self.author
        self.formula
        self.accformula
        self.weight
        self.outputvalue
        self.outputaccuracy
        self.connector# This attributes refers to the class Connector and enables transfer of information with the class Parameter.

def from_dict(cls, val_dict):
        # Assigns each attribute of the imported method to the attributes of this class.

def calculate(self):
        # Calculates the output value (self.outputvalue) and the output accuracy (self.outputaccuracy) of the related parameter
        (self.connector.parameterdct) using the formula and accformula with the value and accuracy of the input parameter(s)

def outputvalue(self):
        # If the output value of the method does not yet exist, def(calculate) is called to calculate the value which is assigned to
        this class as an attiribute (self.outputvalue).

def outputaccuracy(self):
        # If the output accuracy of the method does not yet exist, def(calculate) is called to calculate the temporary accuracy which is
        then multiplied by the weight of the method, self.weight, to determine the final accuracy and assigned to self.outputaccuracy.
```

```
class Connector (abstract object)

def __init__(self, methods, parameters)
        self.methoddct # Methods imported from external data base as dictionary.
        self.parameterdct # Parameters imported from external data base as dictionary.
        self.matrix # Adjacency matrix representing weighted edges (links) between all nodes.

        method.connector= self # Assigns this class as an attribute to each imported method (see self.connector in class Method)

def add_methods_to_parameters(self):
        # For each parameter, checks if this parameter is equal to the output parameter of the method and assigns
        the method to the parameter as an attribute (self.methods).

def create_matrix(self):
        # Creates the adjacency matrix representing the weighted relationships (edges) between all nodes.
        The weights are stored in the abstract method and abstract parameter.

def generate_graph(self, filepath):
        # Generates a graph of all nodes (parameters and methods) which are connected with each other through
        this class.
```

```
class Parameter (abstract object)

def __init__(self, uid, symbol, unit, definition, value=None, accuracy=1.0, weight=1.0, methods=None)
        self.uid
        self.symbol
        self.unit
        self.definition
        self.weight
        self.value
        self.accuracy
        self.methods # This attributes refers to the methods that have the same parameter as output through the class Connector.

def from_dict(cls, val_dict):
        # Assigns each attribute of the imported parameter from the external database to the attributes of this class.

def value(self):
        # Assigns the output value of the related method (self.methods.outputvalue) as an attribute of this class (self.value).

def accuracy(self):
        # Assigns the output accuracy of the related method (self.methods.outputaccuracy) as attribute to self.accuracy.

def methods(self):
        # Assigns the detected methods of the parameter as an attribute of this class (self.methods). If no methods are detected, the
        parameter is a source parameter and an empty list is returned.
```

Figure 3.11: Architecture of the parameter determination system demonstrating how the abstract objects (i.e., Method, Parameter and Connector) are defined and connected with each other.

Chapter 4

Proof of Concept

This chapter presents a proof of concept for the viability of the parameter determination system, in which the graph is generated for the test case, presented in Section 4.1, and the geotechnical case, presented in Section 4.2. The test case is a simple example in which an external database of fictitious parameters and methods is used as input for the parameter determination system, to generate the resulting graph. The geotechnical case is a geotechnical example, using fictitious geotechnical parameters and methods, used as input for the system. The external database for the geotechnical case is established based on the review on the parameter determination in CPT data (Section 2.2), used to generate the resulting graph for the geotechnical case. Finally, a verification is performed for both modules, by comparing the system computed results to the hand computed results.

4.1 Test case: Determining parameter "e".

Before this system can be applied to a geotechnical problem, it had to be tested on a simplified case without using geotechnical terms. An external database was created consisting of all parameters and methods used by the system to generate paths for the resulting graph. The results for all parameters derived by the system, both intermediate and final results, are visualised on the graph. To verify the results, the system computed results were compared with hand calculated results.

4.1.1 External database

The external methods and parameters used as input for the parameter determination system, are given in Table 4.1 and Table 4.2, respectively. The methods and parameters considered in the test case are fictitious. The source parameters of the test case are: *parameter a* and *parameter b*. The derived parameters are: *parameter c*, *parameter d* and *parameter e*, of which the first two are intermediate parameters, since they are used the following method, and the latter is the destination parameter. The aim is to generate a graph showing all paths from the source parameters to the destination parameter, *parameter e*.

Table 4.1: External spreadsheet (CSV file) containing the methods of the test case, used as input for the parameter determination system.

uid	author	parameter_out	formula	accformula	parameters_in	weight
m1	c_method_1	c	$a + b$	$a * b$	a, b	0.6
m2	c_method_2	c	$a + 2 * b$	$a * b$	a, b	0.4
m3	d_method_1	d	$a + c$	$a * c$	a, c	0.7
m4	d_method_2	d	$a + 2 * c$	$a * c$	a, c	0.3
m5	e_method_1	e	$b + d$	$b * d$	b, d	0.6
m6	e_method_2	e	$b + 2 * d$	$b * d$	b, d	0.4

Table 4.2: External spreadsheet (CSV file) containing the parameters of the test case, used as input for the parameter determination system.

uid	symbol	unit	value	accuracy	definition
p1	*a*	-	5	0.8	*parameter a*
p2	*b*	-	4	0.7	*parameter b*
p3	*c*	-			*parameter c*
p4	*d*	-			*parameter d*
p5	*e*	-			*parameter e*

4.1.2 Generated graph

After computing the parameter determination system, each concrete object has derived all of its attributes as defined by the abstract object. A graph is visualised using the Python package *graphviz*. This package provides a Python interface for the Graphviz open source graph visualisation software (Gansner, 2011). The resulting graph of the test case, using the external input data (Table 4.1 and Table 4.2), is presented in Figure 4.1. The graph demonstrates that there are eight possible paths that lead to *parameter e*, resulting in eight possible outcomes for the final value and the final accuracy.

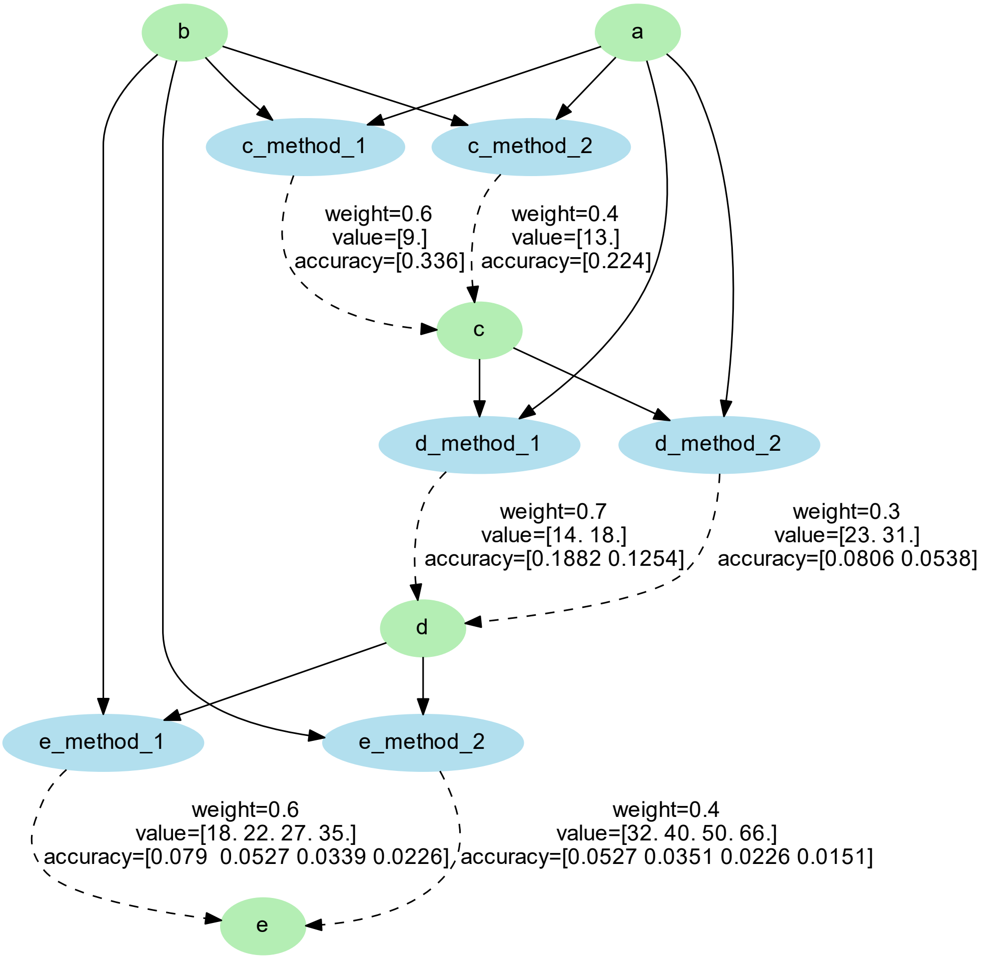


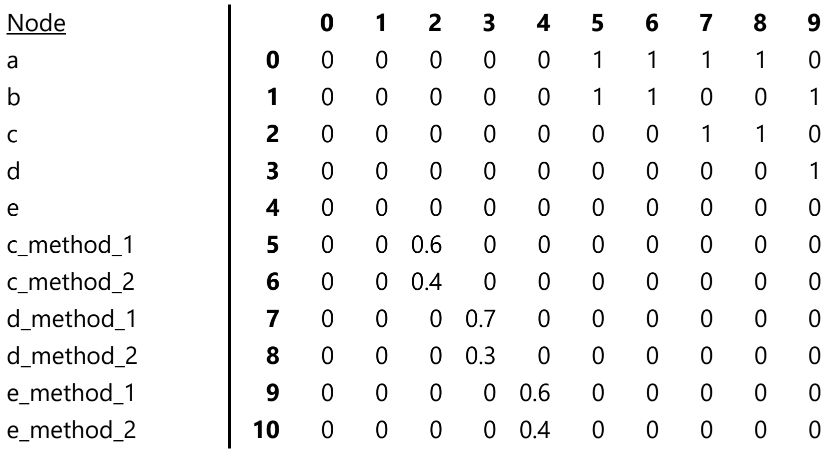
Figure 4.1: Generated graph of the test case after computing the parameter determination system, using the external methods and parameters from Table 4.1 and Table 4.2 as input data.

The computed adjacency matrix is simply a digital representation of the graph from Figure 4.1 and is shown in Figure 4.2. The matrix is an n x n matrix A in which the relation (weight) from each node (n_i) to every other node (n_j) is stored, where:

$$A_{ij} = 1, \text{ if there is an edge from a parameter node to a method node} \tag{4.1}$$

$$A_{ij} = \text{method's weight, if there is an edge from the method node to the parameter node} \tag{4.2}$$

$$A_{ij} = 0, \text{ if there is no edge between the pair of nodes} \tag{4.3}$$



Node		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
a	**0**	0	0	0	0	0	1	1	1	1	0
b	**1**	0	0	0	0	0	1	1	0	0	1
c	**2**	0	0	0	0	0	0	0	1	1	0
d	**3**	0	0	0	0	0	0	0	0	0	1
e	**4**	0	0	0	0	0	0	0	0	0	0
c_method_1	**5**	0	0	0.6	0	0	0	0	0	0	0
c_method_2	**6**	0	0	0.4	0	0	0	0	0	0	0
d_method_1	**7**	0	0	0	0.7	0	0	0	0	0	0
d_method_2	**8**	0	0	0	0.3	0	0	0	0	0	0
e_method_1	**9**	0	0	0	0	0.6	0	0	0	0	0
e_method_2	**10**	0	0	0	0	0.4	0	0	0	0	0

Figure 4.2: Adjacency matrix of the test case storing the relations (weights) of each node to every other node of the graph in Figure 4.1.

4.1.3 Verification of results

Calculations by the system were verified by comparing the results computed by the system with hand calculated results. Table 4.3 and Table 4.4 confirm that the hand calculated results match the system computed results. Notice that the parameters have been renamed to indicate what method is used to derive the parameter and what parameters are involved. For example, parameter d1a refers to d_method_1, using parameter c1 as input parameter, and parameter d2b refers to d_method_2, using parameter c2 as input parameter. The computation time to calculate all parameter outcomes in the system was 0.01 seconds.

Table 4.3: Comparison of hand calculated values, value1, with system computed values, value2, for the derived parameters of the test case. The difference is calculated as: value1 minus value2.

uid	author	parameter	formula	value1	value2	difference
p3	c_method_1	c1	a+b	9	9	0.0
p3	c_method_2	c2	a+2*b	13	13	0.0
p4	d_method_1	d1a	a+c1	14	14	0.0
p4	d_method_1	d1b	a+c2	18	18	0.0
p4	d_method_2	d2a	a+2*c1	23	23	0.0
p4	d_method_2	d2b	a+2*c2	31	31	0.0
p5	e_method_1	e1a	b+d1a	18	18	0.0
p5	e_method_1	e1b	b+d1b	22	22	0.0
p5	e_method_1	e1c	b+d2a	27	27	0.0
p5	e_method_1	e1d	b+d2b	35	35	0.0
p5	e_method_2	e2a	b+2*d1a	32	32	0.0
p5	e_method_2	e2b	b+2*d1b	40	40	0.0
p5	e_method_2	e2c	b+2*d2a	50	50	0.0
p5	e_method_2	e2d	b+2*d2b	66	66	0.0

Table 4.4: Comparison of hand calculated accuracies, accuracy1, with system computed accuracies, accuracy2, for the derived parameters of the test case. The difference is calculated as: accuracy1 minus accuracy2.

uid	author	weight	parameter	accformula	accuracy1	accuracy2	difference
p3	c_method_1	0.6	c1	a*b*weight	0.336	0.336	0.0
p3	c_method_2	0.4	c2	a*b*weight	0.224	0.224	0.0
p4	d_method_1	0.7	d1a	a*c1*weight	0.188	0.188	0.0
p4	d_method_1	0.7	d1b	a*c2*weight	0.125	0.125	0.0
p4	d_method_2	0.3	d2a	a*c1*weight	0.081	0.081	0.0
p4	d_method_2	0.3	d2b	a*c2*weight	0.054	0.054	0.0
p5	e_method_1	0.6	e1a	b*d1a*weight	0.079	0.079	0.0
p5	e_method_1	0.6	e1b	b*d1b*weight	0.053	0.053	0.0
p5	e_method_1	0.6	e1c	b*d2a*weight	0.034	0.034	0.0
p5	e_method_1	0.6	e1d	b*d2b*weight	0.023	0.023	0.0
p5	e_method_2	0.4	e2a	b*d1a*weight	0.053	0.053	0.0
p5	e_method_2	0.4	e2b	b*d1b*weight	0.035	0.035	0.0
p5	e_method_2	0.4	e2c	b*d2a*weight	0.023	0.023	0.0
p5	e_method_2	0.4	e2d	b*d2b*weight	0.015	0.015	0.0

4.2 Geotechnical case: Determining strength and stiffness parameters in sand

The parameter determination system was applied to a geotechnical example to generate a graph that explains how strength and stiffness parameters in sands can be derived using CPT-based empirical correlations. Prior to application of the system, an external database was established based on literature review (Chapter 2) consisting of the selected methods and parameters used for the analysis. Since the current parameter determination system is developed for one type of soil, it should be mentioned that soil profiling is regarded as a pre-processing part of the system and the soil behavioural type index (I_c) is not included as a parameter in the system. Although the current system focuses on one type of soil and one type of in situ test, the universality of the proposed framework allows the system to be extended to a wider range of soils and in situ tests. The system should be developed in such a way that it capable of adapting to an extended or modified external database and that the system can be extended in the future in terms of functionality.

4.2.1 External database

The external database of the parameters involved in the paths to derive the destination parameters for the geotechnical case is given in Table 4.5. The destination parameters are the strength and stiffness parameters that are of specific interest for the Hardening Soil model (Benz, 2007). The considered destination parameters are: the peak friction angle ϕ'_p, the maximum dilatancy angle ψ_p, the reference secant stiffness E_{50}^{ref}, the reference oedometric stiffness E_{oed}^{ref} and the reference unloading-reloading stiffness E_{ur}^{ref} in sand. The methods used for the analysis can be found in the complete external database in Figure B.1, Appendix B.

In Table 4.5, the *destination parameters* are denoted by uid p18 to p22, the *intermediate parameters* are denoted by uid p10 to p17 and the source parameters are denoted by uid p1 to p9. For the geotechnical case, the system assumes the source parameters have been obtained with 100% accuracy and were therefore assigned an accuracy of 1.00. In order to calculate the intermediate parameters and the destination parameters, the source parameters require input values. Values for the cone tip resistance q_c, sleeve friction f_s and measured pore water pressure u_2 were taken at depth z_{ref} is -20 m in a sand layer, based on example CPTU results from Figure A.1, Appendix A.

4.2.2 Generated graph

Based on the external methods and parameters from the external CSV files (Figure B.1, Appendix B), the parameter determination system was able to generate a graph visualising all possible paths between the set of source parameters and the set of destination parameters. The resulting graph is shown in Figure 4.3. The system computed results for the value and the accuracy of the derived parameters are listed in Table 4.6 and are also visible on the graph. For the purpose of clarity, the values on the graph are rounded to two decimal places, while calculations have been performed on unrounded values. Also note that the order of performed calculations may not coincide with the positioning order of the methods on the graph. However, the exact order of calculations can be verified from the output of the code, as shown in Appendix D. The formula used to calculate the value of each parameter is shown in the next section. The accuracy was calculated in the same manner as for the test case, i.e., by a dyadic product of the accuracy of the input parameters multiplied by the weight of the method. The weight of the method was assumed to be 0.60 in case of an empirical correlation and 1.00 in case of theoretical formula. The formulas to calculate the value and accuracy of the derived parameters can be found in Appendix B. The resulting adjacency matrix of the graph can be found in Appendix C.

Table 4.5: External spreadsheet (CSV file) of all parameters involved in the parameter determination system. The source parameters (uid: p1 - p9) have an an assumed input value and an assumed input accuracy. The value and accuracy of the other parameters (uid: p10 - p22) are left empty since they are derived by the system.

uid	symbol	unit	value	accuracy	definition
p1	qc	kN/m2	20000.00	1.00	cone tip resistance
p2	fs	kN/m2	200.00	1.00	sleeve resistance
p3	u2	kN/m2	250.00	1.00	measured pore water pressure
p4	a	kN/m2	0.80	1.00	cone area ratio
p5	gamma_water	kN/m3	9.81	1.00	water unit weight
p6	m	-	0.70	1.00	rate of stress dependency
p7	pa	kN/m2	100.00	1.00	atmospheric reference pressure
p8	phreatic_level	m	0.00	1.00	phreatic water level
p9	z_ref	m	-20.00	1.00	reference height
p10	qt	kN/m2			corrected cone tip resistance
p11	gamma	kN/m3			total soil unit weight
p12	sigw	kN/m2			pore water pressure
p13	sigv_tot	kN/m2			total overburden stress
p14	sigv_eff	kN/m2			effective overburden stress
p15	qt1	-			normalised cone tip resistance
p16	OCR	-			overconsolidation ratio
p17	Dr	-			relative density
p18	phiP	-			peak friction angle
p19	psiP	-			maximum dilatancy angle
p20	E50ref	kN/m2			reference secant stiffness
p21	Eoedref	kN/m2			reference oedometric stiffness
p22	Eurref	kN/m2			reference unloading-reloading stiffness

Table 4.6: System computed results for the value and accuracy of the derived parameters for the geotechnical case.

uid	author	parameter_out	unit	value	accuracy
p10	Robertson1986	qt	kN/m2	20050.00	0.60
p11	MaynePeuchen2012	gamma	kN/m3	19.98	0.60
p12	Theory_sigw	sigw	kN/m2	196.20	1.00
p13	Theory_sigv_tot	sigv_tot	kN/m2	399.63	1.00
p14	Theory_sigv_eff	sigv_eff	kN/m2	203.43	0.60
p15	KulhawyMayne1990c	qt1	-	140.57	0.22
p16	Mayne2009	OCR	-	1.64	0.13
p17	KulhawyMayne1990a	Dr	-	0.65	0.02
p17	Jamiolkowski1985	Dr	-	0.78	0.13
p17	LunneChristopherson1983	Dr	-	0.72	0.36
p18	KulhawyMayne1990b	phiP	-	41.23	0.13
p18	RobertsonCampanella1983	phiP	-	40.62	0.22
p18	Brinkgreve2010a	phiP	-	36.08	0.01
p18	Brinkgreve2010a	phiP	-	37.76	0.08
p18	Brinkgreve2010a	phiP	-	36.97	0.22
p19	Brinkgreve2010c	psiP	-	6.08	0.01
p19	Brinkgreve2010c	psiP	-	7.76	0.08
p19	Brinkgreve2010c	psiP	-	6.97	0.22
p19	Bolton1986	psiP	-	6.08	0.01
p19	Bolton1986	psiP	-	7.97	0.05
p19	Bolton1986	psiP	-	7.09	0.13
p20	Lengkeek2003	E50ref	kN/m2	38762.22	0.01
p20	Lengkeek2003	E50ref	kN/m2	46834.56	0.08
p20	Lengkeek2003	E50ref	kN/m2	43060.09	0.22
p21	Schanz1998	Eoedref	kN/m2	38762.22	0.01
p21	Schanz1998	Eoedref	kN/m2	46834.56	0.08
p21	Schanz1998	Eoedref	kN/m2	43060.09	0.22
p21	Vermeer2000	Eoedref	kN/m2	42067.12	0.36
p22	Brinkgreve2010b	Eurref	kN/m2	116286.66	0.01
p22	Brinkgreve2010b	Eurref	kN/m2	140503.69	0.08
p22	Brinkgreve2010b	Eurref	kN/m2	129180.28	0.22

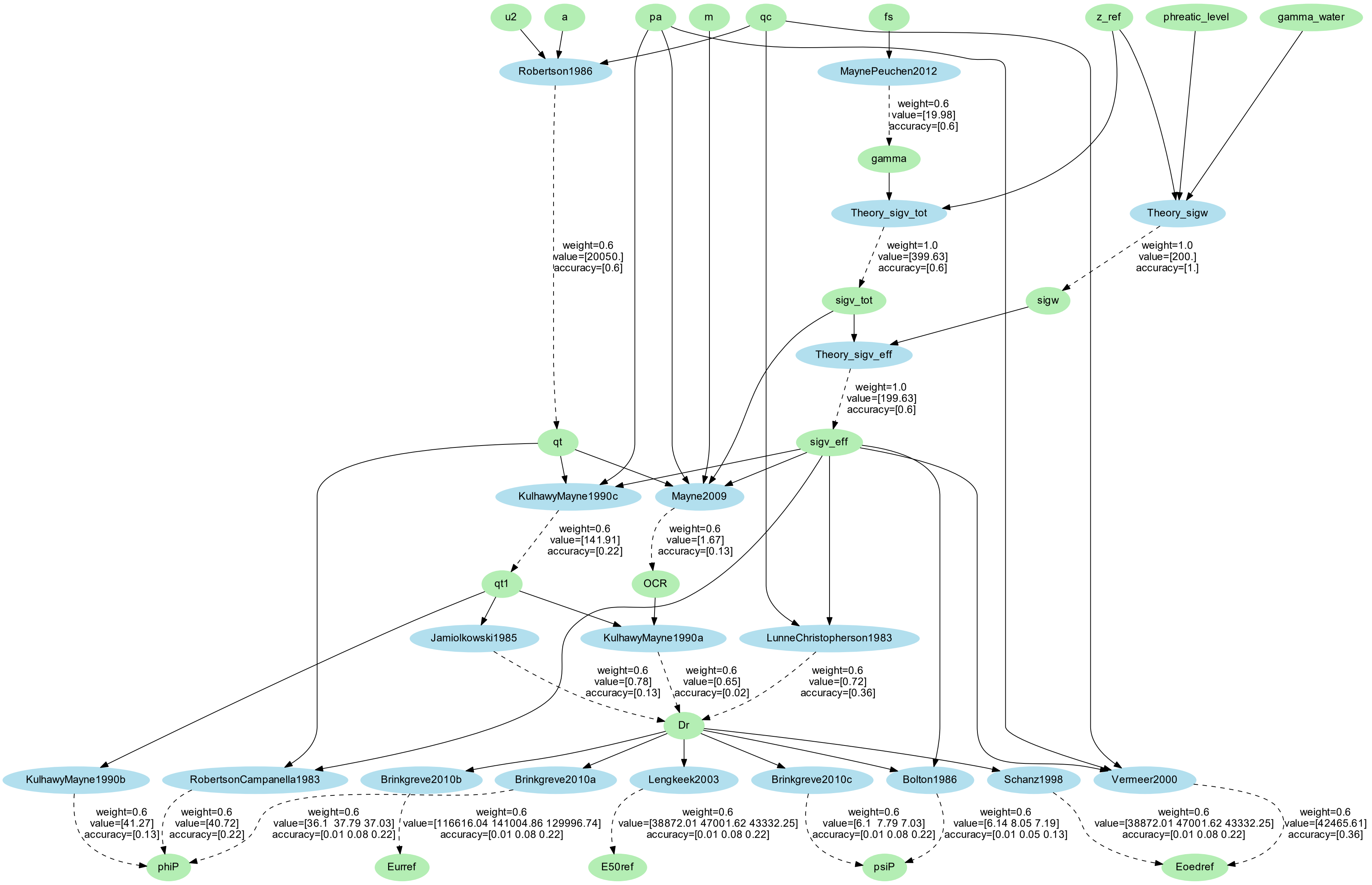


Figure 4.3: Generated graph of the geotechnical case after computing the parameter determination system, using the external methods and parameters from Figure B.1 (**??**) as input data.

4.2.3 Verification of results

The system computed results for the derived parameters, as shown on the graph in Figure 4.3, were compared with hand calculated results in order to verify the system. Although the methodology for calculating the value was already verified for the test case in Table 4.3, the formulas of the methods in the geotechnical case are more complicated and need to be tested. The results in Table 4.7 confirm that the system performs the calculations correctly, as the system computed values match the hand calculated values. Since the methodology for calculating the accuracy of a parameter was already verified for the test case (Table 4.4) and is performed in the exact same manner, it is unnecessary to verify the methodology again for the geotechnical case. The computation time of the parameter determination system to derive all parameter outcomes for the geotechnical case (uid: p10 - p22) was 0.03 seconds.

Table 4.7: Comparison of hand calculated values, value1, with system computed values, value2, for the derived parameters of the geotechnical case. The difference is calculated as: value1 minus value2.

uid	author	parameter_out	formula	value1	value2	difference
p10	Robertson1986	qt	qc+u2*(1-a)	20050.00	20050.00	0.00
p11	MaynePeuchen2012	gamma	26-14/(1+(0.5*log(fs+1))^2)	19.98	19.98	0.00
p12	Theory_sigw	sigw	gamma_water*(phreatic_level-z_ref)	196.20	196.20	0.00
p13	Theory_sigv_tot	sigv_tot	z_ref*gamma*-1	399.63	399.63	0.00
p14	Theory_sigv_eff	sigv_eff	sigv_tot-sigw	203.43	203.43	0.00
p15	KulhawyMayne1990c	qt1	(qt/pa)/(sigv_eff/pa)^0.5	140.57	140.57	0.00
p16	Mayne2009	OCR	((0.33*(qt-sigv_tot)^m)*(pa/100)^(1-m))/sigv_eff	1.64	1.64	0.00
p17	KulhawyMayne1990a	Dr1	(qt1/(305*OCR^0.2))^0.5	0.65	0.65	0.00
p17	Jamiolkowski1985	Dr2	68*(log(qt1)-1)/100	0.78	0.78	0.00
p17	LunneChristopherson1983	Dr3	(1/2.91)*ln(qc/(60*sigv_eff^0.7))	0.72	0.72	0.00
p18	KulhawyMayne1990b	phiP1	17.6+11.0*log(qt1)	41.23	41.23	0.00
p18	RobertsonCampanella1983	phiP2	degrees(atan(0.10+0.38*log(qt/sigv_eff)))	40.62	40.62	0.00
p18	Brinkgreve2010a	phiP3a	28+12.5*Dr1	36.08	36.08	0.00
p18	Brinkgreve2010a	phiP3b	28+12.5*Dr2	37.76	37.76	0.00
p18	Brinkgreve2010a	phiP3c	28+12.5*Dr3	36.97	36.97	0.00
p19	Brinkgreve2010c	psiP4a	2*-1+12.5*Dr1	6.08	6.08	0.00
p19	Brinkgreve2010c	psiP4b	2*-1+12.5*Dr2	7.76	7.76	0.00
p19	Brinkgreve2010c	psiP4c	2*-1+12.5*Dr3	6.97	6.97	0.00
p19	Bolton1986	psiP5a	3*(Dr1*(10-ln(sigv_eff))-1)	6.08	6.08	0.00
p19	Bolton1986	psiP5b	3*(Dr2*(10-ln(sigv_eff))-1)	7.97	7.97	0.00
p19	Bolton1986	psiP5c	3*(Dr3*(10-ln(sigv_eff))-1)	7.09	7.09	0.00
p20	Lengkeek2003	E50ref1a	60000*Dr1	38762.22	38762.22	0.00
p20	Lengkeek2003	E50ref1b	60000*Dr2	46834.56	46834.56	0.00
p20	Lengkeek2003	E50ref1c	60000*Dr3	43060.09	43060.09	0.00
p21	Schanz1998	Eoedref1a	60000*Dr1	38762.22	38762.22	0.00
p21	Schanz1998	Eoedref1b	60000*Dr2	46834.56	46834.56	0.00
p21	Schanz1998	Eoedref1c	60000*Dr3	43060.09	43060.09	0.00
p21	Vermeer2000	Eoedref2	3*qc*(pa/sigv_eff)^0.5	42067.12	42067.12	0.00
p22	Brinkgreve2010b	Eurref1a	180000*Dr1	116286.66	116286.66	0.00
p22	Brinkgreve2010b	Eurref1b	180000*Dr2	140503.69	140503.69	0.00
p22	Brinkgreve2010b	Eurref1c	180000*Dr3	129180.28	129180.28	0.00

4.3 Summary

This chapter presented a proof of concept for developing an automated parameter determination system while ensuring transparency and adaptability. Before applying the system on geotechnical parameters, the system was tested on a small test case with fictitious parameters and methods. Two external spreadsheets were created, one containing the methods and one containing the parameters used by the system. The attributes of the objects in the external databases specified according to the definition of the abstract objects in the system. After correctly establishing the external database, the system was able to automatically visualise the generated graph together with the performed calculations for the derived parameters. The system proved to be transparent since it allows verification of the entire parameter determination process using the graph and the output of the main code. Finally, two other external databases were created for the geotechnical case: one containing the methods (theoretical and empirical formulas) and one containing the parameters (geotechnical), involved in the system. The system proved to be adaptable since no modifications were made to the system in order to generate the graph. The graph for the geotechnical case could be generated by simply replacing the external database. In other words, all changes in a generated graph are caused only by changes in the external database that is used by the system. Furthermore, the results computed by the system were verified by comparing them with hand calculated results and verified that the system performed the calculations as expected.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

This study presented a proof of concept for a parameter determination system, in which constitutive model parameters can be derived from in situ tests while ensuring adaptability and transparency. In situ testing allows soils to be characterised with minimum disturbance. The cone penetration test (CPT) in particular, has become a popular tool since it is an efficient and economical method for routine site characterisation, soil profiling and estimation of constitutive properties of soil. However, engineers are confronted with a large amount of empiricism when interpreting in situ test results. In addition, the rise in complexity of constitutive models with an associated increase in number of parameters, have made soil interpretation from experimental data ever more challenging. The aim of this study was: "*Elaborating a transparent and adaptable parameter determination system to increase the reliability of parameters derived from in situ tests based on graphs*". To meet the aim of this thesis, the answers to the six research questions that were stated in the beginning of this thesis are presented.

- *Generating paths.* How can a system be developed to generate valid paths between geotechnical parameters in a graph?

  The key to generating paths lies in the definition of the objects in the system. By defining an abstract object for a method and a parameter (i.e., nodes), and by defining an additional abstract object, called the "connector", a connection (i.e., an edge) between these nodes can be made and in this way valid paths are generated between geotechnical parameters in the graph.

- *Performing calculations.* How can the system be developed to allow calculations performed for all parameters involved in a path?

  After connecting a related pair of nodes, i.e., a method is connected with its related parameters, the value and the accuracy of each parameter involved in a path were calculated by implementing the formulas for the value and the accuracy as specified by the user in the external database which is used as input for the system. The system performs the calculations for all parameters involved in a path, until all parameters have been calculated.

- *Treating multiple parameter outcomes.* How does the system cope with multiple parameter outcomes?

  The system calculates and presents multiple values including outliers for a parameter. It is expected that further interpretation is left open to the user of the system: the geotechnical engineer. Users of the system should apply their expertise to interpret the results carefully.

- *Accounting for uncertainties.* How are uncertainties in geotechnical parameters accounted for in the system?

  Any uncertainty associated with a parameter was assumed to be related with the uncertainty of the empirical correlation. The accuracy of a derived parameter was determined as a function of the accuracy (weight) of the considered formula and the product of the accuracy of the input parameters. An arbitrary weight below 1.0, weight = 0.6, was assigned to all empirical correlations, since they are an approximation. Theoretical relationships, on the other hand, were assigned a weight of 1.0, since they are based on physics. The weighting procedure for the empirical correlations depends on the limitations and validity of the correlation which requires extensive study and should be researched in the future.

- *Ensuring adaptability.* How can the system ensure adaptability for both the user and the future developer of the system?

  The system ensures adaptability for the *user* by separating abstraction and implementation within the system. Two graphs were generated: one for the test case and one for the geotechnical case. The graph for the test case was generated by using the external database of the test case as input. To generate the graph for the geotechnical case, the external database of the test case was simply replaced by the external database of the geotechnical case and no internal changes were made to the system. This makes it possible for users to incorporate their expertise into the system externally, without having to make modifications to the system. It is emphasised that the user is expected to have the expert knowledge to establish such an external database without any inconsistencies. Furthermore, the system ensures adaptability for the *developer* by structuring the system in a modular way. This allows the developer to add functionalities to the system. For example, implementation of the validity range for an empirical correlation can be realised by extending the definition of an abstract method with an additional attribute.

- *Ensuring transparency.* How can the system ensure transparency such that the user is able to verify the entire parameter determination procedure?

  The output of the system is a graph that visualises all paths leading to the destination parameters, presenting all parameters and methods involved in each path including the intermediate results. Since the user has control over the knowledge used by the system and the system is able to explain how this external knowledge is applied to arrive at a solution, the user is able to verify the entire parameter determination process which confirms the transparency of the system.

5.2 Recommendations

This study provided a proof of concept for developing a transparent and adaptable parameter determination system. It is regarded as a first step towards an automated parameter selection system which requires further improvements and developments in the future, as described by the following points.

- Further improvement is required in the treatment of multiple parameter outcomes. Multiple empirical correlations are often valid when determining a parameter, resulting in multiple parameter outcomes for the same parameter. The current system leaves the interpretation of the final value to the user. In order to mitigate human factors, it may be desired to incorporate some mathematical procedure to convert these values into one value.

- The weighting procedure of the empirical correlations requires further improvement. With regards to calculating the accuracy of a parameter, the procedure shows that the accuracy of a parameter decreases when more empirical correlations are involved. A suggestion is to incorporate the coefficient of variation or standard deviation of empirical correlations into the system.

- Since empirical correlations are generally valid within a specified range, it is recommended to incorporate specific validity ranges for empirical correlations in the system. This should ensure that results are within expectations. The current system does not incorporate any limitations of the empirical correlations. This could be accomplished by extending the definition of an abstract method with a property for the validity of the formula. The user of the system can specify the validity range, e.g., a lower and upper cut-off threshold, of an empirical correlation in the external data base, which can be incorporated by the system during execution.

- When extending the system for a wider range of soils, the system should be modified in such a way that the soil behaviour type index (I_c or I_{sbt}) is incorporated into the weighting procedure of the empirical correlations. As a result, the soil behaviour type index may contribute to the validity of an empirical correlation. A possible drawback with determining the weight of an empirical correlation based on the soil behavioural type index, is that the soil behavourial type index may have too much influence in the weighting procedure of the empirical correlations. Therefore, a suggestion would be to develop the system for discrete groups of soil, e.g., fine-grained and coarse-grained soil. This will result in a separate graph for sand and for clay. Parameters in the graph for sand and clay will carry a soil behavioural type index as a property for sand and clay, respectively. Hence, the system produces a final parameter outcome for sand and for clay. If during the pre-processing part, prior to execution of the system, the considered soil type was a mixed-soil type (e.g., a silt) as determined by the Ic-value, the final parameter calculated by the system may be re-calculated as a weighted average based on the Ic-values of sand and clay.

- Finally, a validation must be performed in order to test the robustness of the system. A first suggestion is to perform unit-tests to check if segments of code perform as expected. The abstract objects of the system (i.e., the method, the parameter and the connector) as well as the external information used by the system (e.g., equations) should be tested since they can be complex equations with built-in conditions for their validity and limitations. Furthermore, it is recommended to build restrictions into the system to prevent the formation of closed loops due to circular referencing. For example, if an empirical correlation for an intermediate parameter in a path involves an input parameter which is also the destination parameter of that path, a loop is formed since this intermediate parameter requires the destination parameter as input; however, the destination parameter also requires the intermediate parameter as input. Circular referencing is caused by inconsistencies in the external database established by the user and can currently only be solved by correcting this external database. Hence, the responsibility of correctly incorporating external expertise into the system lies in the user: the geotechnical expert.

Appendix A

Example CPTU results

Example CPTU results on which the values for the source parameters (qc, u2, fs and z_ref) of the final module were based are shown in Figure A.1.

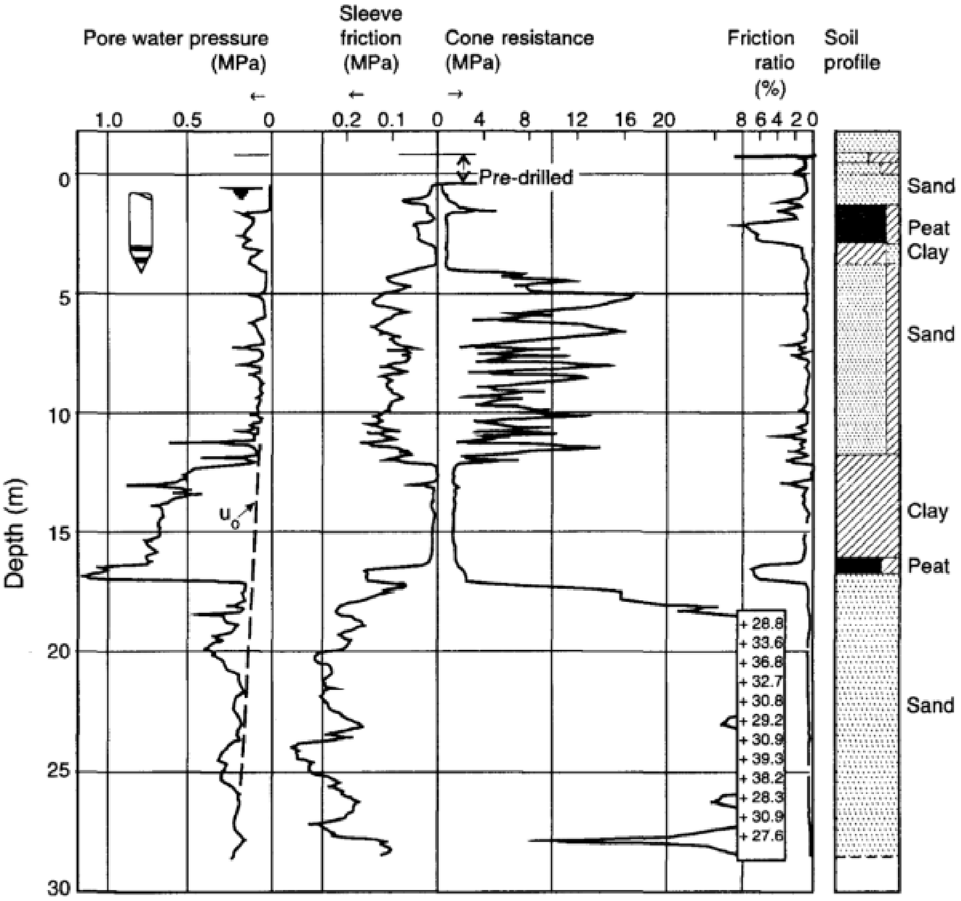
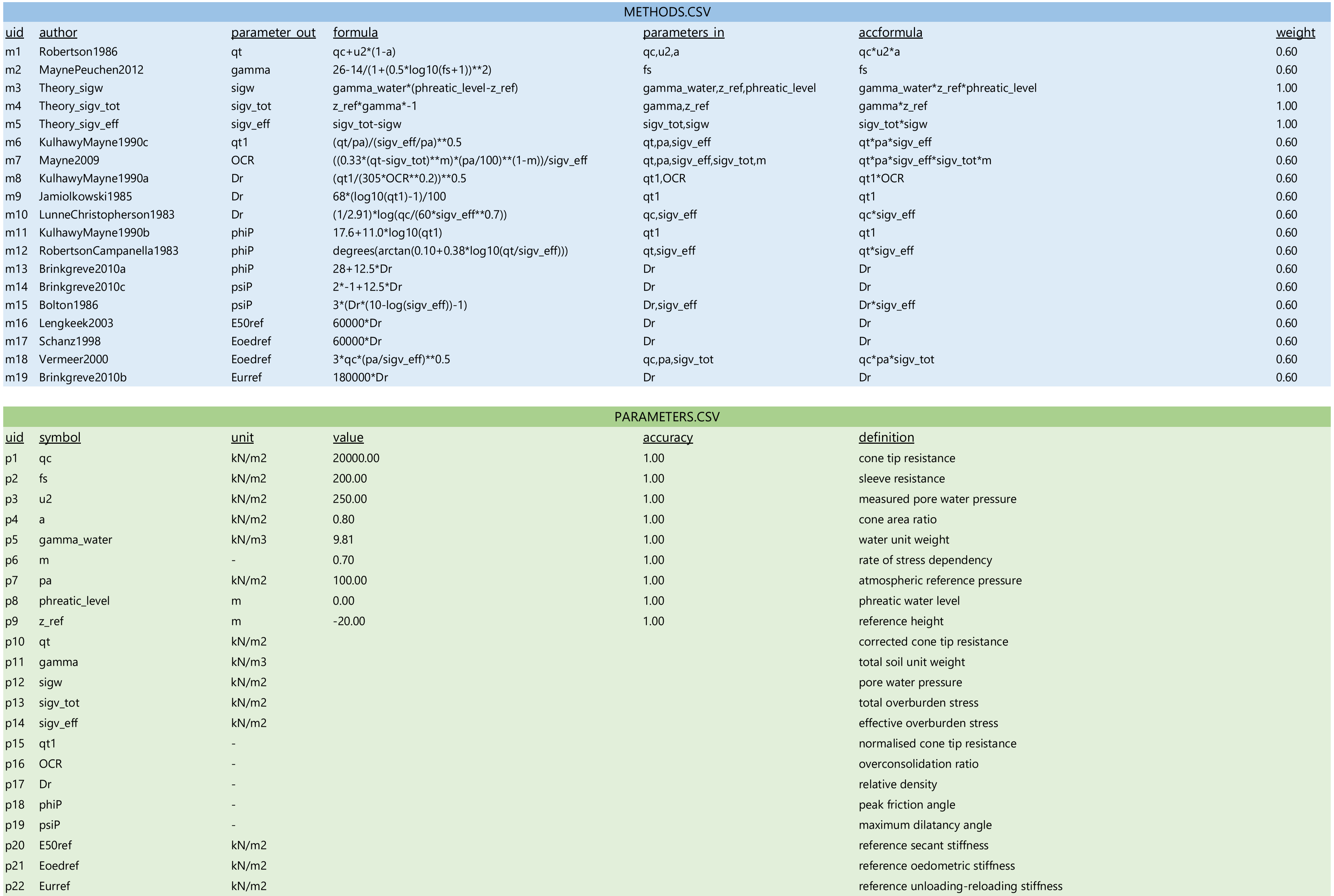


Figure A.1: Example CPTU results from Zuidberg et al. (1982) on which the values for the source parameters (qc, u2, fs and z_ref) were based for the final module (Lunne et al., 1997).

Appendix B

External Database

The external database of the methods (methods.csv) and parameters (parameters.csv) used as input data to generate the graph for the final module is given in Figure B.1.



METHODS.CSV

uid	author	parameter_out	formula	parameters_in	accformula	weight
m1	Robertson1986	qt	qc+u2*(1-a)	qc,u2,a	qc*u2*a	0.60
m2	MaynePeuchen2012	gamma	26-14/(1+(0.5*log10(fs+1))**2)	fs	fs	0.60
m3	Theory_sigw	sigw	gamma_water*(phreatic_level-z_ref)	gamma_water,z_ref,phreatic_level	gamma_water*z_ref*phreatic_level	1.00
m4	Theory_sigv_tot	sigv_tot	z_ref*gamma*-1	gamma,z_ref	gamma*z_ref	1.00
m5	Theory_sigv_eff	sigv_eff	sigv_tot-sigw	sigv_tot,sigw	sigv_tot*sigw	1.00
m6	KulhawyMayne1990c	qt1	(qt/pa)/(sigv_eff/pa)**0.5	qt,pa,sigv_eff	qt*pa*sigv_eff	0.60
m7	Mayne2009	OCR	((0.33*(qt-sigv_tot)**m)*(pa/100)**(1-m))/sigv_eff	qt,pa,sigv_eff,sigv_tot,m	qt*pa*sigv_eff*sigv_tot*m	0.60
m8	KulhawyMayne1990a	Dr	(qt1/(305*OCR**0.2))**0.5	qt1,OCR	qt1*OCR	0.60
m9	Jamiolkowski1985	Dr	68*(log10(qt1)-1)/100	qt1	qt1	0.60
m10	LunneChristopherson1983	Dr	(1/2.91)*log(qc/(60*sigv_eff**0.7))	qc,sigv_eff	qc*sigv_eff	0.60
m11	KulhawyMayne1990b	phiP	17.6+11.0*log10(qt1)	qt1	qt1	0.60
m12	RobertsonCampanella1983	phiP	degrees(arctan(0.10+0.38*log10(qt/sigv_eff)))	qt,sigv_eff	qt*sigv_eff	0.60
m13	Brinkgreve2010a	phiP	28+12.5*Dr	Dr	Dr	0.60
m14	Brinkgreve2010c	psiP	2*-1+12.5*Dr	Dr	Dr	0.60
m15	Bolton1986	psiP	3*(Dr*(10-log(sigv_eff))-1)	Dr,sigv_eff	Dr*sigv_eff	0.60
m16	Lengkeek2003	E50ref	60000*Dr	Dr	Dr	0.60
m17	Schanz1998	Eoedref	60000*Dr	Dr	Dr	0.60
m18	Vermeer2000	Eoedref	3*qc*(pa/sigv_eff)**0.5	qc,pa,sigv_tot	qc*pa*sigv_tot	0.60
m19	Brinkgreve2010b	Eurref	180000*Dr	Dr	Dr	0.60

PARAMETERS.CSV

uid	symbol	unit	value	accuracy	definition
p1	qc	kN/m2	20000.00	1.00	cone tip resistance
p2	fs	kN/m2	200.00	1.00	sleeve resistance
p3	u2	kN/m2	250.00	1.00	measured pore water pressure
p4	a	kN/m2	0.80	1.00	cone area ratio
p5	gamma_water	kN/m3	9.81	1.00	water unit weight
p6	m	-	0.70	1.00	rate of stress dependency
p7	pa	kN/m2	100.00	1.00	atmospheric reference pressure
p8	phreatic_level	m	0.00	1.00	phreatic water level
p9	z_ref	m	-20.00	1.00	reference height
p10	qt	kN/m2			corrected cone tip resistance
p11	gamma	kN/m3			total soil unit weight
p12	sigw	kN/m2			pore water pressure
p13	sigv_tot	kN/m2			total overburden stress
p14	sigv_eff	kN/m2			effective overburden stress
p15	qt1	-			normalised cone tip resistance
p16	OCR	-			overconsolidation ratio
p17	Dr	-			relative density
p18	phiP	-			peak friction angle
p19	psiP	-			maximum dilatancy angle
p20	E50ref	kN/m2			reference secant stiffness
p21	Eoedref	kN/m2			reference oedometric stiffness
p22	Eurref	kN/m2			reference unloading-reloading stiffness

Figure B.1: External database of methods and parameters used as input for final module to generate the graph for the final module.

Appendix C

Adjacency Matrix

The computed adjacency matrix of the graph for the final module in Figure 4.3 is shown in Figure C.1.

Node		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
qc	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
fs	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
u2	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a	**3**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
gamma_water	**4**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	**5**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
pa	**6**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0
phreatic_level	**7**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
z_ref	**8**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
qt	**9**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
gamma	**10**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sigw	**11**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sigv_tot	**12**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
sigv_eff	**13**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0	0	0
qt1	**14**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
OCR	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Dr	**16**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
phiP	**17**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
psiP	**18**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E50ref	**19**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eoedref	**20**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eurref	**21**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Robertson1986	**22**	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MaynePeuchen2012	**23**	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Theory_sigw	**24**	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Theory_sigv_tot	**25**	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Theory_sigv_eff	**26**	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KulhawyMayne1990c	**27**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mayne2009	**28**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KulhawyMayne1990a	**29**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jamiolkowski1985	**30**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LunneChristopherson1983	**31**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KulhawyMayne1990b	**32**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RobertsonCampanella1983	**33**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brinkgreve2010a	**34**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brinkgreve2010c	**35**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bolton1986	**36**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lengkeek2003	**37**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Schanz1998	**38**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LunneRobertsonPowell1997	**39**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Brinkgreve2010b	**40**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

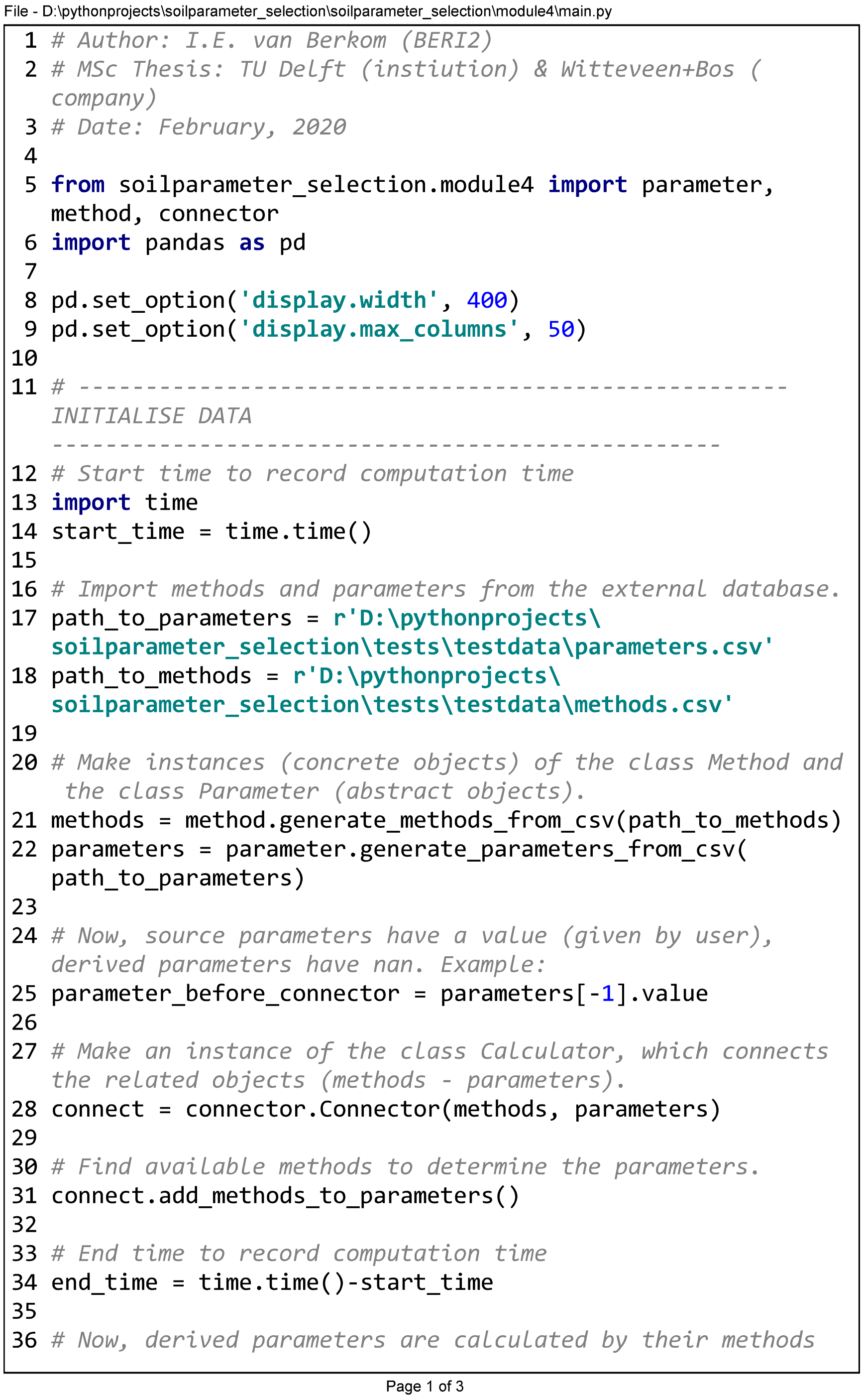
Figure C.1: Adjacency matrix of the final module storing the relations of each node to every other node of the graph in Figure 4.3.

Appendix D

Python Code

The main script, *main.py*, for the final module is given in Figure D.1, Figure D.2 and Figure D.3.

The output of *main.py* showing the calculated parameter values (unrounded) is given in Figure D.4, Figure D.5, Figure D.6, Figure D.7 and Figure D.8.

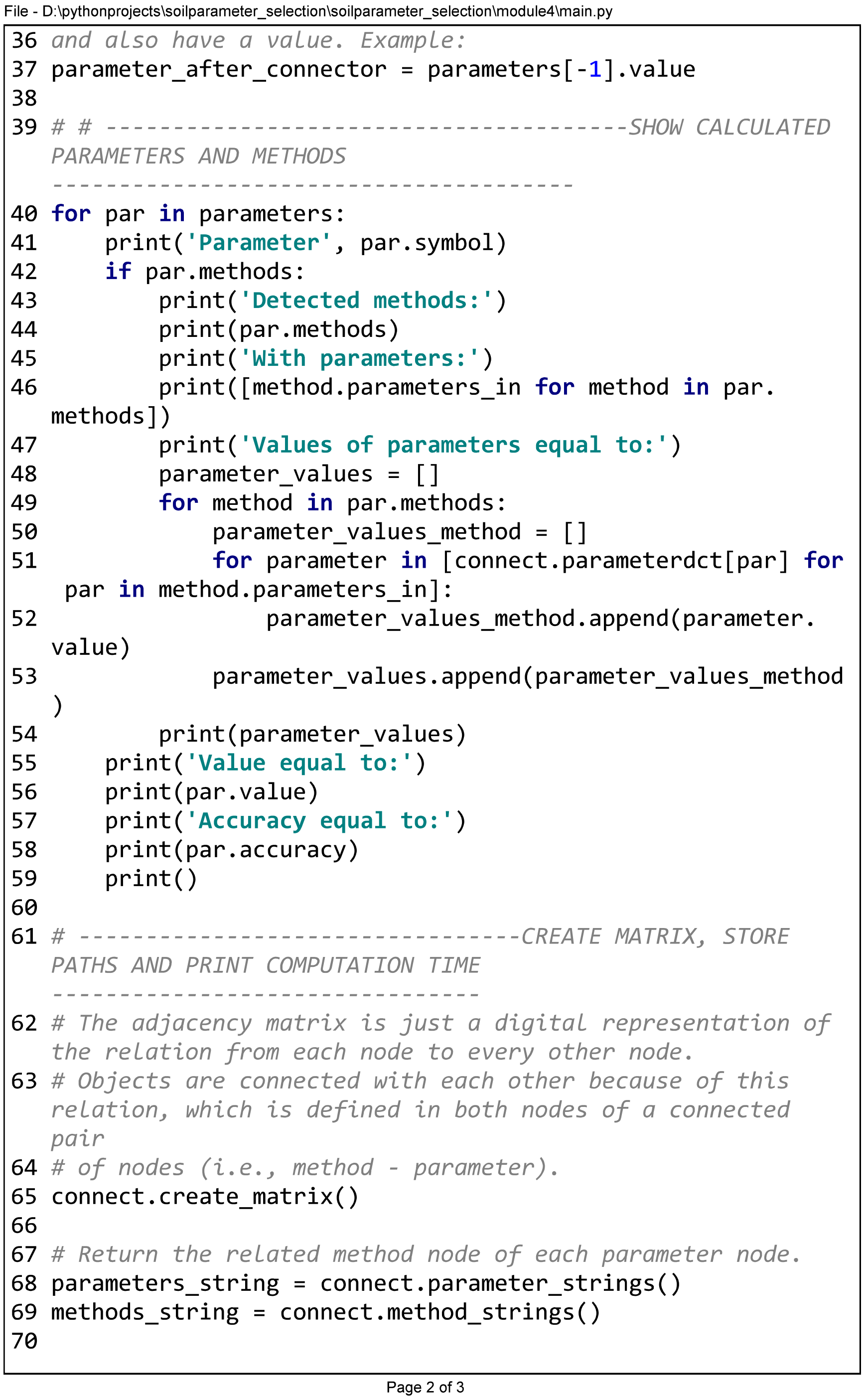


File - D:\pythonprojects\soilparameter_selection\soilparameter_selection\module4\main.py

```python
# Author: I.E. van Berkom (BERI2)
# MSc Thesis: TU Delft (instiution) & Witteveen+Bos (company)
# Date: February, 2020

from soilparameter_selection.module4 import parameter, method, connector
import pandas as pd

pd.set_option('display.width', 400)
pd.set_option('display.max_columns', 50)

# ----------------------------------------------------- INITIALISE DATA --------------------------------------------------
# Start time to record computation time
import time
start_time = time.time()

# Import methods and parameters from the external database.
path_to_parameters = r'D:\pythonprojects\soilparameter_selection\tests\testdata\parameters.csv'
path_to_methods = r'D:\pythonprojects\soilparameter_selection\tests\testdata\methods.csv'

# Make instances (concrete objects) of the class Method and the class Parameter (abstract objects).
methods = method.generate_methods_from_csv(path_to_methods)
parameters = parameter.generate_parameters_from_csv(path_to_parameters)

# Now, source parameters have a value (given by user), derived parameters have nan. Example:
parameter_before_connector = parameters[-1].value

# Make an instance of the class Calculator, which connects the related objects (methods - parameters).
connect = connector.Connector(methods, parameters)

# Find available methods to determine the parameters.
connect.add_methods_to_parameters()

# End time to record computation time
end_time = time.time()-start_time

# Now, derived parameters are calculated by their methods
```

Page 1 of 3

Figure D.1: Main.py (Page 1 of 3).

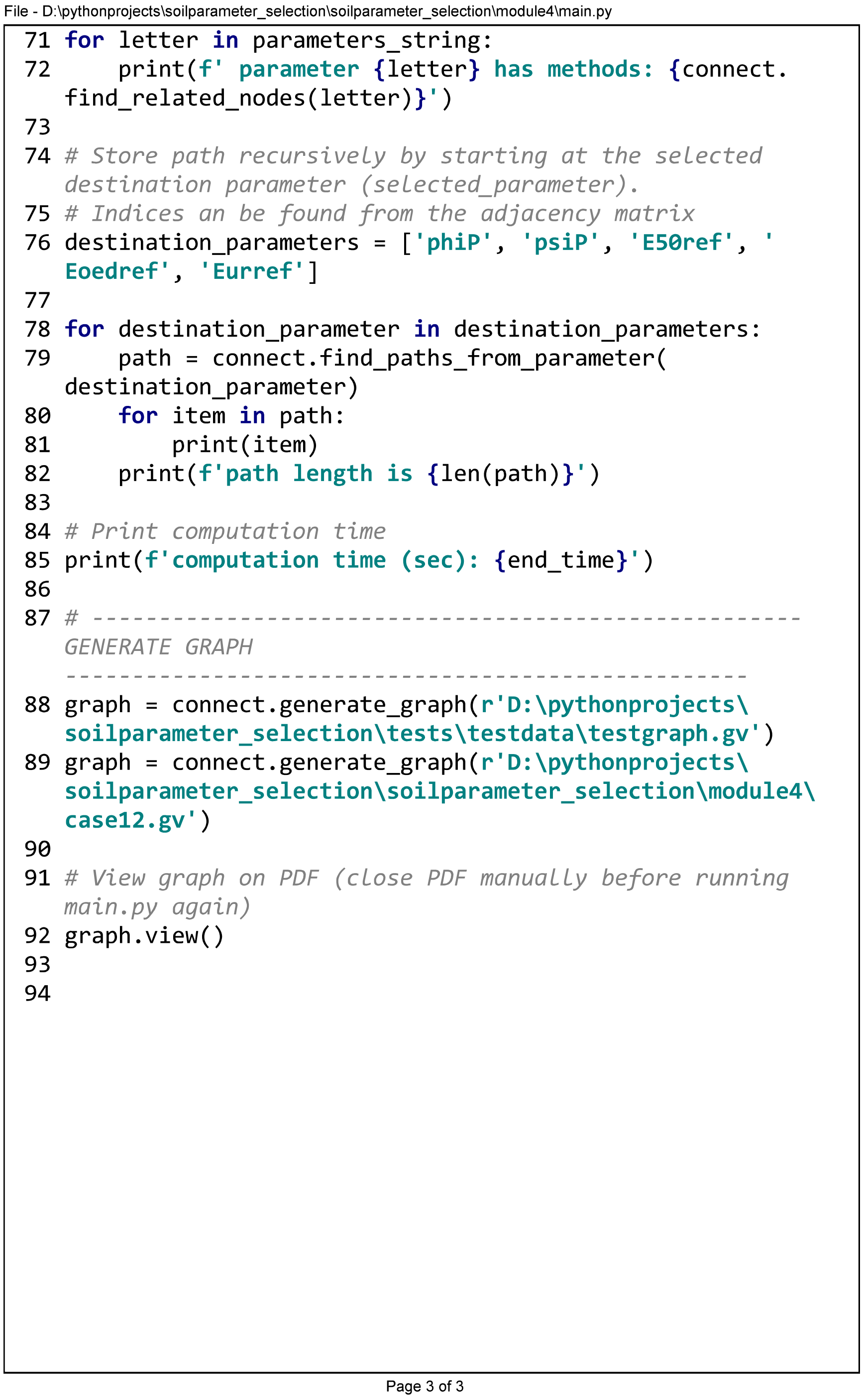


File - D:\pythonprojects\soilparameter_selection\soilparameter_selection\module4\main.py

```python
and also have a value. Example:
parameter_after_connector = parameters[-1].value

# # ---------------------------------------SHOW CALCULATED PARAMETERS AND METHODS ---------------------------------------
for par in parameters:
    print('Parameter', par.symbol)
    if par.methods:
        print('Detected methods:')
        print(par.methods)
        print('With parameters:')
        print([method.parameters_in for method in par.methods])
        print('Values of parameters equal to:')
        parameter_values = []
        for method in par.methods:
            parameter_values_method = []
            for parameter in [connect.parameterdct[par] for par in method.parameters_in]:
                parameter_values_method.append(parameter.value)
            parameter_values.append(parameter_values_method)
        print(parameter_values)
    print('Value equal to:')
    print(par.value)
    print('Accuracy equal to:')
    print(par.accuracy)
    print()

# ---------------------------------CREATE MATRIX, STORE PATHS AND PRINT COMPUTATION TIME --------------------------------
# The adjacency matrix is just a digital representation of the relation from each node to every other node.
# Objects are connected with each other because of this relation, which is defined in both nodes of a connected pair
# of nodes (i.e., method - parameter).
connect.create_matrix()

# Return the related method node of each parameter node.
parameters_string = connect.parameter_strings()
methods_string = connect.method_strings()

```

Page 2 of 3

Figure D.2: Main.py (Page 2 of 3).

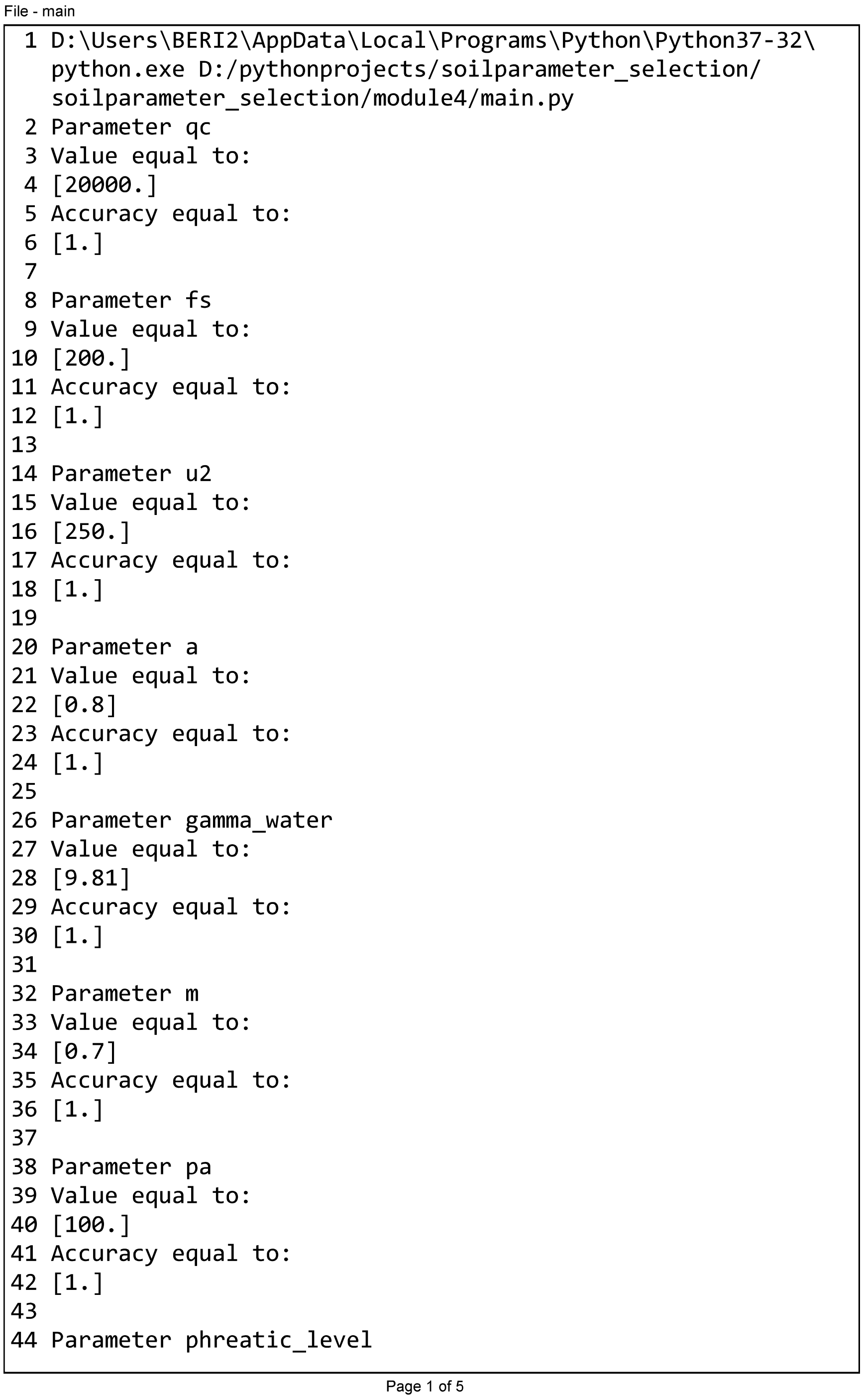


File - D:\pythonprojects\soilparameter_selection\soilparameter_selection\module4\main.py

```python
for letter in parameters_string:
    print(f' parameter {letter} has methods: {connect.find_related_nodes(letter)}')

# Store path recursively by starting at the selected destination parameter (selected_parameter).
# Indices an be found from the adjacency matrix
destination_parameters = ['phiP', 'psiP', 'E50ref', 'Eoedref', 'Eurref']

for destination_parameter in destination_parameters:
    path = connect.find_paths_from_parameter(destination_parameter)
    for item in path:
        print(item)
    print(f'path length is {len(path)}')

# Print computation time
print(f'computation time (sec): {end_time}')

# ----------------------------------------------------- GENERATE GRAPH ---------------------------------------------------
graph = connect.generate_graph(r'D:\pythonprojects\soilparameter_selection\tests\testdata\testgraph.gv')
graph = connect.generate_graph(r'D:\pythonprojects\soilparameter_selection\soilparameter_selection\module4\case12.gv')

# View graph on PDF (close PDF manually before running main.py again)
graph.view()


```

Page 3 of 3

Figure D.3: Main.py (Page 3 of 3).

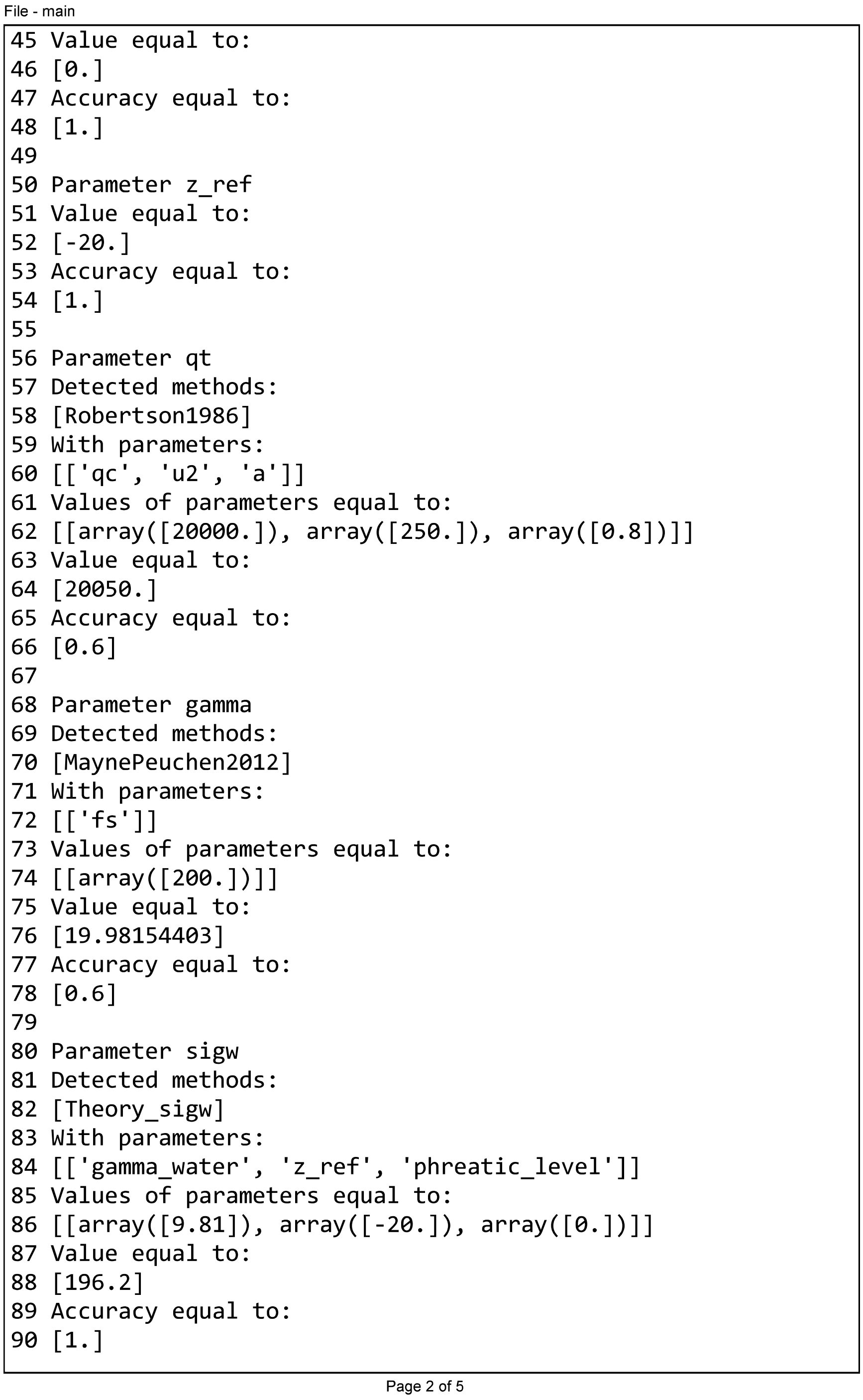


File - main

```
D:\Users\BERI2\AppData\Local\Programs\Python\Python37-32\
python.exe D:/pythonprojects/soilparameter_selection/
soilparameter_selection/module4/main.py
Parameter qc
Value equal to:
[20000.]
Accuracy equal to:
[1.]

Parameter fs
Value equal to:
[200.]
Accuracy equal to:
[1.]

Parameter u2
Value equal to:
[250.]
Accuracy equal to:
[1.]

Parameter a
Value equal to:
[0.8]
Accuracy equal to:
[1.]

Parameter gamma_water
Value equal to:
[9.81]
Accuracy equal to:
[1.]

Parameter m
Value equal to:
[0.7]
Accuracy equal to:
[1.]

Parameter pa
Value equal to:
[100.]
Accuracy equal to:
[1.]

Parameter phreatic_level
```

Page 1 of 5

Figure D.4: Output of main.py (Page 1 of 5).

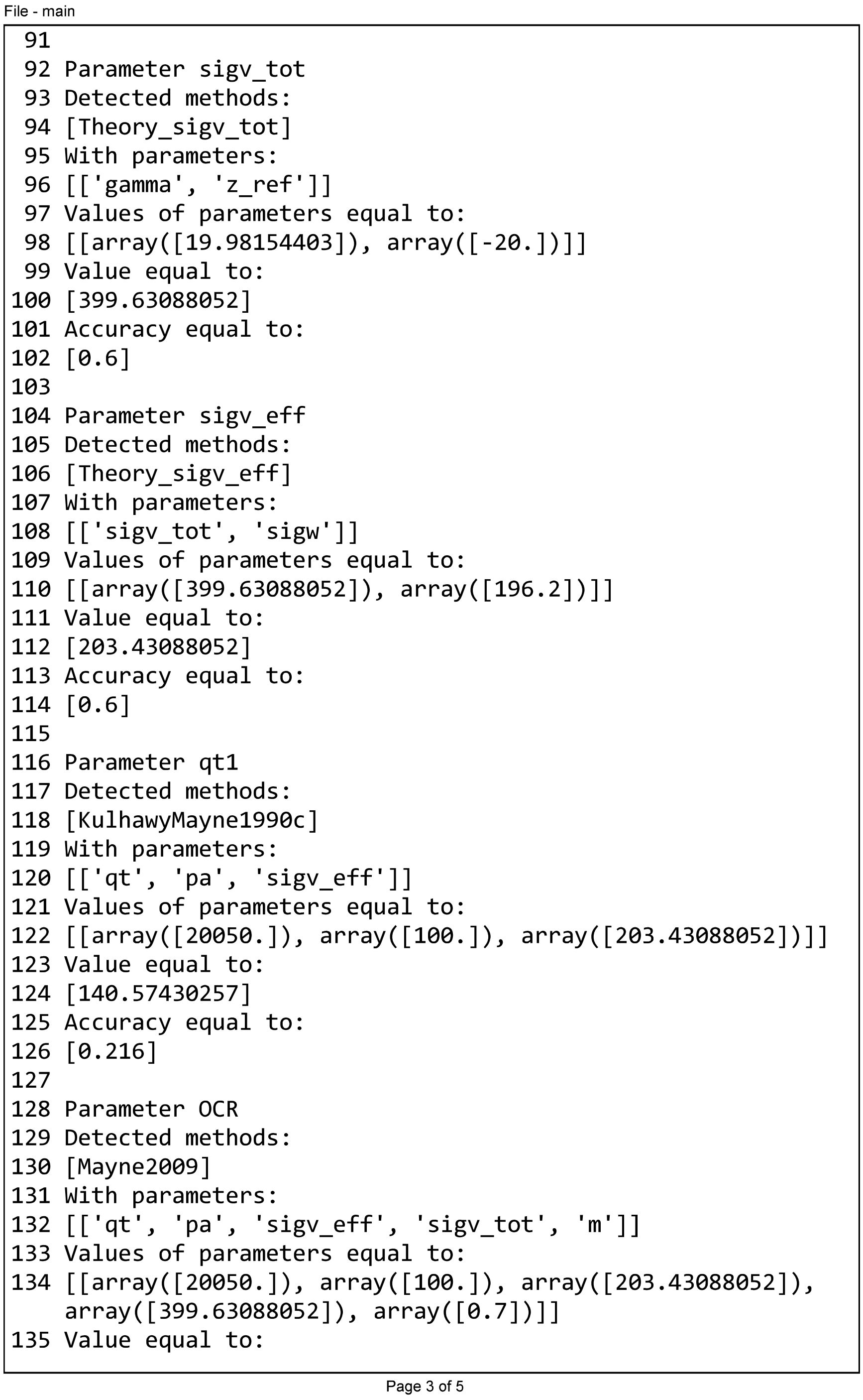


File - main

```
Value equal to:
[0.]
Accuracy equal to:
[1.]

Parameter z_ref
Value equal to:
[-20.]
Accuracy equal to:
[1.]

Parameter qt
Detected methods:
[Robertson1986]
With parameters:
[['qc', 'u2', 'a']]
Values of parameters equal to:
[[array([20000.]), array([250.]), array([0.8])]]
Value equal to:
[20050.]
Accuracy equal to:
[0.6]

Parameter gamma
Detected methods:
[MaynePeuchen2012]
With parameters:
[['fs']]
Values of parameters equal to:
[[array([200.])]]
Value equal to:
[19.98154403]
Accuracy equal to:
[0.6]

Parameter sigw
Detected methods:
[Theory_sigw]
With parameters:
[['gamma_water', 'z_ref', 'phreatic_level']]
Values of parameters equal to:
[[array([9.81]), array([-20.]), array([0.])]]
Value equal to:
[196.2]
Accuracy equal to:
[1.]
```

Page 2 of 5

Figure D.5: Output of main.py (Page 2 of 5).

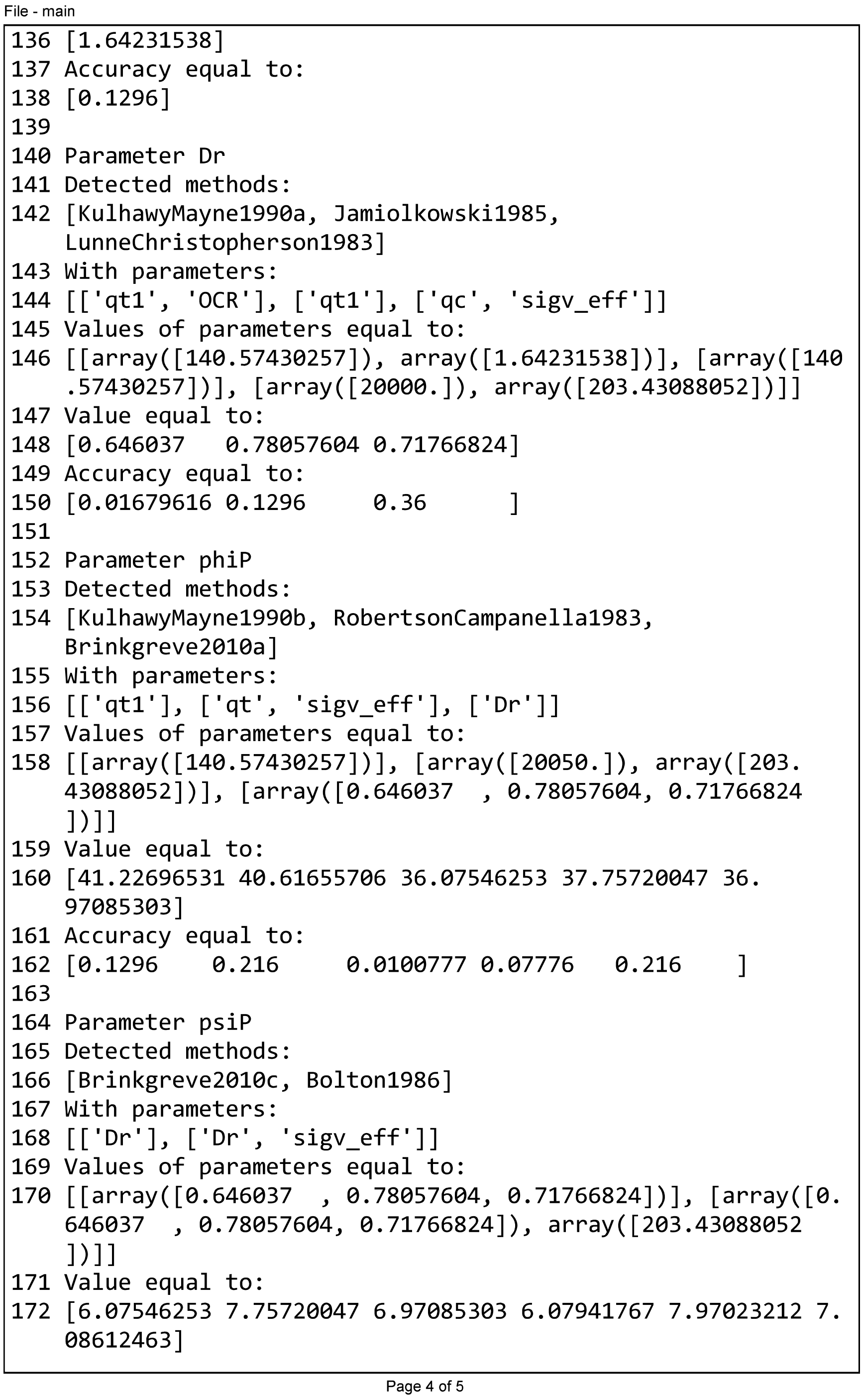


File - main

```

Parameter sigv_tot
Detected methods:
[Theory_sigv_tot]
With parameters:
[['gamma', 'z_ref']]
Values of parameters equal to:
[[array([19.98154403]), array([-20.])]]
Value equal to:
[399.63088052]
Accuracy equal to:
[0.6]

Parameter sigv_eff
Detected methods:
[Theory_sigv_eff]
With parameters:
[['sigv_tot', 'sigw']]
Values of parameters equal to:
[[array([399.63088052]), array([196.2])]]
Value equal to:
[203.43088052]
Accuracy equal to:
[0.6]

Parameter qt1
Detected methods:
[KulhawyMayne1990c]
With parameters:
[['qt', 'pa', 'sigv_eff']]
Values of parameters equal to:
[[array([20050.]), array([100.]), array([203.43088052])]]
Value equal to:
[140.57430257]
Accuracy equal to:
[0.216]

Parameter OCR
Detected methods:
[Mayne2009]
With parameters:
[['qt', 'pa', 'sigv_eff', 'sigv_tot', 'm']]
Values of parameters equal to:
[[array([20050.]), array([100.]), array([203.43088052]),
array([399.63088052]), array([0.7])]]
Value equal to:
```

Page 3 of 5

Figure D.6: Output of main.py (Page 3 of 5).

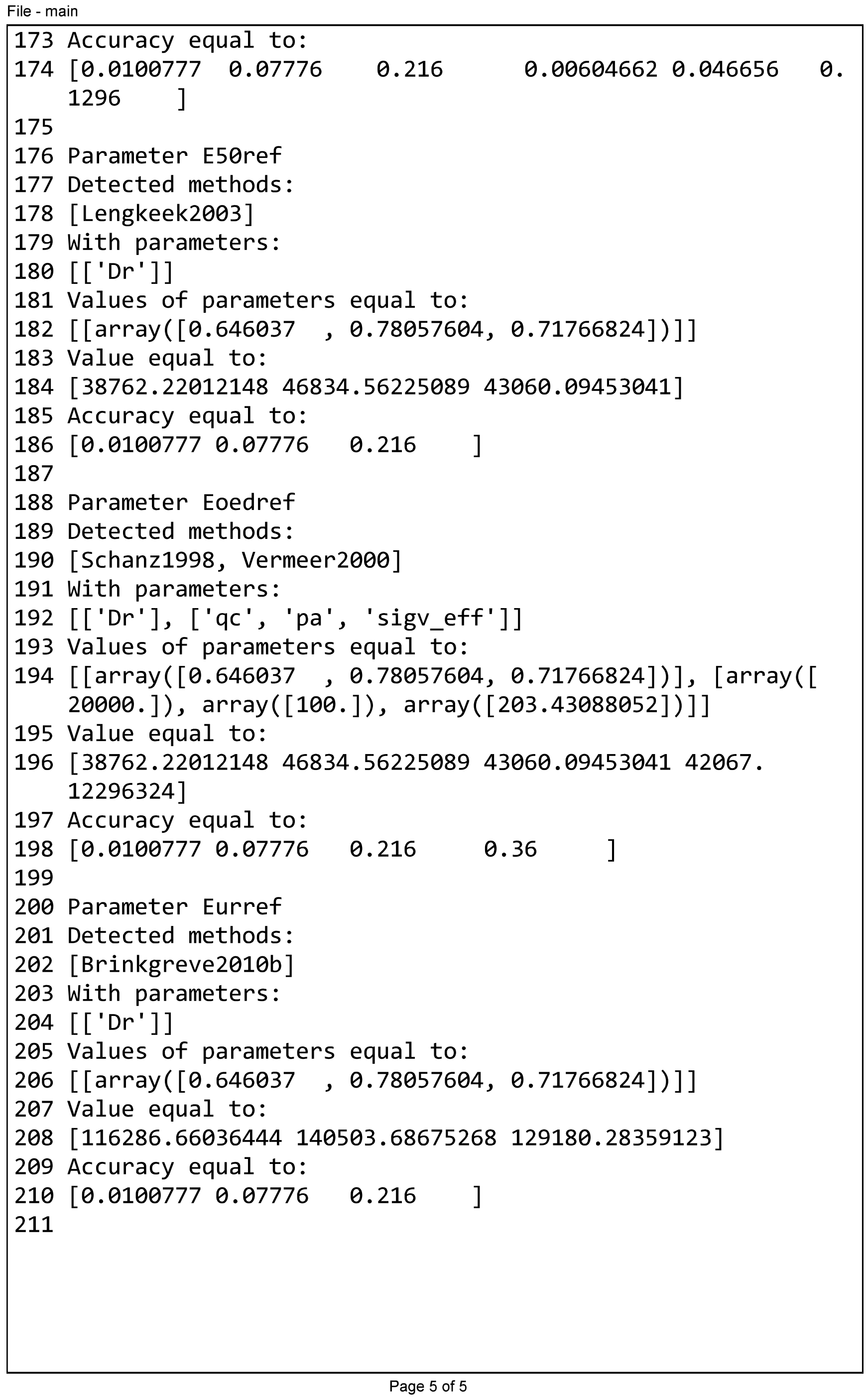


File - main

```
[1.64231538]
Accuracy equal to:
[0.1296]

Parameter Dr
Detected methods:
[KulhawyMayne1990a, Jamiolkowski1985,
LunneChristopherson1983]
With parameters:
[['qt1', 'OCR'], ['qt1'], ['qc', 'sigv_eff']]
Values of parameters equal to:
[[array([140.57430257]), array([1.64231538])], [array([140
.57430257])], [array([20000.]), array([203.43088052])]]
Value equal to:
[0.646037   0.78057604 0.71766824]
Accuracy equal to:
[0.01679616 0.1296     0.36      ]

Parameter phiP
Detected methods:
[KulhawyMayne1990b, RobertsonCampanella1983,
Brinkgreve2010a]
With parameters:
[['qt1'], ['qt', 'sigv_eff'], ['Dr']]
Values of parameters equal to:
[[array([140.57430257])], [array([20050.]), array([203.
43088052])], [array([0.646037  , 0.78057604, 0.71766824
])]]
Value equal to:
[41.22696531 40.61655706 36.07546253 37.75720047 36.
97085303]
Accuracy equal to:
[0.1296    0.216     0.0100777 0.07776   0.216    ]

Parameter psiP
Detected methods:
[Brinkgreve2010c, Bolton1986]
With parameters:
[['Dr'], ['Dr', 'sigv_eff']]
Values of parameters equal to:
[[array([0.646037  , 0.78057604, 0.71766824])], [array([0.
646037  , 0.78057604, 0.71766824]), array([203.43088052
])]]
Value equal to:
[6.07546253 7.75720047 6.97085303 6.07941767 7.97023212 7.
08612463]
```

Page 4 of 5

Figure D.7: Output of main.py (Page 4 of 5)



File - main

```
Accuracy equal to:
[0.0100777  0.07776    0.216      0.00604662 0.046656   0.
1296    ]

Parameter E50ref
Detected methods:
[Lengkeek2003]
With parameters:
[['Dr']]
Values of parameters equal to:
[[array([0.646037  , 0.78057604, 0.71766824])]]
Value equal to:
[38762.22012148 46834.56225089 43060.09453041]
Accuracy equal to:
[0.0100777 0.07776   0.216    ]

Parameter Eoedref
Detected methods:
[Schanz1998, Vermeer2000]
With parameters:
[['Dr'], ['qc', 'pa', 'sigv_eff']]
Values of parameters equal to:
[[array([0.646037  , 0.78057604, 0.71766824])], [array([
20000.]), array([100.]), array([203.43088052])]]
Value equal to:
[38762.22012148 46834.56225089 43060.09453041 42067.
12296324]
Accuracy equal to:
[0.0100777 0.07776   0.216     0.36     ]

Parameter Eurref
Detected methods:
[Brinkgreve2010b]
With parameters:
[['Dr']]
Values of parameters equal to:
[[array([0.646037  , 0.78057604, 0.71766824])]]
Value equal to:
[116286.66036444 140503.68675268 129180.28359123]
Accuracy equal to:
[0.0100777 0.07776   0.216    ]

```

Page 5 of 5

Figure D.8: Output of main.py (Page 5 of 5).

References

ASTM. (1989). *Standard test method for classification of soils for engineering purposes* (Tech. Rep.). American Society for Testing and Materials.

Been, K., Quiñonez, A., & Sancio, R. (2010). Interpretation of the CPT in engineering practice. In *Proceedings of the 2nd International Symposium on Cone Penetration Testing* (pp. 27–45). Huntington Beach, CA, USA.

Begemann, H. (1965). The friction jacket cone as an aid in determining the soil profile. In *Proceedings of the 6th International Conference on Soil Mechanics and Foundation Engineering, ICSMFE, Montreal, September* (pp. 8–15).

Benz, T. (2007). *Small-strain stiffness of soils and its numerical consequences* (No. 55). Stuttgart: Inst. für Geotechnik. (OCLC: 180168062)

Bolton, M. (1986). The strength and dilatancy of sands. *Géotechnique*, *36*(1), 65–78. doi: 10.1680/geot.1986.36.1.65

Brinkgreve, R. (2005). Selection of soil models and parameters for geotechnical engineering application. In *Soil constitutive models: Evaluation, selection, and calibration* (pp. 69–98).

Brinkgreve, R. (2019). Automated Model and Parameter Selection. *Geostrata*, 41–47.

Brinkgreve, R., Engin, E., & Engin, H. (2010). Validation of empirical formulas to derive model parameters for sands. In Benz & Nordal (Ed.), *Numerical methods in geotechnical engineering* (pp. 137–142). Rotterdam, the Netherlands: CRC Press/Balkema.

Campanella, R., Gillespie, D., & Robertson, P. (1982). Pore pressures during cone penetration testing. In *Proceedings of the 2nd European Symposium on Penetration Testing* (pp. 507–512). Amsterdam: A.A. Balkema.

Chakraborty, A., Dutta, T., Mondal, S., & Nath, A. (2018). Application of Graph Theory in Social Media. *International Journal of Computer Sciences and Engineering*, *6*, 722–729.

Douglas, B., & Olsen, R. (1981). Soil classification using electric cone penetrometer. In *Symp. on Cone Penetration Testing and Experience, Geotech. Engrg. Div.* (pp. 209–227). ASCE.

Gansner, E. (2011). Drawing graphs with Graphviz. *Technical report, AT&T Bell Laboratories, Murray, Tech. Rep, Tech. Rep.*.

Goh, A., & Kulhawy, F. (2003). Neural network approach to model the limit state surface for reliability analysis. *Canadian Geotechnical Journal*, *40*(6), 1235–1244.

Graham, J. (2006). The 2003 R.M. Hardy Lecture: Soil parameters for numerical analysis in clay. *Canadian Geotechnical Journal*, *43*(2), 187–209. doi: 10.1139/t05-098

Jamiolkowski, M. (2001). Where are we going. In *Proceedings of the 2nd International Symposium on Pre-failure Deformation Characteristics of Geomaterials, Torino* (Vol. 2, pp. 1251–1262).

Jamiolkowski, M., Ladd, C., Germaine, J., & Lancellotta, R. (1985). New developments in field and laboratory testing of soils. In *Proceedings of the 11th International Conference on Soil Mechanics and Foundation Engineering* (Vol. 1, pp. 57–153). Rotterdam: Balkema Publications.

Jamiolkowski, M., Lo Presti, D., & Manassero, M. (2001). Evaluation of Relative Density and Shear Strength of Sands from CPT and DMT. In *Soil Behavior and Soft Ground Construction* (pp. 201–238). Cambridge, Massachusetts, United States: American Society of Civil Engineers. doi: 10.1061/40659(2003)7

Jefferies, M., & Been, K. (2006). *Soil liquefaction: A critical state approach* (2nd ed.). Abingdon, UK: Taylor & Francis.

Jefferies, M., & Davies, M. (1991). Soil classification by the cone penetration test: Discussion. *Canadian Geotechnical Journal*, *28*(1), 173–176.

Jefferies, M., & Davies, M. (1993). Use of CPTU to estimate equivalent SPT N 60. *Geotechnical Testing Journal*, *16*(4), 458–468.

Kulhawy, F., & Mayne, P. (1990). *Manual on estimating soil properties for foundation design* (Tech. Rep.). Palo Alto, CA (USA): Electric Power Research Institute, EPRI.

Lengkeek, H. (2003). EstimationOfSandStiffnessParametersFromConeResistance. *Plaxis Bulletin*, *13*, 15–19.

Lengkeek, H., de Greef, J., & Joosten, S. (2018). CPT based unit weight estimation extended to soft organic soils and peat. In *Proceedings of the 4th international symposium on cone penetration testing* (pp. 389–397). Delft, the Netherlands: CRC Press.

Lunne, T., & Christoffersen, H. (1983). Interpretation of cone penetrometer data for offshore sands. In *Offshore Technology Conference.* Offshore Technology Conference.

Lunne, T., Powell, J., & Robertson, P. (1997). *Cone penetration testing in geotechnical practice.* CRC Press.

Mayne, P. (2005). Integrated ground behaviour: In-situ and laboratory tests. In *Proceedings of the 3rd International Symposium on Deformation Characteristics of Geomaterials: Taylor and Francis* (pp. 155–176).

Mayne, P. (2006a). In-situ test calibrations for evaluating soil parameters. In *Characterisation and engineering properties of natural soils* (Vol. 3, pp. 1601–1652). Singapore: Taylor & Francis Group, London.

Mayne, P. (2006b). The Second James K. Mitchell Lecture Undisturbed sand strength from seismic cone tests. *ResearchGate.*

Mayne, P. (2007). NCHRP Synthesis 368, Cone Penetration Testing. *Transportation Research Board, Washington, DC*, *118*.

Mayne, P. (2009). State-of-the-art paper (SOA-1): Geomaterial behavior and testing. *International Converence on Soil Mechanics and Geotechnical Engineering.*

Mayne, P. (2013). Evaluating yield stress of soils from laboratory consolidation and in-situ cone penetration tests. In *Sound Geotechnical Research to Practice: Honoring Robert D. Holtz II* (pp. 405–419).

Mayne, P. (2014). Interpretation of geotechnical parameters from seismic piezocone tests. In *Proceedings, 3rd International Symposium on Cone Penetration Testing* (pp. 47–73). Las Vegas.

Mayne, P., & Kulhawy, F. (1982). K0 - OCR Relationships in Soil. *Journal of the Soil Mechanics and Foundations Division*, *108*(6), 851–872.

Mayne, P., & Peuchen, J. (2012). Unit weight trends with cone resistance in soft to firm clays. In *Geotechnical and Geophysical Site Characterization 4.* London: Taylor & Francis.

Mayne, P., Peuchen, J., & Bouwmeester, D. (2010). Soil unit weight estimation from CPTs. In *Proceedings of the 2nd International Symposium on Cone Penetration Testing* (Vol. 2, pp. 167–176). Huntington Beach, CA, USA: CPT '10 Organizing Committee.

Nhuan, B. (1981). *Sample disturbance and its effects on the geotechnical properties of clays* (Tech. Rep.). Linköping: Swedish Geotechnical Institute.

Reale, C., Gavin, K., Librić, L., & Jurić-Kaćunić, D. (2018). Automatic classification of fine-grained soils using CPT measurements and Artificial Neural Networks. *Advanced Engineering Informatics*, *36*, 207–215.

Robertson, P. (1990). Soil classification using the cone penetration test. *Canadian Geotechnical Journal*, *27*(1), 151–158. doi: 10.1139/t90-014

Robertson, P. (2009). Interpretation of cone penetration tests — a unified approach. *Canadian Geotechnical Journal*, *46*(11), 1337–1355. doi: 10.1139/T09-065

Robertson, P. (2010). Soil behaviour type from the CPT: An update. In *2nd International Symposium on Cone Penetration Testing* (pp. 9–11). Huntington Beach, CA, USA.

Robertson, P. (2015). *Guide to cone penetration testing for geotechnical engineering* (6th ed.). Gregg Drilling & Testing.

Robertson, P. (2016). Cone penetration test (CPT)-based soil behaviour type (SBT) classification system — an update. *Canadian Geotechnical Journal*, *53*(12), 1910–1927. doi: 10.1139/cgj-2016-0044

Robertson, P., & Cabal, K. (2010). Estimating soil unit weight from CPT. In *Proceedings of the 2nd International Symposium on Cone Penetration Testing* (Vol. 3, pp. 2–40). Huntington Beach, CA, USA.

Robertson, P., & Campanella, R. (1983). Interpretation of cone penetration tests. Part I: Sand. *Canadian geotechnical journal*, *20*(4), 718–733.

Robertson, P., Campanella, R., Gillespie, D., & Greig, J. (1986). Use of piezometer cone data. In

Use of in situ tests in geotechnical engineering (pp. 1263–1280). ASCE.

Robertson, P., & Wride, C. (1998). Evaluating cyclic liquefaction potential using the cone penetration test. *Canadian Geotechnical Journal*, *35*(3), 442–459.

Schnaid, F. (2009). *In situ testing in geomechanics: The main tests.* Taylor & Francis.

Schweiger, H., Fabris, C., Ausweger, G., & Hauser, L. (2018). Examples of successful numerical modelling of complex geotechnical problems. *Innovative Infrastructure Solutions*, *4*(1), 1–10.

Senneset, K., & Janbu, N. (1985). Shear strength parameters obtained from static cone penetration tests. In *Strength Testing of Marine Sediments: Laboratory and In-situ Measurements.* ASTM International.

Shahin, M., Jaksa, M., & Maier, H. (2001). Artificial Neural Network Applications in Geotechnical Engineering. *Australian Geomechanics*, *3*, 49–62.

Shu-Xi, W. (2012). The Improved Dijkstra's Shortest Path Algorithm and Its Application | Elsevier Enhanced Reader. *Procedia Engineering*, *29*, 1186–1190.

Singh, H., & Sharma, R. (2012). Role of adjacency matrix & adjacency list in graph theory. *International Journal of Computers & Technology*, *3*(1), 179–183.

Stroud, M. (1988). The standard penetration test–its application and prediction. In *Proceedings of the geotechnology conference 'penetration testing in the UK'organized by the Institution of Civil Engineers.* Birmingham.

Vermeer, P. (2000). Column Vermeer. *Plaxis Bulletin*(9).

Wood, D. (2004). *Geotechnical Modelling.* Oxfordshire: Taylor & Francis.

Wroth, C. (1988). Penetration testing - A more rigorous approach to interpretation. *ISOPT-1, 1988*, *1*, 303–311.

Zuidberg, H., Schaap, L., & Beringen, F. (1982). A penetrometer for simultaneously measuring of cone resistance, sleeve friction and dynamic pore pressure. In *Proc. of the 2nd International Symposium on Penetration Testing, ISOPT-II, Amsterdam* (Vol. 2, pp. 963–980). Rotterdam: Balkema Publications.