UNIVERSIDAD **NACIONAL** DE COLOMBIA

Reliability Assessment of Rock Slopes by Evidence Theory

Rodrigo Hernández Carrillo

Reliability Assessment of Rock Slopes by Evidence Theory

Rodrigo Hernández Carrillo

Tesis presentada como requisito parcial para optar al título de:
Doctor en Ingeniería Civil

Directora:
Gloria Inés Beltrán, IC, MSc, PhD

Línea de Investigación:
Geotecnia y Riesgos Geoambientales
Grupo de Investigación:
Grupo de Investigación en Geotecnia (GIGUN)

Universidad Nacional de Colombia
Facultad de Ingeniería
Área Curricular de Ingeniería Civil y Agrícola
Bogotá, Colombia
Agosto, 2020

A Kate, mi esposa, sin tu apoyo y dedicación
no lo hubiese logrado

A mis padres, para quienes nuestra
educación siempre fue prioridad

Aknowledgements

Quiero expresear mi gratitud con (I want to express my gratitude with):

Abstract

This research project aims to develop a methodology to perform rock slope stability analysis considering the aleatory and epistemic uncertainty when the information on geomechanical parameters is limited. In rock mechanics, deterministic and probabilistic approaches are widely used in the decision-making process. However, the earlier does not consider the uncertainty, and the latter has limitations to account for the epistemic uncertainty and requires assumptions on probability distributions when robust data sets are not available. Therefore, we resorted to the Evidence Theory as a tool to describe the epistemic and aleatory uncertainty of input geomechanical variables and propagate them trough limit equilibrium models, in which the geometry is controlled by the joints orientation. To perform a better description of the variability of the rock mas properties, the project utilized a short-range photogrammetry system, which allowed us to have robust and reliable data sets on joints geometry to be modeled as Kent distributed variables. Besides, we suggested a procedure to update the reliability analysis acknowledging that orientations follow a Kent distribution. The application of the methodology to a rock slope in a sandstone mine showed its suitability to be applied in actual engineering projects. Consequently, the main contribution of this project is an rock slope evidence theory reliability-based framework for combining robust data sets on joints orientation, with limited information on geomechanical parameters, that can be updated as new information is available.

Key words: Theory of Evidence, Dempster Shafer Structure, reliability, epistemic uncertainty, aleatory uncertainty, slope stability, limit equilibrium, rock wedge, discrete fracture network (DFN), short-range photogrammetry, mixtures model .

Resumen

El objetivo de este proyecto de investigación es desarrollar una metodología para efectuar análisis de confiabilidad de la estabilidad de taludes rocosos, teniendo en cuenta la incertidumbre cuando la información sobre los paráme-

tros geomecánicos de entrada es limitada. En mecánica de rocas, los métodos determinísticos y probabilísticos son ampliamente utilizados en el proceso de toma decisiones. No obstante, el primero no considera la incertidumbre y el segundo tiene limitaciones para representar la incertidumbre epistémica y tiene que asumir la distribución de probabilidad de las variables de entrada. Por lo tanto, se recurre a la Teoría de la Evidencia como una herramienta para describir la incertidumbre aleatoria y epistémica de los parámetros geomecánicos y propagarla a través de modelos de equilibrio límite, en los que la geometría es controlada por la orientación de las discontinuidades. Para llevar a cabo una mejor descripción de la variabilidad en el macizo, el proyecto utilizó fotogrametría de corto alcance, lo que permitió obtener series de datos robustas y confiables de la geometría de las discontinuidades, que fue modelada como una variable aleatoria con distribución Kent. Además, se desarrolló un procedimiento para actualizar los análisis de confiabilidad teniendo en cuenta la distribución de probabilidad de la orientación de las discontinuidades. La aplicación de la metodología en un talud rocoso de una mina de arenisca mostró su aplicabilidad a proyectos reales. Consecuentemente, la principal contribución de este trabajo es la generación de un marco de referencia para efectuar la evolución de confiabilidad de taludes rocoso basado en la teoría de la evidencia que permite combinar las series robustas de la orientación de los planos de discontinuidad, con información limitada de sus parámetros de resistencia, que puede ser actualizada a medida que se genera nueva información.

Palabras clave: Teoría de la Evidencia, Estructuras Dempster Shafer, confiabilidad, incertidumbre aleatoria, incertidumbre epistémica, estabilidad de taludes, equilibrio límite, cuñas de roca, redes discretas de fracturamiento, fotogrametría de corto alcance, modelo de mezclas.

Content

List of Figures

List of Tables

1. Research Framework

Colombia is a developing country with active mining industry and growing development of infrastructure. In most of these projects, engineered rock slopes need to be excavated. Thus, engineers should be able to perform reliable designs in rock masses, considering their nature and variability.

Rock masses are subjected to a long history of stresses such as tectonic loads, earth-quakes, glaciations, subsidence, tidal effects, and gravity. In most cases, these stresses bring fracturing to the rock mass, which controls the rock mass response. Therefore, the main feature of rock masses that makes them different from other civil engineering materials is the presence of discontinuities. Hence, rock joints should be explicitly accounted for in rock mass models intended for predicting its response under perturbations. This can be achieved by defining mechanisms of failure or by modeling the entire rock mass structure explicitly.

Regarding the mechanisms of failure, the planar, wedge, and toppling failure are widespread models utilized in rock slope practice. The block theory [Shi et al., 1985] provides an alternative to identify arbitrary shape unstable blocks within several joint sets. However, it does not consider the spatial location of those joints. The discrete fracture networks (DFN) overcome the key block limitations because they model each fracture plane, with its corresponding location within a reference volume.

Regardless of the representation of the rock structure, each model is just a realization of such an uncertain geometry. Hence, to have a better understanding of the most likely response of the rock mass, several realizations of the rock mass geometry are required along with the corresponding mechanical analysis.

The reliability analysis provides a systematic framework to account for the uncertainty and propagate it through the desired model. However, it requires

a sound selection of the model because the computational cost constraints the applicability of the reliability assessment when several realizations are required.

Conventionally, reliability assessment is performed by probabilistic techniques that need to know beforehand the probability distribution of the inputs. However, this is not possible in many geotechnical problems, because of the limited information on materials properties. Therefore, alternative concepts have been developed to cope with uncertainty properly.

These alternative approaches are based mainly in interval estimates, including fuzzy sets, interval probability, and Evidence Theory, which have been subsumed in the term imprecise probability. Among these approaches, the Dempster Shafer Theory of Evidence, DST, is a suitable tool for taking into account the uncertainty with limited information and can be conveniently visualized [Oberguggenberger, 2012]

In consonance with the aforementioned, this research aims at developing a methodology for performing reliability assessment of rock slopes considering that discontinuities control the rock mass response. This is accomplished by assessing the most common mechanism of failure and DFN, within a DST framework.

At this point, it is worth to mention that this research was part of the project "Quantitative Risk Assessment in Mining Rock Slopes" funded by the National University of Colombia and Colciencias. Hence, the results and contributions of this research were relevant inputs to that project.

As for the structure of this document, this chapter presents the problem statement and justification, objectives, methodology, and the main contributions; in the second chapter, the Dempster Shafer Theory of Evidence, DST, is introduced as a framework to deal with the epistemic and aleatory uncertainty.

Subsequently, several wedge failure stability analyzes are conducted, in which the uncertainty components are progressively added. Thus, in Chapter 3, probabilistic approaches with Kent distribution are utilized to represent the aleatory uncertainty of joints orientation. After, in Chapter 4, an algorithm to combine structural with mechanical information for computing rock wedge stability within the DST framework is presented. This procedure is extended

to the stability analysis using Discrete Fracture Networks in Chapter 5. Additionally, Chapter 6 explores the DST as probable reasoning for updating the reliability analysis of the stability of rock slopes. Finally, Chapter 7 compiles the main conclusions and recommendations for future research.

Each chapter is illustrated by examples, based on the information collected in a sandstone mine (quarry) located in Une, Cundinamarca, Colombia, operated by Gravillera Albania that provided the information they have collected since 1997, as part of their thorough supervision program. Besides, in this project, a short range stereoscopy technique was introduced in Colombia to gather structural information on the rock mass, as detailed in Appendix A.

1.1.   Problem statement

Rock masses are natural geological materials affected by fractures at different scales. These fractures highly influence the mechanical response of the rock mass, even more, when fractures are filled with any fluid. Therefore, in most cases, rock masses behave as a discrete medium, in which joints and blocks are continuously interacting.

To mimic the response of rock slopes, several models are available at different scales. For one to one mapping scale, numerical models are the more suitable modeling technique. These models are mainly continuum and discontinuum based. Considering the discrete nature of rock masses, the discontinuum based methods are the most appropriate for modeling the performance of rock slopes. The key concept of the discrete approach is that the problem domain is treated as an assemblage of rigid or deformable blocks, and the contacts between them need to be identified and continuously updated during the entire deformation/motion process, and represented by proper constitutive models [Jing, 2003]. In other words, multiple discrete blocks interact with continuously changing contacts. Hence, such methods are computationally expensive and time-consuming [Khan, 2010].

Alternatively, limit equilibrium based method, in which potentially unstable blocks are identified, and subsequently, their stability is assessed. This approach is suitable when input parameters are uncertain, and several realizations are required to capture and propagate such an uncertainty since this

method is computationally less expensive than discrete elements.

The limit equilibrium method requires both mechanical and geometrical properties of rock masses. Nonetheless, the determination of these properties is not straightforward, since the rock mass is a heterogeneous material, in which physical and mechanical properties change from point to point. This variability introduces an aleatory uncertainty to the model. Moreover, there is an epistemic uncertainty linked to the 'imperfect knowledge' about the property. Besides, given the nature of the information and technical and financial constraints, the amount and quality of information are very heterogeneous as well. In rock mechanics, there are robust data sets on rock mass geometrical features (joints dip and dip direction, separation, and persistence), while information on joints strength parameters is limited to a few results. This feature should be considered when the rock mass response is analyzed.

There are several alternatives to deal with uncertainty. For engineers, the most popular alternative is the purely probabilistic approach. However, the classic probability cannot consider the epistemic uncertainty, and the input probabilities are required to perform the reliability assessment. Otherwise, simplifications and assumptions on the input probability distributions are required. [Oberguggenberger, 2012].

In this context, alternative approaches have appeared since the early 90ies under the name of imprecise probabilities and are based on interval analysis, such as DST. These models of the data uncertainty should reflect and incorporate the level of information available in the data and propagate it through numerical computation and deliver an output whose uncertainty is formulated in the same terms [Oberguggenberger, 2012].

Accordingly, the DST is a suitable tool to deal with uncertainty (aleatory and epistemic) in rock slopes modeling, since these provide an output that reflects the uncertainty of input parameters, even when these are the result of limited information. Techniques such as DST allow defining bounds of the probability of the model response.

Based on the above mentioned, there is a need to deal with uncertainty (under limited information) in rock mass models that capture its discrete nature. Consequently, an approach capable of coping with the uncertainty associated with the geometry and mechanical properties of rock masses, based on limit

equilibrium methods is required. Therefore, in this project, the capabilities of DST as a tool to perform reliability assessment of rock slopes, articulating robust data sets on rock slope geometry, and limited information on geomechanical parameters were explored.

1.2.   Justification

The mining industry in Colombia has had an increasing role in the economy in the last 10 years. Nowadays, 8 % of GDP is provided by the mining activity. In 2014 Colombia extracted 16.765.586 tons of minerals, mainly limestone for cement production (92 %) [UPME, 2015], see Figure **1-1** . This figure includes the production of mineral in Colombia aside from limestone.

Most of the limestone and coal is extracted in opencast excavations. Hence, this industry has to deal with many rock slopes, which have to be properly designed to guarantee a safe operation.
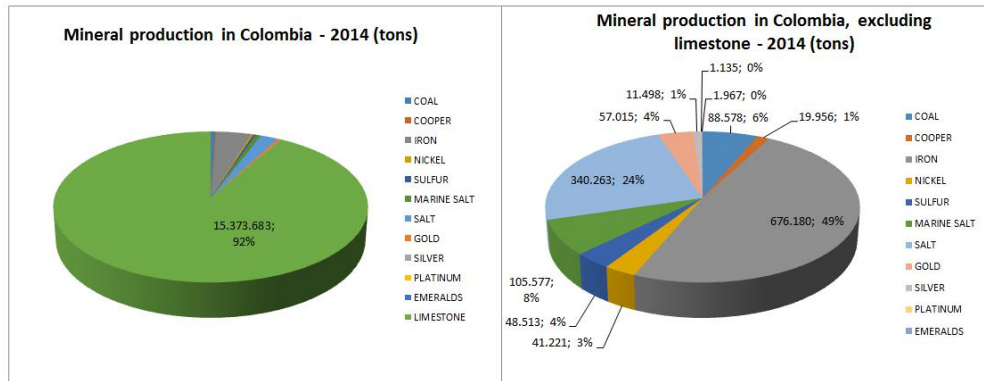
Figure 1-1.: Mineral production in Colombia in 2014, figures from UPME [2015]

Unfortunately, the Colombian mining industry has a high rate of accidents with many injuries and fatalities per year. As stated in Agencia Nacional de Mineria [2019] between 2005 and 2019, 1272 emergencies related to mining activity were reported, in which 1448 people died, see Figure **1-2**. From these casualties, 37 % of the fatalities are associated with excavation failures, either underground or surface excavations, as can be observed in Figure **1-3**.

On the other hand, Colombia is a developing country with 16700 km of
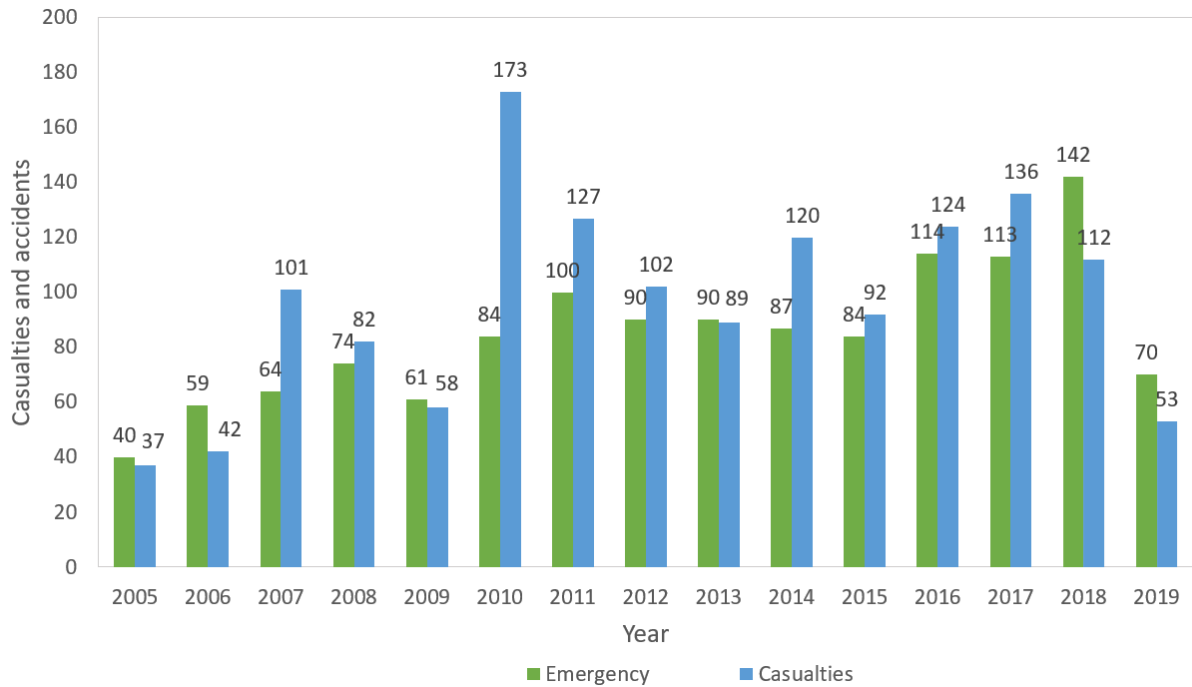
Figure 1-2.: Fatalities associated with mining activities between 2005 and July 2019 [Agencia Nacional de Mineria, 2019]
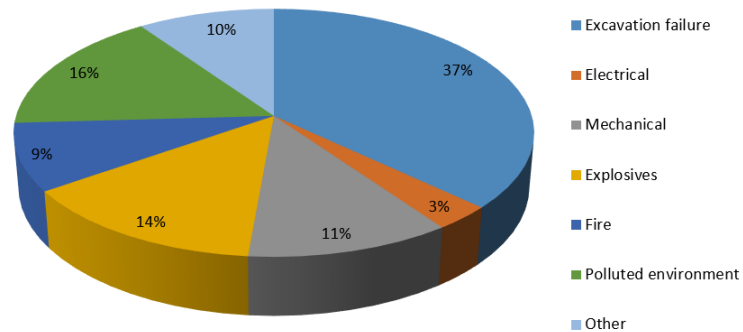
Figure 1-3.: Distribution of fatalities associated with mining activities between 2005 and July 2019 [Agencia Nacional de Mineria, 2019]

primary road network in 2012 [DPN, 2013], in which 71 % of freight transport was made by roads in 2010 (around 181.021 million tons) [Mintransporte, 2011] . This road network is concentrated in mountainous areas of the country (Andean region of Colombia). Indeed, 52 % of the whole Colombian road

network is located in highly undulating or mountainous terrain.

Consequently, there is an increasing need for developing more realistic and reliable rock mass models to face the engineering challenges generated by new infrastructure projects in Colombia, but also to reduce the number of casualties associated to rock mass failure in mining.

Two key aspects of rock masses have to be considered to formulate more realistic and accurate models: the discontinuum nature of rock masses and the uncertainty associated with their heterogeneity.

So far, few pieces of research have been published on the analysis of the mechanical behavior of the rock mass using probability-based methods to consider the randomness of the discontinuity geometry [Ma and Fu, 2014]. Reliability assessment using probabilistic point estimate, response surface, Monte Carlo, and Latin Hypercube methods were presented by Hammah [2009]. An approach to consider the randomness of rock discontinuities with robust block generation algorithms and the key block was proposed by Ma and Fu [2013] and Ma and Fu [2014]. Gheibie et al. [2013] presented a 3D DEM modeling of a rock slope in Turkey, defining a reliability index for the input geomechanical parameters and considering the geometry of the joints as deterministic. Shen and Abbas [2013] developed a 2D DEM model of a landslide in China, in which the uncertainty of geomechanical parameters is represented by random sets, again the discontinuities are assumed as deterministic.

Although some efforts have been made to take into account the uncertainty in rock mass modeling, none research has incorporated the geometrical and geomechanical uncertainty, to perform reliability assessment in rock masses taking advantage of the capabilities of Dempster Shafer Theory of Evidence.

1.3. Objectives

1.3.1. Aim (General objective)

The main objective of this project is to:

Develop a methodology to perform rock slope stability reliability assessment considering both the epistemic and the aleatory uncertainty by Dempster

Shafer Theory of Evidence, combining robust data sets on joint orientation with limited information on geomechanical parameters.

1.3.2. Specific objectives

1. To model the aleatory and epistemic uncertainty of rock mass geomechanical parameters by directional probability distributions and Dempster Shafer structures, respectively.

2. To assess the stability of rock slopes by Dempster Shafer Theory and limit equilibrium models.

3. To develop a procedure for updating the reliability assessment of rock wedges stability under a DST framework.

1.4. Methodology

The first part of the project comprised a detailed characterization of the rock mass. El Pedregal Mine, located in Une Cundinamarca was selected as a case study because they follow a thorough and responsible exploitation process, which includes periodic inspections of the stability of the mine slopes. Moreover, the company shared its database on geomechanical parameters with the university.

This stage was supervised by an experienced geologist who developed a geological characterization of the area identifying the primary lithologies, geological structures, and joint sets, according to geological interpretation. It is desirable to have information collected at different stages of the project or locations since one of the objectives is to suggest an alternative for updating the reliability assessment when new information is gathered.

Simultaneously, a field mapping of representative slopes is carried out to collect detailed information on the joints dip and dip direction, spacing, and trace length. Most of the mapping was performed by a contactless image system. This technique assesses surface using three-dimensional images based on digital photos. The method is intended to extract information to solve engineering geology problems [3GSM GmbH, 2011]. The reconstruction of

every point P(X, Y, Z) at the slope face is performed from a corresponding pair of image points P(u,v) and the known camera orientations. This task will be developed using the commercially available system: ShapeMetrix3D [3GSM GmbH, 2011]. This is included in Appedix A.

When this method is properly calibrated, it allows collecting an important and representative data set on joints geometry since the whole slope face can be automatically recorded, which is not possible by traditional manual surveying techniques.

Regarding the mechanical properties, block samples were taken from the slopes. Then, rock cores were extracted from these blocks and strength parameters along discontinuities measured. Unlike joints geometry, the information obtained from these tests is limited in number, due to the conventional time and budget constraints of almost every engineering project. This is one of the main issues that will be studied in this project.

Once the information from the tests was gathered and processed alternatives to represent the uncertainty of both the geometrical and geomechanical parameters of the rock mass were defined. The former is treated with directional statistics, and the latter with DST, as shown in chapters 3 and 4, respectively. After, based on the selected alternatives to represent the input information, a framework to perform reliability assessment, according to DST, was developed. This is not straightforward since two types of input sets are considered, one with a large amount of information (joint geometry) and one with limited information (geomechanical parameters). This problem was addressed involving Monte Carlo simulations and DST. The proposed methodology was applied to rock slope stability problems, in which the stability is evaluated by limit equilibrium, according to chapters 4 and 5.

Finally, with the information collected at different locations or times, an alternative for updating the resulting distribution functions was proposed, taking advantage of the capabilities of the DST and the mixture model, as described in Chapter 6.

The resulting methodology to perform reliability assessment and the specific results applied to the case study are a relevant input for the "Quantitative Risk Assessment in Mining Rock Slopes"; since this will be the starting point to assess the hazard.

In Figure **1-4**, a diagram of the methodology just described is presented. Here, the shadowed regions show the relation of this procedure with the specific objectives.
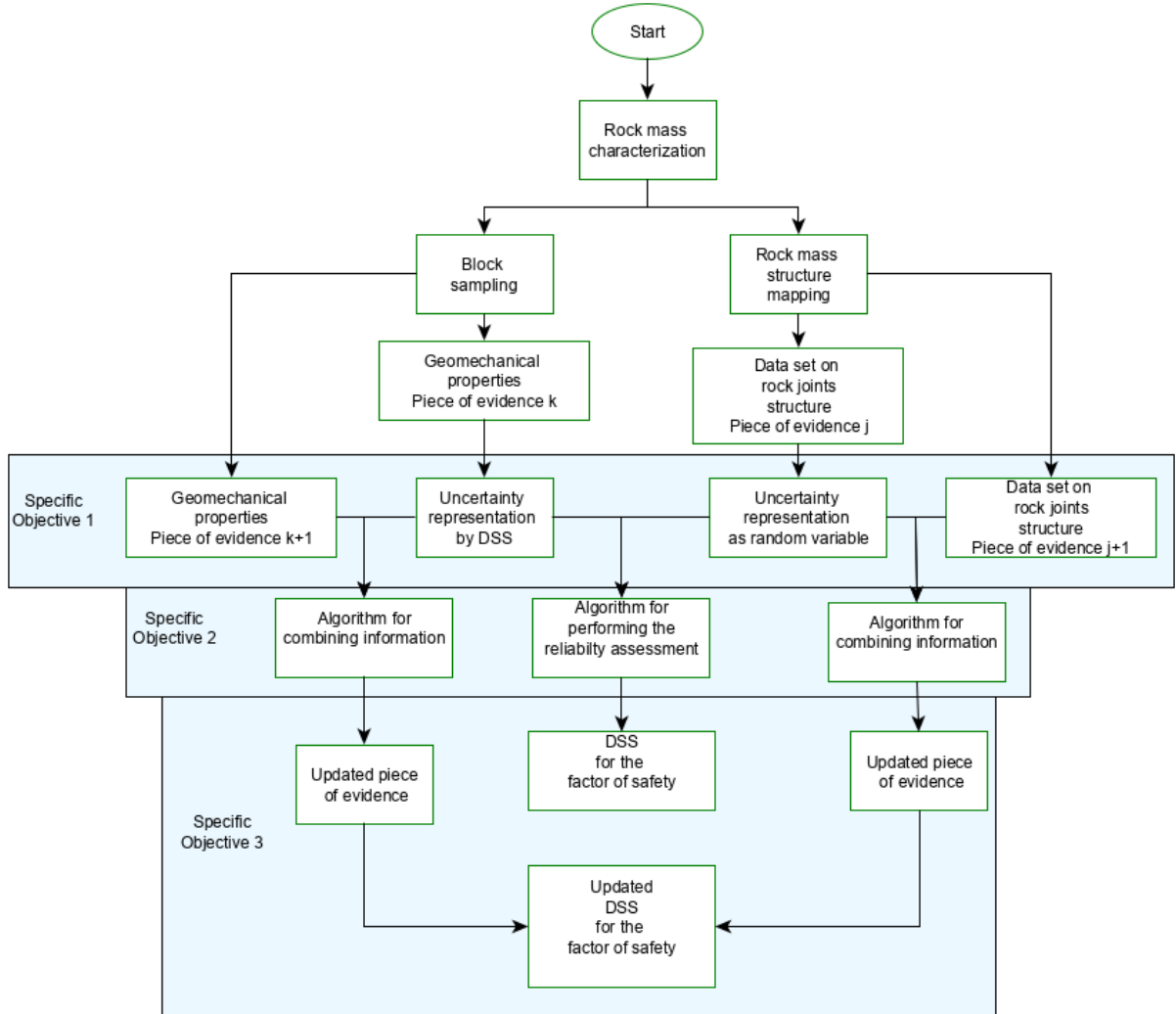
Figure 1-4.: Project methodology

The algorithms suggested in this research project were programmed in Python 3.6 available at www.python.org, using the integrated development environment, IDE, PyCharm 2018.1 available at www.jetbrains.com. These algorithms were coded as functions, which makes them flexible to be incorporated into more general application codes. As for the project, these scripts make feasible the reliability assessment that requires several realizations of the stability model, as shown in the following chapters. The scripts are included in

Appendix D.

1.5.   Contributions

This project deals with the issue of the uncertainty involved in rock engineering coming from the inherent variability of the parameters, and limited information. Accordingly, the main contributions of the project are:

1. An evidence theory-based reliability assessment framework for combining robust datasets on joints orientation that have an aleatory uncertainty, with limited sets of information on geomechanical parameters.

2. A methodology for updating the reliability assessment of the factor of safety as new information is available, based on evidence theory and the mixture model.

Moreover, relevant algorithms were developed to accomplish the general objective of the project. These algorithms are contributions of this research as well:

1. A Monte Carlo simulation of the rock wedge stability considering the orientation of rock joints as a Kent-distributed random variable.

2. A modification of the rock wedge stability computation model proposed by Low [1979, 1997] to consider a broader range of planes combination, which is required for the above mentioned Monte Carlo simulations.

3. A framework for involving the generation of discrete fracture networks within an evidence theory framework.

2. Dempster-Shafer Theory of Evidence

2.1. Introduction

Decision making under uncertainty is one of the main challenges that a geotechnical engineer has to face when dealing with rock slope stability. Any rock mass is by itself a very complex material, which response is controlled by the presence of discontinuities. Therefore, a proper characterization of fracturing is required for design purposes. However, this is not a straightforward task, since rock masses are heterogeneous materials, in which physical and mechanical properties are variable even for the same lithology.

Such a natural variability on properties introduces an aleatory uncertainty to the problem. In addition, the 'imperfect knowledge' resulting from the limited amount of information available, brings an epistemic component to the uncertainty. For making an informed decision, both epistemic an aleatory uncertainty should be accounted for throughout the design process, which is accomplished by performing a reliability assessment, that can be seen as a systematic way to account for uncertainty.

The need for considering the uncertainty in civil engineering has been acknowledged since the 1950ies [Freudenthal, 1956]. Later, Casagrande [1965] highlighted the necessity of calculating and even more important, dealing with uncertainty in geotechnical projects. Whitman [1984] presented a comprehensive description of the evaluation alternatives approaches to use probability theory in geotechnical applications, such as optimization of exploration and design, risk assessment and reliability. Whitman stressed that the quality of probabilistic results is not better than the quality of the deterministic model underlying and the quality of the inputs.

Reliability assessment has been used to solve rock slope stability problems since 1970s [Glynn, 1979, McMahon, 1971], including the conditional proba-

bility of failure, to account for the kinematics and kinetics of the failure. As mentioned in Chapter 3, the primary trend to consider uncertainty has been using the probabilistic approach to solve individual slopes, such as Low [1979, 1997], Park [1999], Park and West [2001], considering correlated non-normal input parameters [Ahmadabadi and Poisel, 2016, Low, 2007] and failure as a system of multiple blocks [Jimenez-Rodriguez and Sitar, 2006, Jimenez-Rodriguez et al., 2006, Park et al., 2005], among others.

All these publications focus on developing a reliability assessment of rock slope by probabilistic techniques. They model structurally controlled failures with explicit formulations, which are suitable for performing several realizations of the model at a low computational cost. In fact, many of the proposed methodologies are compared against Monte Carlo simulations. On the other hand, researchers resort to secondary information or expert opinion to define the probabilistic input parameters required for implementing these techniques, due to the limited information available.

The main limitations of this purely probabilistic approach are the need for information, which is not available in most geotechnical projects [Oberguggenberger, 2012] and its limitations to capture epistemic uncertainty [Sentz and Ferson, 2002]. Alternative concepts have been developed to properly deal with uncertainty, considering that probabilistic models are limited to deal with uncertainty in the geotechnical field. The concept behind those approaches is that the presence of epistemic uncertainty leads to a family of probability functions, with a structure derived from a mathematical model representing the aleatory uncertainty [Johnson et al., 2008].

These alternative approaches are based mainly on interval estimates, including fuzzy sets, interval probability, and evidence theory, which have been subsumed in the term imprecise probability. Among these approaches, the Dempster-Shafer structures provide a framework for bridging the gap between probability and interval analysis and admitting easily accessible visualization tools [Oberguggenberger, 2012].

2.2. Theoretical background

The uncertainty evaluation in engineering modeling is a particular case of error propagation. This concept implies that the uncertainty of input parameters corresponds to an error that has to be propagated throughout the calculation process performed by the engineering model [Baecher and Christian, 2003]. Accordingly, the reliability assessment should be able to incorporate the level of information available in the final result [Oberguggenberger, 2012]. There are several ways to refer to uncertainty and can be even categorized [Baecher and Christian, 2003]. Hence, to avoid ambiguity, the term used in this project are epistemic and aleatory uncertainty and the scope each type is the following:

- Aleatory uncertainty: inherent to the assessed process, in this case, the natural variability of properties.

- Epistemic uncertainty: which reflects the lack of knowledge on the assessed event. Here the uncertainty linked to the limited information on input parameters is addressed.

Other sources of uncertainty are beyond the scope of this project.
Peschl [2004] explains these types of uncertainty from the point of view of two observers. If a parameter can behave in a random way (aleatory), its probability is taken to be in a frequency of occurrence in a long series of events. Therefore, if two observers have the same evidence, they should obtain the same value. On the contrary, when uncertainty refers to imperfect knowledge (epistemic), two observes with the same information can get different values of probability.
Regarding geological uncertainty, Baecher and Christian [2003] presented an excellent example to clarify the concept: a die is an example of a randomizing device, in which any information related to the way it is thrown is irrelevant to predict its outcome. Conversely, the outcome of a deck of cards depends entirely from the deck arrangement. If one knows this arrangement, knows certainly the card outcome. Otherwise, additional information is required to predict the outcome. Indeed, the more information, the more accurate the

prediction. This could be taken as an analogy with a geological uncertainty because a specific geological environment has a setup, but engineers and geologists do not know what it is. Exploration is just a way to improve the prediction of geological and geotechnical conditions.

In general sense, there are several ways to account for uncertainty. The selection of a particular approach will be conditioned by the type of modeling, the desired accuracy of the analysis, the type and amount of information related to the input parameters. Some of the available alternatives are [Oberguggenberger, 2012]:

1. Deterministic approach: this is the simplest way to model, in which any parameter X is represented by a single value x. The modeler implicitly considers the uncertainty by the selection of x. However, this approach does not reflect the uncertainty associated with X. This is by far, the most common approach used in Colombia nowadays. Fortunately, this tendency is changing lately; in fact, lately the Colombian Geological Service developed a guideline to perform hazard, vulnerability and risk assessment in slopes involving probabilistic concepts [SGC, 2015]

2. Intervals: in this approach, a parameter X is described by an interval $[x_L, x_U]$ that reflects the variability of the parameters. Nevertheless, it does not give any additional information on its uncertainty. In mechanics, this approach has been applied using the concept of convex intervals [Ben-Haim, 1994, Chen et al., 2017, Qiu et al., 2008, Wang, 2010] and has been extended to solve stability problems of rock wedges in underground excavations [Dong et al., 2017].

3. Probability: this approach gives a more detailed description of the aleatory uncertainty of a parameter X, since it describes the variability and uncertainty of X in terms of probability distributions. Since the 1960ies there has been a geometric growth on the number of publications on this topic. However, it requires enough information to define the probability distribution of the variable.

4. Sets of probability measures: in this case, a single measure of probability is replaced by a set of probability measures. Dempster-Shafer structures

and fuzzy sets are particular prescriptions to obtain sets of probability measures [Oberguggenberger, 2012].

Traditionally, uncertainty has been treated within a probabilistic framework. This approach has some limitations when applying to rock mechanics modeling with discrete elements [Oberguggenberger, 2012]:

- The probability distributions of inputs has to be known, otherwise assumed.

- Techniques like Monte Carlo simulation, aside from inputs probability distributions, require several realizations to compute the probability distribution of the response. This is relevant when complex modeling techniques are applied.

- In rock mechanics, the information sources range from a limited number of field and laboratory results to expert opinion. This sort of information is not suitable for probabilistic analysis unless assumptions are introduced.

- Probabilistic approach can deal with aleatory uncertainty, rather than epistemic.

Therefore, alternative approaches based on interval analysis has appeared in engineering modeling since the end of the 20th century. Within those techniques, Dempster-Shafer evidence theory arises as a promising alternative to probabilistic approach, because it can deal with limited information on input parameters, presented as intervals, without any assumption related to their probability distribution. Moreover, it allows to represent both epistemic and the aleatory uncertainty and propagate it through the engineering model.

2.2.1. General concepts on probability

Before introducing the concept of Dempster-Shafer evidence theory, some general concepts on probability are required.

Firstly, it is important to clarify that concepts are developed for random experiments, but can be applied to phenomena in nature [Nguyen, 2006], and accordingly to rock mass properties. The set of possible outcomes, Ω is defined as a *sampling space*. A *power set* 2^Ω corresponds to the set of all subsets of the sampling space. A map is defined from the power set to the interval [0,1], which assigns a probability, P, to any subset $B \subseteq \Omega$, which is written as $P : 2^\Omega \rightarrow [0, 1]$.

A *probability space* is a triple $(\Omega, \mathcal{H}, P)$, in which $\mathcal{H}$ is a σ-algebra, σ-field *i.e.*:

1. $\Omega \in \mathcal{H}$
2. If $H \in \mathcal{H}$, then $H^c \in \mathcal{H}$
3. If $H_n \in \mathcal{H} \, for \, n \geq 1$, then $\cup_{n \geq 1} H_n \in \mathcal{H}$

A *measurable space* is a pair $(\Omega, \mathcal{H})$ and P: $\mathcal{H} \rightarrow [0, 1]$ is a probability measure, such that: $P(\emptyset) = 0, P(\Omega) = 1, P(\cup_i H^i) = \Sigma_i P(H^i)$, $\forall$ subset H^i in $\mathcal{H}$. The latter is a property referred as σ-additivity of P.

Given this probability space, a random variable X is a map from a set Ω to $\Re$, such that $X^{-1}(\mathcal{B}(\Re) \subseteq \mathcal{H}$ i.e. $\forall B \in \mathcal{B}(\Re))$, $X^{-1}(B) \in \mathcal{H}$

2.2.2. Dempster - Shafer structures (DSS)

Scientific literature refers to Dempster-Shafer structures and random sets to name the same mathematical object. It makes the first approach to the topic very challenging for engineering background researchers. Hence, some clarification is needed.

The concept of Dempster-Shafer structure as presented in this document corresponds to the interpretation of Dempster [1967] of the Belief function as the lower probabilities induced by a multivalued mapping. According to this approach, a multivalued mapping from a probability distribution generates a set-valued random variable, that is a more complex object than the standard random variable.

Likewise, this approach can be seen as the epistemic interpretation of a random finite set, in which a random set is a natural representation of a set of possible random variables, one of which is the actual one. This interpretation

of random sets differs from the original application of random sets to continuous spaces to handle the uncertainty of spatial data, where random sets are precisely located sets or regions[Couso et al., 2014].

This distinction is essential because the semantics of belief functions and random sets is hard to approach, due to the different mathematical interpretations and applications developed around each approach. Accordingly, in order to avoid ambiguity in terminology and to stress the approach followed, the set-valued random variable is referred to as Dempster-Shafer structures, with the acronym DSS. This denomination has been already adopted in several publications [Bernardini and Tonon, 2010, Ferson et al., 2002, Oberkampf et al., 2004a,b, Sentz and Ferson, 2002, Tucker and Ferson, 2006], among others.

The original work developed by Dempster was reinterpreted and generalized by Shafer to develop their Evidence Theory [Shafer, 1976], which building block is the probability distribution on the power set of a finite set Couso et al. [2014], i.e., a Dempster-Shafer structure (DSS).

Shafer [1976] presented both, a theory on evidence and, a theory of probable reasoning. A theory of evidence, because it copes with weights of evidence. Also, a theory on probable reasoning because it allows the combination of evidence. Back then, this combination was carried out according to Dempster's rule. The probable reasoning is one of the most relative advantages of the Dempster-Shafer theory of evidence because uncertainty on rock mass properties is first weighted according to available information and subsequently can be updated as new evidence is available. This will be explicitly addressed in Chapter 6.

Although before Shafer [1976], there were no practical applications of Dempster-Shafer theory, since 1990ies there has been a geometrical growth on the number DST applications [Beynon et al., 2000a]. Some examples of the most recent publications on the topic include measuring uncertainty in big data [Dutta, 2018], multisensor-based activity recognition in smart homes [Al Machot et al., 2018], skin diseases [Khairina et al., 2018], cancer detection [Kim et al., 2018], fault in power transformers [Kari et al., 2018], heritage evaluation[Liu et al., 2018], thermal plants monitoring [Moradi et al., 2018], natural hazard risk assessment [Ballent et al., 2019b], GIS-based risk assessment ap-

plied to hydrocarbon exploration [Seraj et al., 2019], structural reliability assessment [Wang et al., 2019] and chemical risk assessment [Rathman et al., 2018]. These are just a few examples to illustrate how relevant is this approach to handle decision making under uncertainty. Unfortunately, this is not the case in geotechnical engineering, where the technique has a huge potential.

The amount of publication on the geotechnical is by no means, comparable to other fields like AI or medicine. However, there is a growing interest of the civil engineering and geotechnical community in involving the evidence theory in the decision-making process, since its applicability is evident.

The entry paper of the set-valued random variable was presented by Tonon et al. [1996] to predict the rock mass response around a tunnel excavation. Tonon and Bernardini [1999] pursued the problem of optimization of the lining using DSS and fuzzy sets. Subsequently, Tonon et al. [2000a] presented an application to compute rock mass parameters (RMR) of rock masses and some examples. As well as, Tonon et al. [2000b] performed some calculations of plane failure and tunnel lining design with simple explicit models using DSS and Monte Carlo simulation. They also showed the advantages of using the concept of strictly monotonic functions to reduce the number of computations, which reduces computational needs.

A framework to develop a reliability assessment of geotechnical problems by finite element methods and DST was developed by Peschl [2004]. This work presents a comprehensive methodology to perform reliability assessment, in which inputs are expressed as Dempster-Shafer structures, leading to a bounded probability function for the finite element model. This application was named random sets finite element method, RS FEM. Subsequently, related documents have been published with applications of the RS FEM [Schweiger and Peschl, 2004, 2005a,a]. Besides, this method has been applied to the analysis of tunnel excavations [Nasekhian and Schweiger, 2010, 2011].

A similar approach to the one given by Peschl [2004] was selected by Klapperich et al. [2012], Shen and Abbas [2013], but instead of finite elements, distinct elements are applied. In this case, the deterministic reference model is run using the discrete element software UDEC, according to the realizations defined by the combinations of the input Dempster-Shafer structures. With the results, cumulative probability distributions are built.

These are the most relevant publications on DST applied to geotechnical and rock mechanics related problems. Based on these, DST can be applied to more specific problems in geotechnics. For instance, so far, there are no publications on the probable reasoning involved in DST, i.e., on the combination of information from different sources. Besides, although, there are applications in rock mechanics, the problem of modeling spherical data under a framework like this, has not been addressed yet. Besides, solutions to generate alternatives to combine limited information from geomechanical parameters, with more complete sets of information on joint planes orientations have not been studied.

With this framework, the research project explores the possibilities of DST the perform rock slope reliability assessment considering:

1. Rock joint orientation distributed as spherical data with a Kent distribution (See Chapter 3).

2. Rock joint strength parameters as DSS (See Chapter 4).

DSS definition

Before giving a formal description of the Dempster-Shafer structures, some intuitive concepts are presented to introduce the topic. As mention before, the Dempster approach assumed multi-valued mapping from a probability into a set space. This is a natural representation of uncertain information, as presented in the following example.

A dog is hit by a car. One camera reports three cars close to the accident: one blue, one red and one yellow. One witness says that he saw a blue car hitting the dog. However, the witness is drunk 70 % of the time. During that time his information is not considered reliable. Figure **2-1** schematizes the situation.

The set of car colors defines the probability space. Then, there is multi-valued mapping, from this to the sets of car. When the witness is drunk, the set with the three cars is mapped. However, if the witness is sober, the evidence shows that the blue car hit the dog. As a result, the Dempster-Shafer structure with the focal elements and probability assignments listed in Table **2-1**. These concepts are described in the following.

Table 2-1.: Example of multivalued mapping to generate a DSS

Focal elements	Probability Assignment
Blue car, yellow car, red car	0.7 (drunk)
Blue car	0.3 (sober)
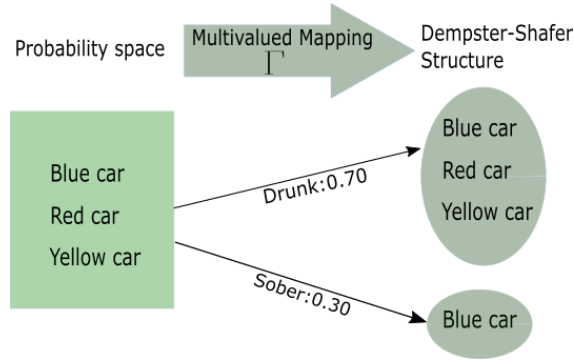
Figure 2-1.: Example of multivalued mapping to generate a DSS

To define a DSS, let suppose that M observations were made on a parameter $u \in U$, each of which resulted in an imprecise (non-specific) measurement given by a set A of values. Let c_i denote the number of occurrences of the set $A_i \in U$, and $\phi(U)$ the set of subsets of U. A frequency function m can be defined such that [Tonon et al., 2000b]:

$$\phi = \{A_i : i = 1...n\} \tag{2-1}$$

$$m : \phi(U) \to [0, 1] \tag{2-2}$$

$$m[\emptyset] = 0 \tag{2-3}$$

$$\sum_{A \in \phi(U)} m(A) = 1 \tag{2-4}$$

ϕ is called the support of the DSS, the subsets A_i are the focal elements and m is the basic probability assignment. A pair (ϕ, m) defines a Dempster-Shafer structure on U. Each set, $A_i \in \phi$, contains possible values of the variable, u, and $m(A)$ is the probability associated to A [Schweiger and Peschl, 2005a]. In order to obtain the basic probability assignment, consider a probability measure, $\phi(z)$, defined on a universal set Z (set of observations) related to

U (Set of values of measurements) through a multivalued mapping $\Gamma : Z \to \phi(U)$, as is shown in Figure **2-2**, for U corresponding to the real line [Tonon et al., 2000a]: Then, the basic probabilistic assignment is:

$$m(A_i = P(z_i)) = \frac{c_i}{M} \tag{2-5}$$
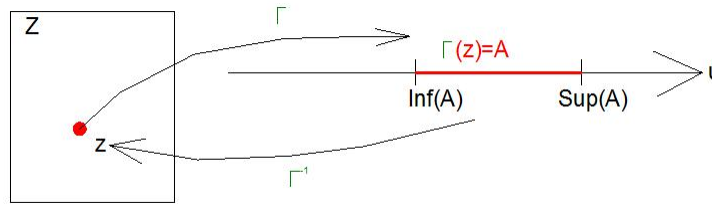
Figure 2-2.: A random set induced by a multivalued mapping defined for a set Z. After Tonon et al. [2000a]

In fact, the probability assignment $m(A_i)$ represents the extent to which all available information supports the claim that a particular element belongs to the set A_i alone and does not imply anything regarding subsets of A_i. Any additional information on subsets of A_i must be represented by another focal element with its corresponding probability assignment [Bernardini and Tonon, 2010].

According to this multivalued mapping, it is not possible to assign a single number to each observation, which means that the nature of the method is imprecise. Hence, it is not possible to calculate the precise probability, *Pro*, of a generic $u \in U$, but only lower and upper bounds of this probability [Schweiger and Peschl, 2005b]. It means that in DST, instead of counting outcomes of the singleton $u \in U$ counts outcomes of subsets $A_i \in U$ and an observation of a subset $A_i \in U$ indicates an event somewhere in A_i, without any specification of the probability that the event be a specific point on A_i, this is shown in Figure **2-3**

When working in the space of real numbers, the support of the DSS is given by intervals, with their corresponding probability assignment. In this chapter, the uncertain parameters are considered as intervals (rock joints mechanical properties), without any specific information related to the distribution or
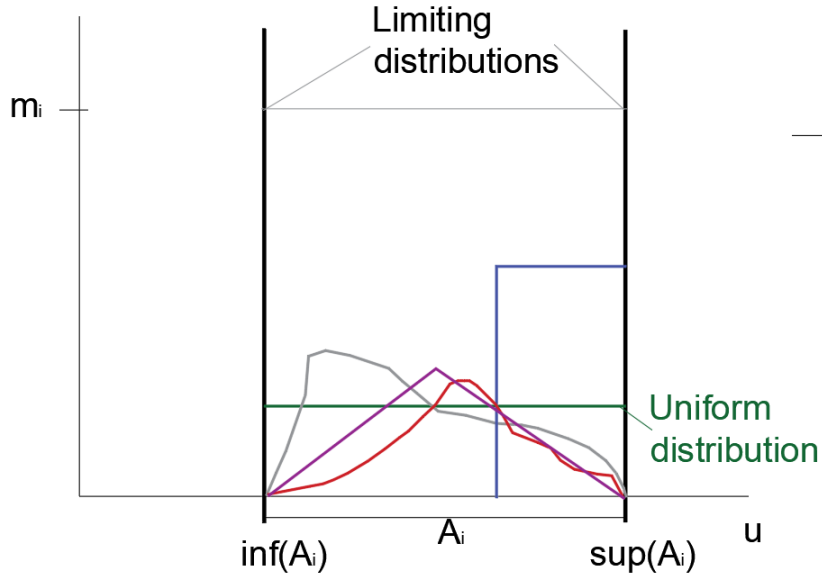
Figure 2-3.: Any distribution probability can be assumed between the upper and lower bound of the ith focal element. After Tonon et al. [2000a]

variation between the extremes of the interval, along with the probability assignment.

As for the upper and lower probability calculation, the exact probability should be somewhere in between the upper and lower bounds of a generic subset E, this is:

$$Bel(E) \leq Pro(E) \leq PI(E) \tag{2-6}$$

Where Bel corresponds to the lower bound and is known as the belief function. Pl corresponds to the upper bound and is known as the plausibility function

The belief function is a set value function which can be calculated through the summation of the basic probability assignment of the subsets A_i included in E, and the plausibility function is a set value function obtained through the summation of basic probability assignments of subsets A_i having a non-zero intersection with E. They are envelopes of all possible cumulative distributions compatible with the data, as can be seen in Figure **2-4** [Schweiger and

Peschl, 2005b].

$$Bel(E) = F_* = \sum_{A_i:A_i \subset E} m(A_i) \tag{2-7}$$

$$Pl(E) = F^* = \sum_{A_i:A_i \cap E \neq \emptyset} m(A_i) \tag{2-8}$$
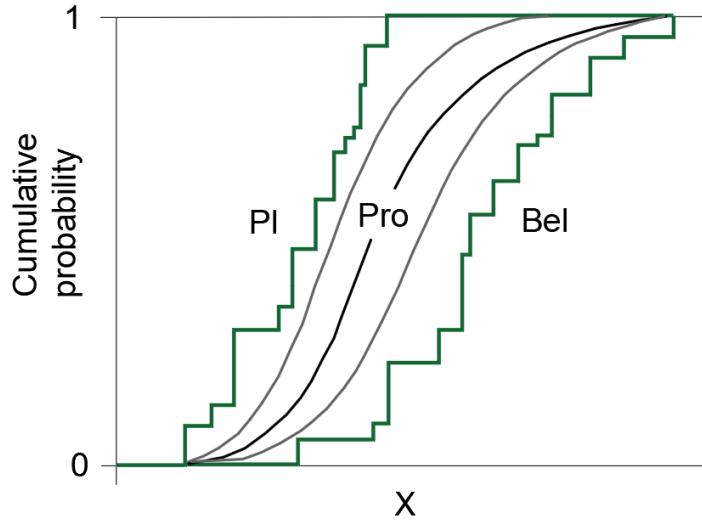
Figure 2-4.: Upper bound (Pl) and lower bound (Bel) on precise probability. After Schweiger and Peschl [2005b]

In other words, to obtain the left envelope (upper bound), the distribution function of each interval in the calculation matrix is assumed to be concentrated at the lower bound of each focal element. On the other hand, for the right envelope (lower bound) the probability mass for each interval is assumed to be concentrated at the upper bound of the interval [Schweiger and Peschl, 2005b]. The bounded distribution function defined by the upper and lower bounds is referred to as BDF or DSS.

The following example illustrates the concept of a DSS. Let assume a layer of sandstone, from which four block samples were taken. From each block, two cores were taken and tested to measure the uniaxial strength, UCS. Table **2-2** summarizes the test results.

Table **2-2**.: Example: UCS results

Sample	UCS 1 (MPa)	UCS 2 (MPa)
Block 1	53.2	57.9
Block 2	49.3	51.8
Block 3	54.3	55.9
Block 4	60.1	66.3

Table **2-3**.: Example: DSS constructed from UCS test results.

Focal elements	Focal element UCS, MPa		Probability assignment
	Upper bound	Lower bound	
1	53.2	57.9	0.45
2	49.3	51.8	0.10
3	54.3	55.9	0.35
4	60.1	66.3	0.10

The design team expects a UCS between 55MPa and 60PMa, since this is the interval specified in technical reports for similar sandstones. Based on this information, their experience and judgment, the design team defined a DSS for the sandstone uniaxial compression, like the one shown in Table **2-3**. Such a DSS has four focal elements, A_i, defined by the strength intervals resulting from each sample. The set of these intervals forms the support of the DSS. Moreover, each focal element has a probability attached, i.e., the probability assignment, m. This probability was assigned considering the expected values and the engineers opinion.

In this example, the results from each block defined the focal elements. However, the selection of the focal elements and probability assignments should be the result of sound engineering judgment.

The bounded probability distribution function (BDF) for the UCS of the sandstone can be constructed according to Eqs. 2-7 and 2-8. Figure **2-5** shows the BDF, where the original focal elements bounds are explicitly marked.
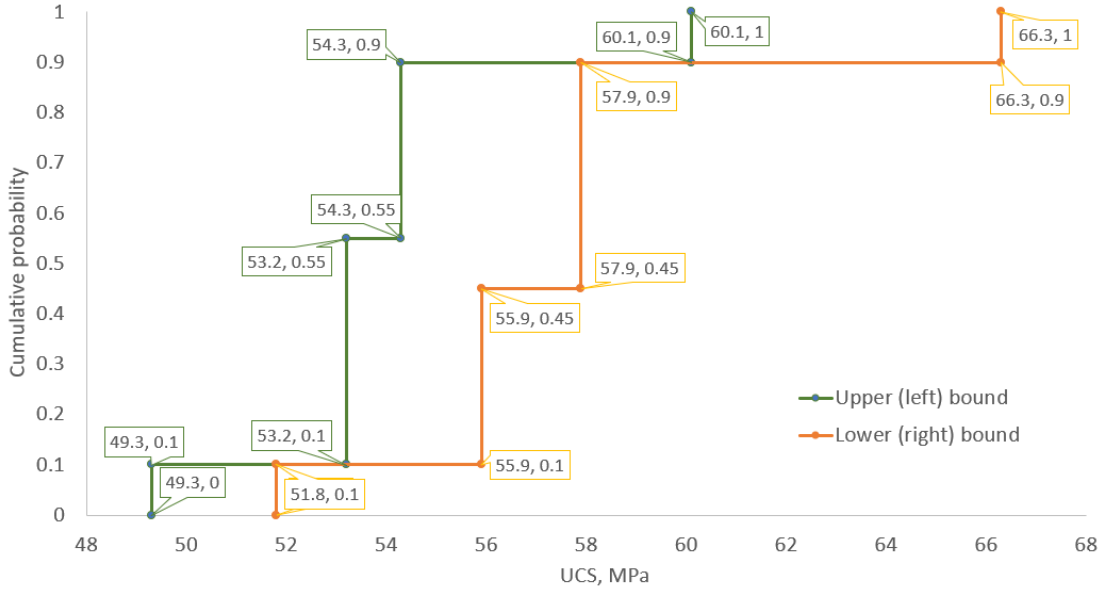
Figure 2-5.: DSS constructed from UCS test results.

Green lines is the upper bound (left) bound of the UCS DSS and orange line corresponds to the lower (right) bound

Bounds of the system response

Once the DSS of input parameters are defined, the next step is propagating the uncertainty, represented as DSS through a model, which means that from the input DSS a DSS of the output is calculated.

Three main tasks should be performed to compute a bounded distribution function (BDF) of the assessed system, i.e., the DSS for the response of the model. Firstly, the image (output) of the input DSS through a function f has to be established. Subsequently, a probability assignment should be assigned to the computed image. Finally, a bounded cumulative distribution function of the output is calculated, according to Eqs. 2-7 and 2-8. Below, a brief description of these steps is presented.

Assume that the random set (R, ρ), which is the image of (T, m) through a function f. The set (R, ρ) is described as follows:

$$R = \{R_j = f(A_i), A_i \in T\} \tag{2-9}$$

$$f(A_i) = \{f(x), x \in A_i\} \tag{2-10}$$

$$\rho(R_j) = \sum_{i=1}^{n} m_i(A_i), A_1 \times \ldots \times A_n \in R \qquad (2\text{-}11)$$

If $A_1, \ldots, A_n$ are sets on $X_1 \times \ldots \times X_n$, respectively, and $x_1, \ldots, \xi_n$ are random independent sets, then the joint probability assignment is measured by:

$$m(A_1 \times \ldots \times A_n) = \prod_{i=1}^{n} m_i(A_i), A_1 \times \ldots \times A_{n \in R} \qquad (2\text{-}12)$$

If the focal element A_i is a closed interval of real numbers: $f(A_i) = \{x/x \in [li, ui]\}$, then the lower $F_*(x)$, and upper, $F^*(x)$, probability mass function, at some point x can be obtained as follows:

$$F_* = \sum_{i:x \geq u_i} m(A_i) \qquad (2\text{-}13)$$

$$F^* = \sum_{i:x \geq u_i} m(A_i) \qquad (2\text{-}14)$$

Assuming that the function $f(A_i)$ is continuous in all $A_i \in T$, and no extreme points exist in this region, except at the vertices, the vertex method applies to calculate the image (R, ρ) of the input random set through the function f. Assume each focal element A_i is an N-dimensional box, whose 2^N vertices are indicated as $v_k, k = 1, \ldots, 2N$. If the vertex method applies then, the lower and upper bounds R_{j*} and R_j^* on each element $R_j \in R$ will be located at one of the vertices [Schweiger and Peschl, 2005b]:

$$R_{j*} = min\{f(v_k) : k = 1, \ldots, 2^N\} \qquad (2\text{-}15)$$

$$R_j^* = max\{f(v_k) : k = 1, \ldots, 2^N\} \qquad (2\text{-}16)$$

Thus, the function $f(A_i)$ has to be evaluated 2^N times for each focal element A_i. The number of realizations, n_c, required for finding the bounds of the system response is [Schweiger and Peschl, 2005b].

$$n_c = 2^N \prod_{i=1}^{N} n_i \tag{2-17}$$

Where N is the number of basic variables and n is the number of information sources available for each observation. Nevertheless, the number of calculation can be reduced if the function is strictly monotonic with respect to each parameter. In that case, it is possible to predict which combination or nodes of the p-dimensional box will give the maximum and minimum output, prior to the calculation.

The general procedure is summarized in Figure **2-6**, in which input variables expressed as DSS that go through a calculation model and as a result a DSS of the model response is obtained. This DSS is a bounded distribution function of the response, named as BDF in this document. The BDF is equivalent to DSS.
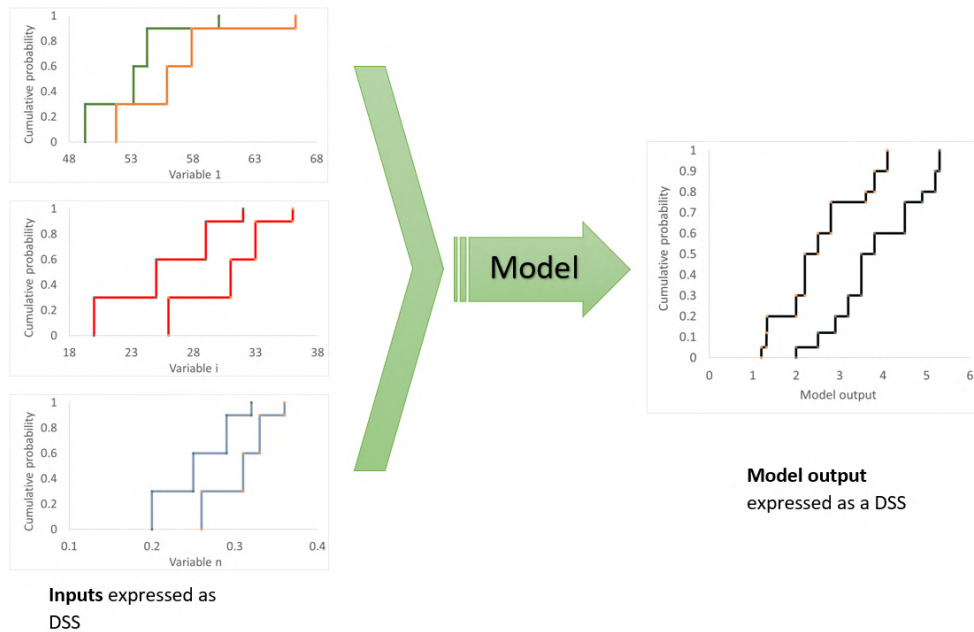
Figure 2-6.: Process to compute the bounded probability function of the system response, based on Dempster Shafer structures

The concepts presented in this chapter will be applied to represent and propagate the mechanical information on joint parameters through a wedge sta-

bility model in Chapter 4). Moreover, this approach will be combined with discrete fracture networks (DFN) in Chapter 5.

3. Stability Analysis of Rock Wedges with Kent Distribution

3.1. Introduction

A systematic and reproducible technique for collecting information is required to gather reliable information on rock mass structure. In this regard, this project introduced the use in Colombia of a short range stereoscopy technique to measure the geometrical features of the rock mass. This tehcnique was implemented in a sandstone quarry in Une, Cundinamarca as described in Appendix A.

The approach does not eliminate the variability of joint planes orientation. But, allows to collect more robust data sets to describe properly such a variability, as shown in this chapter.

A crucial step in the reliability analysis of rock wedges is a proper characterization of the variability of dip and dip direction of planes. This is a complex task, considering that planes are described as sets of unit vectors. Hence, traditional real-valued probability distributions are not suitable to represent this sort of information.

Directional statistics provide a framework to accurately represent vectors, in which distributions are described on surfaces, such as circles, spheres, or torus. The Fisher distribution is a well-known spherical distribution, analogous to the symmetric Gaussian distribution, wrapped around a unit sphere [Kasarapu, 2015], which means that it has circular contours of constant probability density. The Fisher distribution is commonly used to perform reliability assessment in rock slopes, and more general distributions have received little attention in rock mechanics for evaluating the stability of rock slopes.

However, the Gaussian symmetry condition is not always the best assum-

ption. To illustrate this, Figure **3-1** depicts a set of poles collected in a sandstone quarry in Une, Cundinamarca, Colombia. A simple visual inspection shows that at least the sets defined by green and red poles are not symmetrically distributed about the mean. Here is evident the need for using a more flexible distribution over the Fisher distribution.
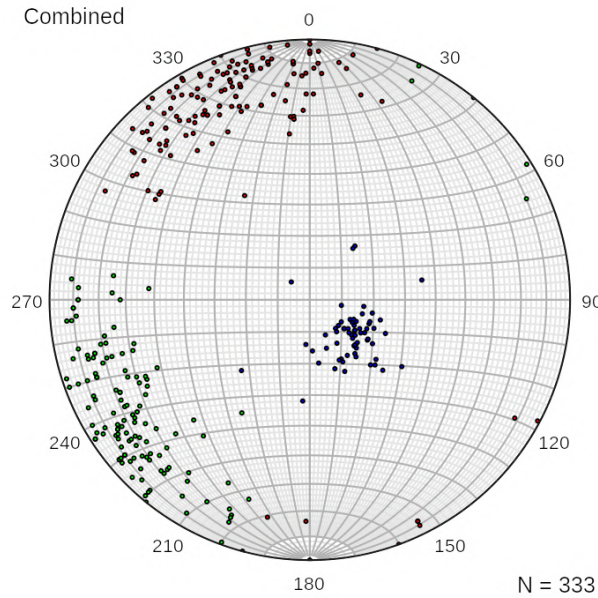
Figure 3-1.: Structural information collected in El Pedregal Mine in 1997. Three joint sets can be clearly identified

The Fisher-Bingham family distribution provides such a flexible framework, particularly the 5-parameters Fisher-Bingham, referred to as Kent distribution. Nevertheless, the complexity for estimating the five parameters and subsequent simulation have diminished the application in rock slope stability assessment [Park, 2000].

As for the structure of the chapter, firstly, a theoretical framework is presented, including the selected approaches to estimate and simulate the Kent distribution. This background is included because it is not a widespread topic in rock mechanics, and because the algorithms were programmed and incorporated in the subsequent reliability assessments. Then, the proposed algorithms to compute the probability scenarios under different conditions are introduced. Afterward, some examples illustrate those algorithms. The

chapter closes with conclusions on the implications of using the Kent distribution

3.2. Modeling orientation of joints

Modeling orientation data is part of a growing discipline of statistics that has applications on earth sciences, meteorology, medicine, physics, and biology [Kasarapu, 2015]. The concept is to define a distribution for a random variable $z \in \mathcal{R}^d$ conditioned to lie on, or projected onto the unit sphere [Paine et al., 2017]. Then, the cloud of orientations measured for a given joint set from Figure **3-1** are expressed as a 3D vector lying on the unit sphere and fitted to such distribution.

In the case of the general Fisher-Bingham distribution, a multivariate normal distribution is conditioned to lie on the unit sphere. Hence, the probability density function f at any point $\mathbf{x}$ on the sphere takes the form [Mardia, 1975]:

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) \propto exp\left\{\kappa\boldsymbol{\gamma}_1^T\boldsymbol{x} + \beta_2(\boldsymbol{\gamma}_2^T\boldsymbol{x})^2 + \beta_3(\boldsymbol{\gamma}_3^T\boldsymbol{x})^2\right\} \tag{3-1}$$

where $\boldsymbol{\Theta}$ is a parameters vector, the parameters $\boldsymbol{\gamma_1}$, $\boldsymbol{\gamma_2}$ and $\boldsymbol{\gamma_3}$ are unit vectors, and β_2 and β_3 are real values. To formulate this distribution, 8 parameters are required, thus, it is known as the 8-parameters Fisher-Bingham distribution, FB_8. However, there are difficulties to estimate and interpret the 8 parameters of this distribution[Kent, 1982]

This is because Kent [1980, 1982] suggested the alternative 5-parameters Fisher-Bingham (FB5) distribution. For the purpose of this document, the FB5 will be referred to as the Kent distribution. This distribution is an special case of the FB_8 and is obtained by assuming that $\boldsymbol{\gamma_1}$, $\boldsymbol{\gamma_2}$, $\boldsymbol{\gamma_3}$ from a orthogonal system of vectors and are constrained by $\beta_2 = -\beta_3 = \beta$ [Kasarapu, 2015]. Based on this, the distribution takes the form:

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = c(\kappa, \beta)exp\left\{\kappa\boldsymbol{\gamma}_1^T\boldsymbol{x} + \beta\left[(\boldsymbol{\gamma}_2^T\boldsymbol{x})^2 - (\boldsymbol{\gamma}_3^T\boldsymbol{x})^2\right]\right\} \tag{3-2}$$

In which parameters are interpreted as:

$\gamma_1, \gamma_2, \gamma_3$ orthogonal unit vectors that correspond to the mean, major and minor axis, respectively. κ and β are shape parameters and represent the concentration and the ovalness, respectively. The eccentricity, λ, is computed as $2\beta/\kappa$.

This spherical distribution is equivalent to the general Gaussian distribution, which means that it has an ellipse like iso-probability contours on the sphere. Consequently, the Kent distribution is not analogous to a symmetric Gaussian distribution as Fisher is. Figure **3-2** shows the variation of the iso-probability contours as the parameters κ and β change, where the density decreases from the center outwards. The size of the circle/ellipse represents the concentration of the data. Therefore, the higher κ and β, the more elongated the ellipse.

When $\beta = 0$, Kent distribution becomes the Fisher distribution [Fisher, 1953]. Its application for modeling joints orientation is widespread, given its simplicity since it depends only on the concentration parameter and the mean [Park, 2000, Park and West, 2001, Park et al., 2005, Priest, 1993]. In fact, the popular software Swedge from Rocscience implements the Fisher distribution for simulating joints orientation.

Despite its simplicity, the Fisher distribution is analogous to a symmetric Gaussian distribution, which means that iso-probability contours on the sphere are circles rather than ellipses. This assumption can be inaccurate in many cases, as illustrated in Figure **3-1**, where red and green poles would be represented more appropriately by ellipse like contours than circles. This chapter presents the implications of assuming a Kent distribution, over a Fisher distribution, on the probability function of a wedge factor of safety.

Given the flexibility of the Kent distribution, it has been used to model sets of data in different fields such us describing protein chains [Boomsma et al., 2006, Hamelryck et al., 2006, Kasarapu and Allison, 2015, Kent, 2012, Kent and Hamelryck, 2005], machine learning in bioinformatics [Hamelryck, 2009], and image processing [Lunga and Ersoy, 2011].

As for rock mechanics applications, the original work [Kent, 1980, 1982] estimates Kent parameters for directions of magnetism. Besides, McLachlan and Peel [1999], Peel et al. [2001], Yamaji [2016] utilized this distribution to identify rock joint sets by clustering. Other than these, the application of Kent distribution has not been extensively applied in rock mechanics to model and
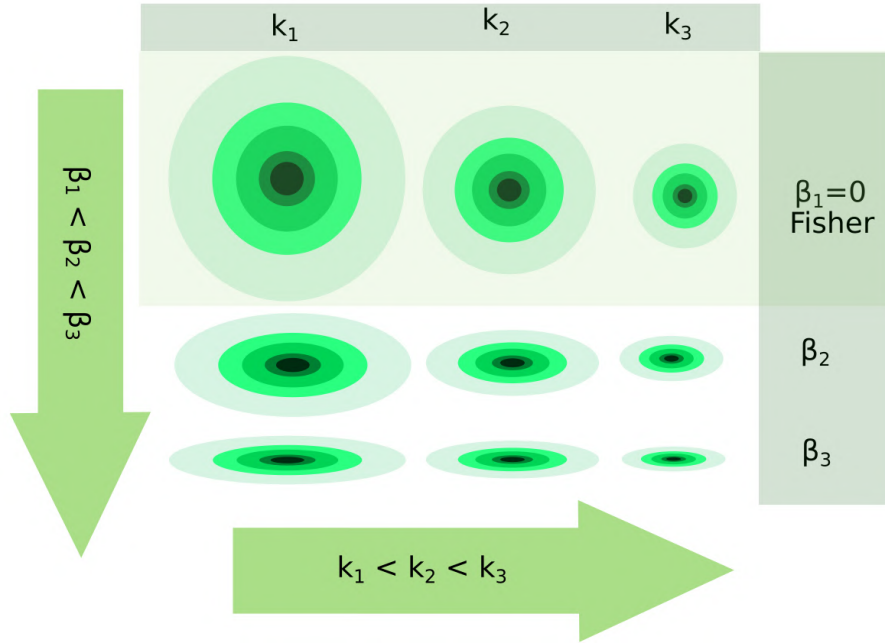
Figure 3-2.: Kent distribution shape for different values of eccentricity

simulate sets of joints orientations. Even though Kent distribution has been used to characterize rock joints sets, there are no publications that address the assessment of the stability of rock slopes modeling the orientation with this distribution.

3.2.1. Kent distribution parameters estimation

There are several alternatives to estimate the five parameters of the Kent distribution such us, maximum likelihood estimates (MLE), Bayesian inference using minimum message length (MML) [Kasarapu, 2015]. Both MLE and MML require an iterative process for parameters estimation.

On the other hand, Kent [1982] proposed the moment estimates as an alternative approximate method for estimating the five parameters of the Kent distribution. This approach has the following properties :

1. The moment estimates are consistent estimates of the true parameters.

2. The orientation matrix can be calculated explicitly.

3. If the eccentricity is small or the κ is large, the moment estimates are closed to MLE.

4. If the concentration is large, the concentration parameters can be computed explicitly.

The moment estimates was selected for estimating the five parameters of the Kent distribution, not just because it is explicitly formulated, which simplifies its implementation and reduces the linked computational cost: but, also, because it can also be used as an alternative for mixture construction [Peel et al., 2001] (See chapter 6). For the sake of clarity, in the following, a description of the method is presented.

The parameters from the Kent distribution are three orientation parameters $\gamma_1, \gamma_2, \gamma_3$ and two concentration parameters. For computing the orientation parameters (they are orthogonal), the idea is to align the reference coordinate system as depicted in Figure **3-3**. First, a rotation about X_1 takes γ_1 to plane X_1, X_2. Then, a rotation about X_3 aligns γ_1 with X_1. The process is repeated until axes are fully aligned. A detailed description of this concept is presented in Kasarapu [2015].
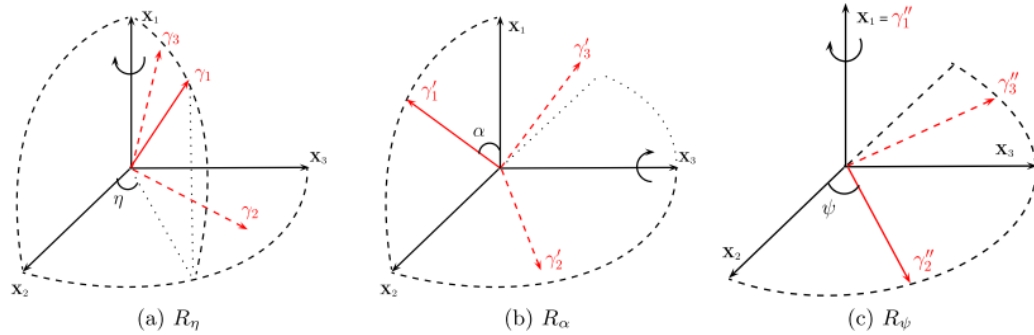
Figure **3-3**.: Ordered axis rotation to align data with reference coordinate system. Adapted from Kasarapu [2015]

The general aim is to compute the five parameters of Kent distribution, having a set of measured joint planes expressed as the dip and dip direction, as one of the three sets shown in Figure **3-1**. The computation algorithm is depicted in Figure **3-4** and follows the procedure described by Kasarapu [2015], Kent [1980, 1982].

Once planes are expressed in terms of colatitude (γ_2) and longitude (γ_1), the unit cosines are computed for each plane as follows:

$$x_1 = \cos(\gamma_2) \qquad x_2 = \sin(\gamma_2)\cos(\gamma_1) \qquad x_3 = \sin(\gamma_2)\sin(\gamma_1) \tag{3-3}$$

Then, the sample mean $\bar{x}$ and dispersion matrix S are computed:

$$\bar{x} = \frac{1}{N}\sum_{i+1}^{N} x_i \tag{3-4}$$

$$S = \frac{1}{N}\sum_{i+1}^{N} x_i x_i^T \tag{3-5}$$

Assuming that $\tilde{\kappa}, \tilde{\beta}, \tilde{G} = (\gamma_1, \gamma_2, \gamma_3)$ are the moment estimates of κ, β, G. The normalization of $\bar{x}$ yields the moment estimate $\tilde{\gamma_1}$ and the diagonalization of S return the moment estimates $\tilde{\gamma_2}$ and $\tilde{\gamma_3}$ [Kasarapu, 2015]. Hence, the procedure to compute $\tilde{G}$ is:

1. Compute the rotation matrix $\mathbf{H}$, about axis X_1 and X_3.

$$\mathbf{H} = \begin{bmatrix} \cos\gamma_2 & -\sin\gamma_2 & 0 \\ \sin\gamma_2\cos\gamma_1 & \cos\gamma_2\cos\gamma_1 & -\sin\gamma_1 \\ \sin\gamma_2\sin\gamma_1 & \cos\gamma_2\sin\gamma_1 & \cos\gamma_1 \end{bmatrix} \tag{3-6}$$

2. Transform the dispersion matrix $\mathbf{S}$, this defines the matrix $\mathbf{B}$

$$\mathbf{B} = \mathbf{H}^T \mathbf{S} \mathbf{H} \tag{3-7}$$

3. Compute angle ψ between the transformed matrix $\mathbf{B}$ and X_1. This is accomplished by diagonalizing the lower 2×2 submatrix of $\mathbf{B}$.

$$\tan 2\psi = \frac{2b_{23}}{b_{22} - b_{33}} \tag{3-8}$$

4. Define the transformation matrix $\mathbf{K}$, to rotate and angle ψ.

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$
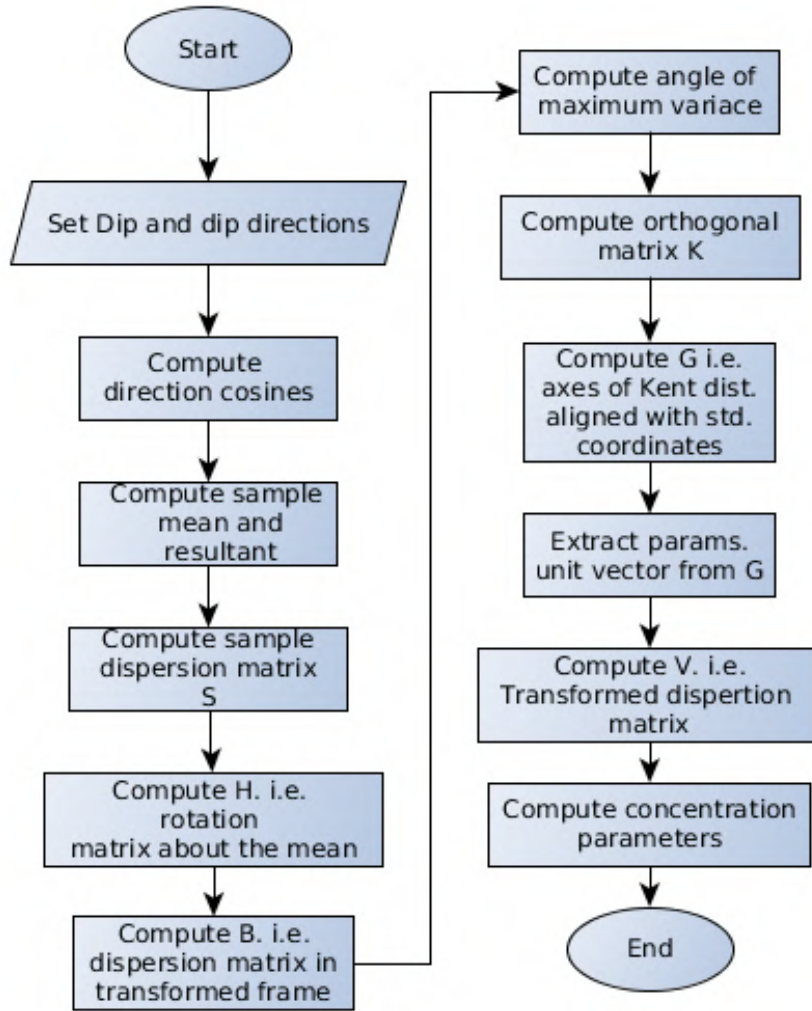
Figure 3-4.: Algorithm to estimate Kent distribution parameters from a cloud of poles

5. Calculate the matrix $\tilde{G}$ that is the moment estimate of $(\gamma_1, \gamma_2, \gamma_3)$. This is also the transformation matrix that aligns the axes of FB_5 with the reference coordinate system.

$$G = HK = (\gamma_1, \gamma_2, \gamma_3) \tag{3-9}$$

6. Rotate the dispersion matrix by applying the transformation matrix $\tilde{G}$.

$$V = \tilde{G}^T S \tilde{G} \tag{3-10}$$

$$W = v_{22} - v_{33} \tag{3-11}$$

7. Compute moment estimates $\tilde{\kappa}$ and $\tilde{\beta}$.

$$\tilde{\kappa} = (2 - 2\bar{R} - W)^{-1} + (2 - 2\bar{R} + W)^{-1} \tag{3-12}$$

$$\tilde{\beta} = \frac{1}{2}\left[(2 - 2\bar{R} - W)^{-1} - (2 - 2\bar{R} + W)^{-1}\right] \tag{3-13}$$

where $\bar{R}$ is the resultant of input planes.

3.2.2. Kent distribution simulation

Estimation of input parameters is the first step to carry out the simulation. Once these parameters have been established, a random sampling on the distribution can be performed. Again, this is not straightforward, because of the complexity of the function, which makes that an inverse transformation cannot be applied. This is because, a simulation technique based on the acceptance-rejection, A/R, algorithm was selected to simulate samples from a Kent distribution. The A/R is an efficient simulation algorithm. This approach has been used for simulating the Fisher-Bingham distribution [Ganeiber, 2012, Kent, 2012, Kent et al., 2004, 2017, Kent and Hamelryck, 2005, Wood, 1987].

An explanation of the algorithm in simple words is as follows. When it is not possible to simulate a probability distribution f, but it is possible to find an alternative probability function G, with a density function g, for which an efficient simulation algorithm is available. Then, the density function f can be simulated from g. Density functions f and g may be expressed as:

$$f(x) = c_f f^*(x) \qquad\qquad g(x) = c_g g^*(x) \tag{3-14}$$

where c_f and c_g are normalizing constants that can be unknown. However, it is not necessary to know those constants to simulate f.

The algorithm is built based on the next assumption [Ganeiber, 2012]:

$$f^*(x) = M^* g^*(x) \tag{3-15}$$

in which M^* is a known bound. With this assumption, the density function f is simulated as follows:

If $M = f(x)/g(x) > 1$, the efficiency of the algorithm is M^{-1}. From this, two important conclusions can be drawn. Firstly, the closer the auxiliary function g to f, the more efficient the algorithm is. Secondly, to assess the efficiency of the algorithm, the normalization constants have to be known. In addition, the ratio $h(x) = f(x)/(Mg(x))$ has to be maximized in order to maximize the efficiency of the algorithm. A demonstration of the algorithm can be found in Ganeiber [2012], among others.

This general approach was followed to simulated the Kent distribution, f, using a truncated exponential envelope as bounding function, g [Kent, 2012], as shown in Figure **3-5**. This function can be expressed in terms of parameters σ and τ, for any x:

$$\frac{1}{2}(\sigma|x| - \tau)^2 \geq 0 \qquad\qquad (3\text{-}16)$$
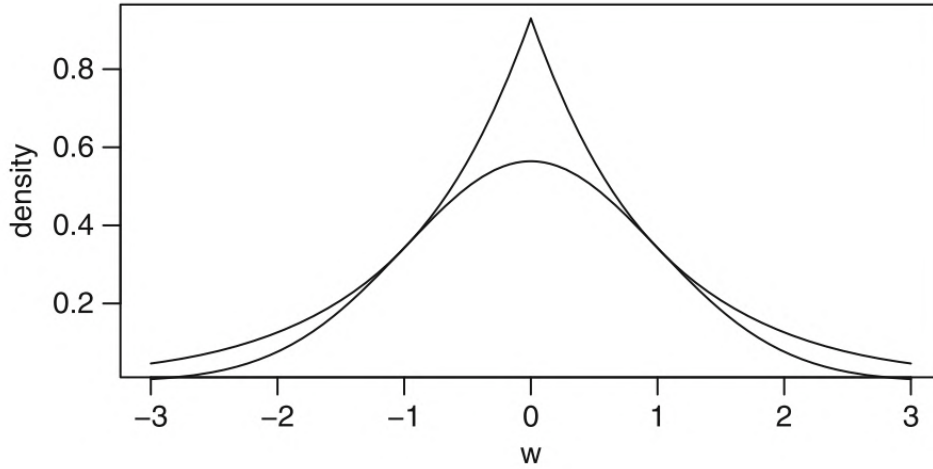
Figure 3-5.: Truncated exponetial envelope to simulate the Kent distribution [Kent, 2012]

Eq3-16 is parametrized in terms of the variables t_1 and t_2. Subsequently, they will be related to Kent distribution variables and parameters.

In the following, the procedure developed by Kent [2012] to parametrize Eq. 3-16 is applied twice. First, $\sigma = \gamma^{1/2}$, $\tau = 1$ and $x = t_1^2$, and then $\sigma = (a + 2\gamma^{1/2})^{1/2}$, $\tau = 1$ and $x = t_1$. This leads to:

$$-\frac{1}{2}(at_1^2 + \gamma t_1^4) \leq \frac{1}{2} - \frac{1}{2}(a + 2\gamma^{1/2})t_1^2 \leq c_1 - \lambda_1|t_1| \qquad\qquad (3\text{-}17)$$

where

$$c_1 = 1 \qquad\qquad \lambda_1 = (a + 2\gamma^{1/2})^{1/2} \qquad\qquad (3\text{-}18)$$

In similar manner, Eq. 3-16 is parametrized in terms of t_2, as follows:

$$-\frac{1}{2}(bt_1^2 + \gamma t_2^4) \le -\frac{1}{2}(b - \gamma)t_2^2 \le c_2 - \lambda_2|t_2| \qquad\qquad (3\text{-}19)$$

where

$$c_2 = \frac{B}{2(b - \gamma)} \qquad\qquad \lambda_2 = b^{1/2} \qquad\qquad (3\text{-}20)$$

In order to relate this parametrized function with the Kent distribution, Eq. 3-2 has to be modified to be expressed as a Gaussian distribution. Under high concentration, the FB_5 has a bivariate normal distribution [Kent, 2012], concentrated near $x_1 = 1$, $x_2 = 0$ and $x_3 = 0$. Considering this and with the following change of variable:

$$t_1 = \sin\frac{\gamma_2}{2}B\cos\gamma_1 \qquad\qquad t_2 = \sin\frac{\gamma_2}{2}B\sin\gamma_1 \qquad\qquad (3\text{-}21)$$

the Kent distribution in (t_1, t_2) coordinates takes the form:

$$f(t_1, t_2) = \exp -\frac{1}{2}[at_1^2 + \gamma t_1^4 + bt_2^2 - \gamma t_2^4] \qquad\qquad (3\text{-}22)$$

in which

$$a = 4\kappa - 8\beta \qquad\qquad b = 4\kappa + 8\beta \qquad\qquad c = 8\beta \qquad\qquad (3\text{-}23)$$

Eq. 3-22 shows that Kent distribution can be expressed as two functions, one in terms of t_1 and the other in terms of t_2. Hence, the simulation will be performed separately. However, since the points are conditioned to lie within the unit disc, the condition $t_1^2 + t_2^2 < 1$ is checked for each simulated pair. The procedure for simulating t_1 is sketched below.
First, Eq. 3-17 is exponentiated to obtain:

$$\exp\frac{1}{2} + \gamma t_1^4 \le \exp c_1 * \exp -\lambda_1|t_1| \qquad\qquad (3\text{-}24)$$

From which

$$f(t_1) = \exp\left(\frac{1}{2} + \gamma t_1^4\right) \qquad g(t_1) = \exp\left(-\lambda_1 |t_1|\right) \tag{3-25}$$

Based on the formulation above, Figure **3-6** shows the algorithm to perform a simulation of random orientations from a Kent distribution. The algorithm was programmed in Python. The input parameters correspond to the concentration of κ and ovalness β for each joint set. In the following, a description of the algorithm is presented:

1. Compute constants related to κ and β:

   a) Eccentricity

   b) a, b and γ from Eq. 3-23

   c) c_1, λ_1 from Eq. 3-18 and c_2, λ_2 from Eq. 3-20

2. Maximize function $h_i(t_i) = f_i(t_i)/(M_i g_i(t_i))$. This means, calculating the values t_i that solve the cubic equation resulting from equalizing the derivative of $h_i(t_i)$ with respect to t_i to zero.

3. Iterative process repeated until a simulation of size N is obtained.

   a) j=0

   b) Simulate $U_i \sim U(0,1)$

   c) Compute $t_i = \ln|U_i|/\lambda_i$

   d) Calculate $f(t_i)$, $gt_i)$ from Eq. 3-25 and $h(t_i)$

   e) Independently simulate $U_i' \sim U(0,1)$

   f) If $U_1' \leq h(t_1)$ and $U_2' \leq h(t_2)$

      ■ If yes: accept t_1 and t_2 and go to next step

      ■ If not: go back to step b)

   g) Compute longitude and colatitude

   h) Update $j = j + 1$

   i) Repeat until $j > N$
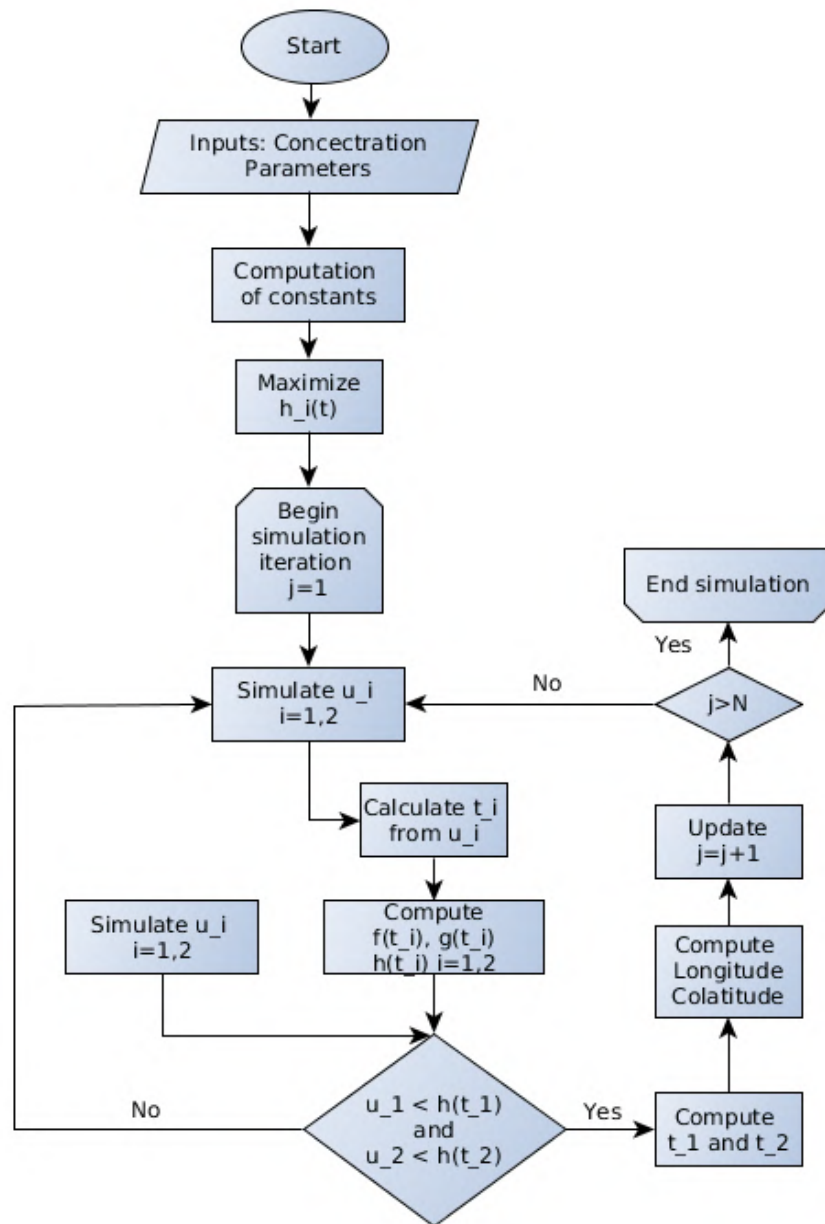
Figure 3-6.: Algorithm to simulate a sample of N planes with a Kent distribution by an A/R algorithm

3.3.   Probabilistic analysis of stability

This section presents a framework to evaluate the effect of modeling the orientation of planes in the stability of a rock slope. This proposal has the following features:

1. Probabilistic method: Monte Carlo simulation. Hence, several realizations of the deterministic stability models are required

2. Deterministic model: Wedge failure. This mechanism of failure is frequent in rock slopes and is widely accepted within rock mechanics practitioners. Moreover, the model is complex enough to account for the geometry of for different planes, two joint sets, and two joint planes. Hence, it is suitable to assess the impact of considering the Kent distribution.

3.3.1. Deterministic wedge stability model

The geometry of the wedge analyzed in this chapter is defined by four planes, two discontinuities and two free surfaces (slope face and upper slope), as shown in Figure **3-7**. The stability assessment of wedges implies the formulation of equilibrium of forces in a three-dimensional space. This problem has been solved by using stereographic projections [Goodman and Taylor, 1966, Hoek et al., 1973]. However, when reliability assessment is performed, an explicit solution is more suitable, because several realizations of the model have to be computed. Low [1979] developed a closed-form equation without using graphical assistance. Subsequently, Low [1997], Low and Einstein [1992] extended the equation to an inclined upper slope. This solution has been applied to perform wedge failure reliability assessment [Jimenez-Rodriguez and Sitar, 2006, Low, 1997, Low and Einstein, 1992]. Other solutions have been applied to perform a probabilistic analysis of wedges [Park, 2000, Park and West, 2001, Park et al., 2005], involving directional statistics, but assuming Gaussian symmetry through the Fisher distribution.

The solution proposed by Low was selected to account for the uncertainty in this chapter since it is convenient to perform several realizations of the model. A brief description of this proposal is given below.

Figure **3-7** shows the geometry of a potentially unstable wedge. The model can analyze four modes of failure in this sort of wedges: failure along the intersection of both joint planes, failure along either joint plane 1 or plane 2, and lifting failure.
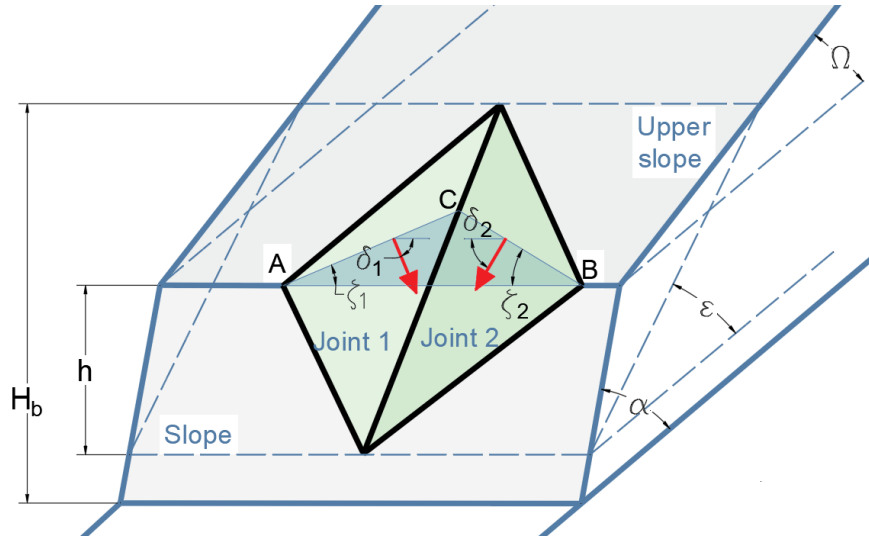
Figure 3-7.: Tetrahedral wedge model [Low, 1997], adapted from Jimenez-
Rodriguez and Sitar [2006]

To illustrate the model, the wedge factor of safety is computed according to
Eq.3-26, when the failure occurs along the intersection line.

$$F_s = \left(a_1 - \frac{b_1 G_{w1}}{s_\gamma} \right) * \tan \phi_1 + \left(a_2 - \frac{b_2 G_{w2}}{s_\gamma} \right) * \tan \phi_2 + 3b_1 \frac{c_1}{\gamma h} + 3b_2 \frac{c_2}{\gamma h} \quad (3\text{-}26)$$

where, ϕ_1 and ϕ_2 are the friction angles of planes 1 and 2, respectively
c_1 and c_2 correspond to cohesion of planes 1 and 2, respectively
G_w water pressure coefficient of a pyramidal distribution.
h is the slope height, and H is the total height, including the upper slope
a_1, a_2, b_1 and b_2 are geometric coefficient, function of the following angles (for
a detailed description see [Low, 1997])
ζ_1 is the horizontal angle between the strike line of plane 1 and the line of
intersection between the upper ground surface and the slope face. Idem for
ζ_2 in plane 2 [Jimenez-Rodriguez and Sitar, 2006]
δ_1 and δ_2 are the dip angle of planes 1 and 2, respectively.
Eq.3-26 is straightforwardly programmable, it resorts to angles ζ_1, ζ_2, δ_1 and
δ_2. Low [1997] presents a graphical approach to compute these angles, based
on the strikes of joint and slope planes. Low [1997] applies the model to a
wedge that fails along the intersection line. However, there is no reference to

a systematic way to compute those horizontal angles, which is crucial, since they define the mode of failure.

For this project, a broad range of possible combinations of joint and slope planes are evaluated, considering the planes as a random variable Kent-distributed. Hence, different modes of failure are expected (along intersection lines, plane 1, plane 2 or lost on contact). Moreover, millions of realizations of the model are carried out. Hence, a programable algorithm for computing those horizontal angles is necessary to perform such probabilistic analysis. Accordingly, an algorithm to compute angles ζ_1, ζ_2, δ_1, and δ_2 is proposed in this section, based on the trend of the planes. The angles are obtained from the construction of the horizontal triangle BCD plotted in Figure **3-8**. Figure **3-9** shows a plan view of such a triangle, in which target angles are shown.
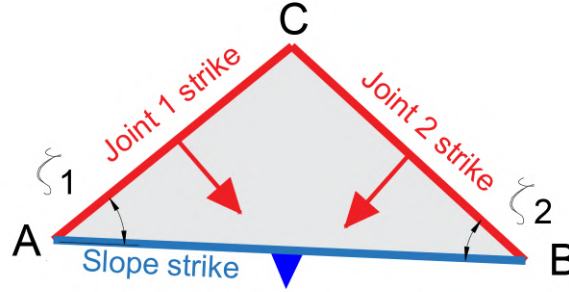
Figure 3-8.: Horizontal angles defined by Low's wedge model

The first step is to generate a reference set-up of planes, which applies to all possible relative orientation of input planes. The proposed procedure to find this arrangement is the following:

1. Define two horizontal vectors dd_i (*plunge* $= 0$) oriented as the dip direction of joint planes

$$dd_i = (\alpha_i, 0) \tag{3-27}$$

2. Define the horizontal vector t_i and express it according to their direction cosines

$$t_i = (\sin(\alpha_i), \cos(\alpha_i), 0) \tag{3-28}$$
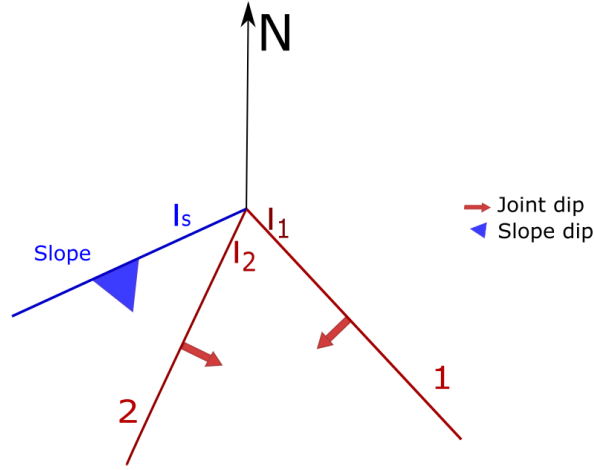
Figure 3-9.: Plane triangle. Joints and slope planes set-up

3. Order joint planes in such a way that third component of the cross product is positive. This condition guarantees that a unique order assignation for joint planes.

$$t_1 \times t_2 = (0, 0, k) \qquad \forall t_1, t_2 \backslash k > 0 \qquad (3\text{-}29)$$

4. Then, considering that planes are vectors, rather than axis, a convenient definition of the orientation of joint planes (ι_1, ι_2) and the slope (ι_s)

$$\iota_1 = \alpha_1 - 90 \qquad \iota_2 = \alpha_2 + 90 \qquad \iota_s = \alpha_2 + 90 \qquad (3\text{-}30)$$

With the described algorithm, a set-up such as the one plotted in Figure **3-9** is obtained. With this arrangement, the angle between joint planes is always less than 180°, which generates four regions where the slope line may lie, as depicted in Figure **3-10**. The regions correspond to:

1. Region I: Wedge slides along plane 1.

2. Region II: Wedge slides along plane 2. $\iota_1 + 180 < \iota_s < \iota_2 + 180$

3. Region III: wedge slides along intersection of planes. $\iota_2 < \iota_s < \iota_2 + 180$

4. Region IV: No sliding wedge is formed. $\iota_2 + 180 < \iota_s < \iota_1$
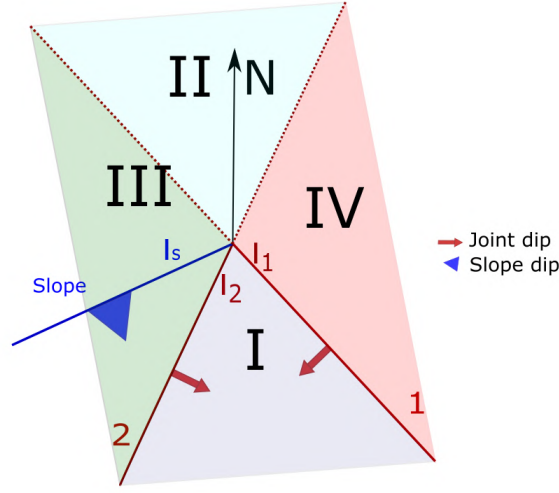
Figure 3-10.: Plane triangle. Regions defined by joints combination

With the convention defined, the algorithm consists of identifying within which region ι_s falls, and then computing the angles ζ_1, ζ_2, δ_1 and δ_2. This task requires some additional refinement, depending on within what quadrants ι_1 and ι_2 are located. As an example, the results for $0 \leq \iota_1 < 180$ and $0 \leq \iota_2 < 180$ are presented below.

1. Region I: If $\iota_1 < \iota_s < \iota_2$, then:

$$\zeta_1 = \iota_s - \iota_1 \qquad \zeta_2 = \iota_2 - \iota_s \qquad \delta_1 = dip_1 \qquad \delta_2 = dip_2 \qquad (3\text{-}31)$$

2. Region II: If $\iota_1 + 180 < \iota_s < \iota_2 + 180$, then:

$$\zeta_1 = \iota_s - (\iota_1 + 180) \zeta_2 = (\iota_2 + 180) - \iota_s \delta_1 = dip_1 \delta_2 = 180 - dip_2 \quad (3\text{-}32)$$

3. Region III: If $\iota_2 + 180 < \iota_s < \iota_1 + 180$, then:

$$\zeta_1 = (\iota_1 + 180) - \iota_s \qquad \zeta_2 = \iota_s - \iota_2 \qquad \delta_1 = dip_1 \qquad \delta_2 = dip_2 \qquad (3\text{-}33)$$

There are some commercial alternatives to compute the stability of wedges, such as Swedge de Rocsicence (available at rocscience.com). However, none of the available alternatives allows the simulation of planes with a Kent distribution. Besides, this project requires the assessment of a large variety of

scenarios. Hence, a flexible way to input the information is required, which is not available in commercial software. Finally, none of these commercial software has the possibility of considering the input data as intervals, which will be useful in subsequent chapters.

Considering the limitations of available software, in this project, the Low's approach to compute the wedges' stability was programmed in Python, the program includes the geometrical algorithm proposed to compute the involved angles. Besides, the script can be conveniently called from the adopted probabilistic approach. The module is flexible enough to handle different types of inputs, including deterministic and probabilistic, for both conventional probability distributions and directional data.

3.3.2.   Removable and unstable wedges

Two conditions have to be fulfilled for a wedge to fail. First, the concept of removability [Goodman and Shi, 1985]. This condition refers to the possibility that the combination of joint planes orientation, relative to slope face, makes it feasible the slope to slide, which is the kinematic analysis [Park and West, 2001]. To avoid ambiguity, from now on, block kinematically unstable will be referred to as removable blocks. In rock mechanics, the removable blocks are also known as key blocks [Goodman and Shi, 1985].

The second condition defines if the block will effectively slide, this is accomplished by checking the balance of resisting and driving forces, under a limit equilibrium framework, to compute the factor of safety (FOS). If the acting forces exceed the resisting forces, the $FOS < 1,0$, which means that a removable block fails. This is a kinetic analysis [Park et al., 2005].

Based on this, for this research project, the probability of failure of a block is computed at two stages, considering a conditional probability, as the probability of having a factor of safety less than 1.0, given that the block is removable. This approach was initially proposed by McMahon [1971] and Glynn [1979]. From that standpoint, the mistake incurred when the analysis focus only on the limit equilibrium analysis is evident since it assumes that the probability of having removable blocks is already 1.0, so it would overestimate the probability of failure. This is discussed by Park and West [2001],

Park et al. [2005].

In regarding the kinematic analysis, two conditions have to be fulfilled by a wedge to be removable [Norrish and Willey, 1996]:

1. The trend of joints intersection and slope must be similar

2. The plunge of intersection line must be lower than slope dip

These conditions are normally checked graphically on stereonet. However, for a large number of computations, the graphical assessment is not feasible. Accordingly, Low and Einstein [1991] proposed that removable wedge are those that meet:

$$\omega < \epsilon < \alpha \qquad (3\text{-}34)$$

where ω is the upper slope, ϵ is the dip angle of the slope intersection and α is the slope dip.

Subsequently, Park and West [2001] found that Eq. 3-34 may interpret as removable, stable blocks for wedges where the angle between the trend of intersection line, β_i and slope dip direction, β_s, is large. Hence, they modified the condition, to compare the apparent slope dip, $\alpha_{apparent}$, in direction of the intersection, rather than the real slope dip, as stated in:

$$\omega < \epsilon < \alpha_{apparent} \qquad (3\text{-}35)$$

in which:

$$\alpha_{apparent} = \arctan(\tan \alpha \cos(\beta_i - \beta_s)) \qquad (3\text{-}36)$$

With this approach, the probability of having unstable blocks would be the probability of having removable blocks times the probability of getting a factor of safety less than 1.0 given that is removable:

$$PF = P(FS < 1,\!0) = P(removable) * P(FS < 1,\!0|removable) \qquad (3\text{-}37)$$

3.4. Probabilistic analysis of rock wedge stability

Up to this point, a full definition of the elements required for the probabilistic analysis proposed in this chapter has been presented, including:

1. Estimation and simulation of spherical data through Fisher and Kent distributions

2. Wedge stability deterministic model

3. Kinematic and kinetic analysis

4. Definition of the conditional probability of failure

This section presents a framework to perform probabilistic analysis of rock wedges considering different scenarios. First, the input properties are divided into four groups, to define the scenario to evaluate:

1. Joint plane orientation

2. Rock slope orientation

3. Rock joints strength parameters

4. Slope geometry

As for rock joint planes, they are first modeled as deterministic, assuming the mean direction is the representative value for each piece of evidence. Then, the stability is computed for all possible combinations of measured planes. This case is referred to as a combination. Finally, joints are modeled as Kent distributed random variables, with the parameters estimated according to the above-described estimation algorithm.

The slope orientation, first, it is assumed as deterministic. The second scenario considers the dip and dip direction as independent variables and models them as a univariate uniform and normal distributions. Subsequently, the

Table 3-1.: Main scenarios considered

Variable	Assumption		
Joint Orientation	Deterministic	Combinations	Fisher/Kent
Slope Orientation	Deterministic	Probabilistic	Fisher/Kent
Strength params	Deterministic	Probabilistic	Probabilistic
Scenario	1	2	3

same slope orientation is modeled as a random variable following a Kent distribution. In this case, several combinations of concentrations and ovalness can be assessed.

Regarding the strength parameters, they are modeled as deterministic and as a random variable, assuming they follow a normal and uniform distribution. Parameters of the assumed distribution were calculated based on the available information.

Given that there is a wide range of scenarios and combinations of those, Table **3-1** summarizes the most relevant scenarios considered. The algorithms can be modified to consider any other combinations of input parameter assumptions.

Below, a description of the most relevant cases is presented, starting with the deterministic approach.

3.4.1. Deterministic modeling

Figure **3-11** includes the algorithm for each realization of the deterministic stability model. This deterministic approach can be applied to information from different sources since the main idea of this project is to deal with uncertain information from different sources.

3.4.2. Joints: combinations, slope: probabilistic univariate, strength: probabilistic

Regarding the joint planes information, this scenario is based entirely on the information gathered during the field mapping work. Here, the stability
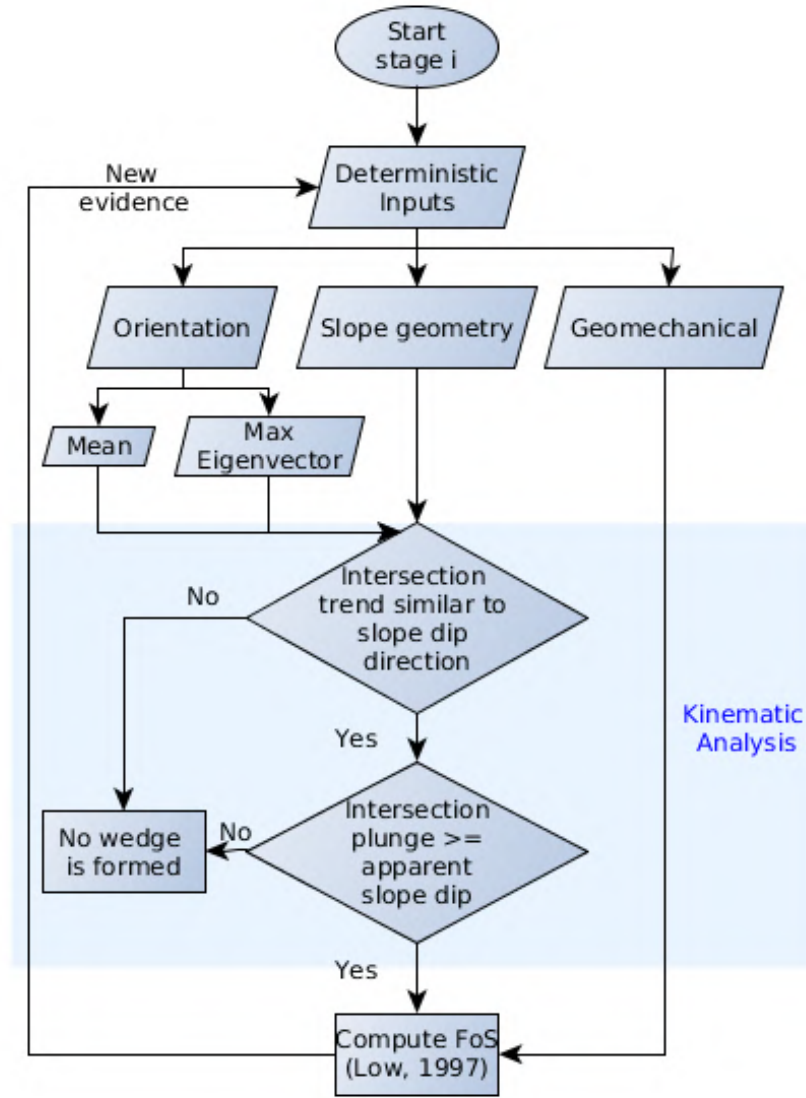
Figure 3-11.: Algorithm to calculate the wedge factor of safety

of wedges formed from each possible combination of planes from each joint set is computed. The number of combinations of planes N controls the total number of realizations, N , as defined by Eq. 3-38. Below a description of the algorithm is given:

$$N = N_{j1} * N_{j2} \tag{3-38}$$

where N_{j1} and N_{j1} are the number of measured planes for joints 1 and 2, respectively.

The algorithm for this simulation is included in Figure **3-12**. The computatio-
nal cost of this approach is low compared to scenario 3, since no simulation of
spherical data is required. However, the most consuming task is the simulation
of univariate parameters. Specially with the truncated normal distribution.
In the following a brief description of the computation algorithm is given.

1. For a piece of information i

2. Define a combination j of joint planes 1 and 2

3. Simulate the slope dip and dip direction, according to the univariate
   probability distribution

4. Simulate the strength parameters, according to the asssumed probabi-
   lity distribution

5. Verify the kinematic conditions for failure. If not removable go back
   to 2. If kinematically unstable update number of removable blocks
   $N_{removable} = N_{removable} + 1$.

6. Compute the factor of safety, FS. If $FS < 1,0$, update the number of
   unstable blocks $N_{unstable} = N_{unstable} + 1$.

7. Go back to step 2, update $j = j + 1$

8. Repeat while $j < N$

9. Repeat for piece of evidence $i + 1$

As a result, the factor of safety for each realization is obtained, along with
the number of removable and unstable blocks. Consequently, the probability
of failure can be computed, as described by Eq. 3-37, leading to:

$$PF = \frac{N_{removable}}{N} * \frac{N_{unstable}}{N_{removable}} = \frac{N_{unstable}}{N} \tag{3-39}$$

From these results, a probability function can be built for the removable
blocks, since the total cumulative probability must be equal to 1. Hence, for

the wedge failure, the value PF computed as in Eq. 3-39 is always lower than the one measured from this probability function. It is important to stress that the meaningful probability distribution is the one defined by Eq.3-39.
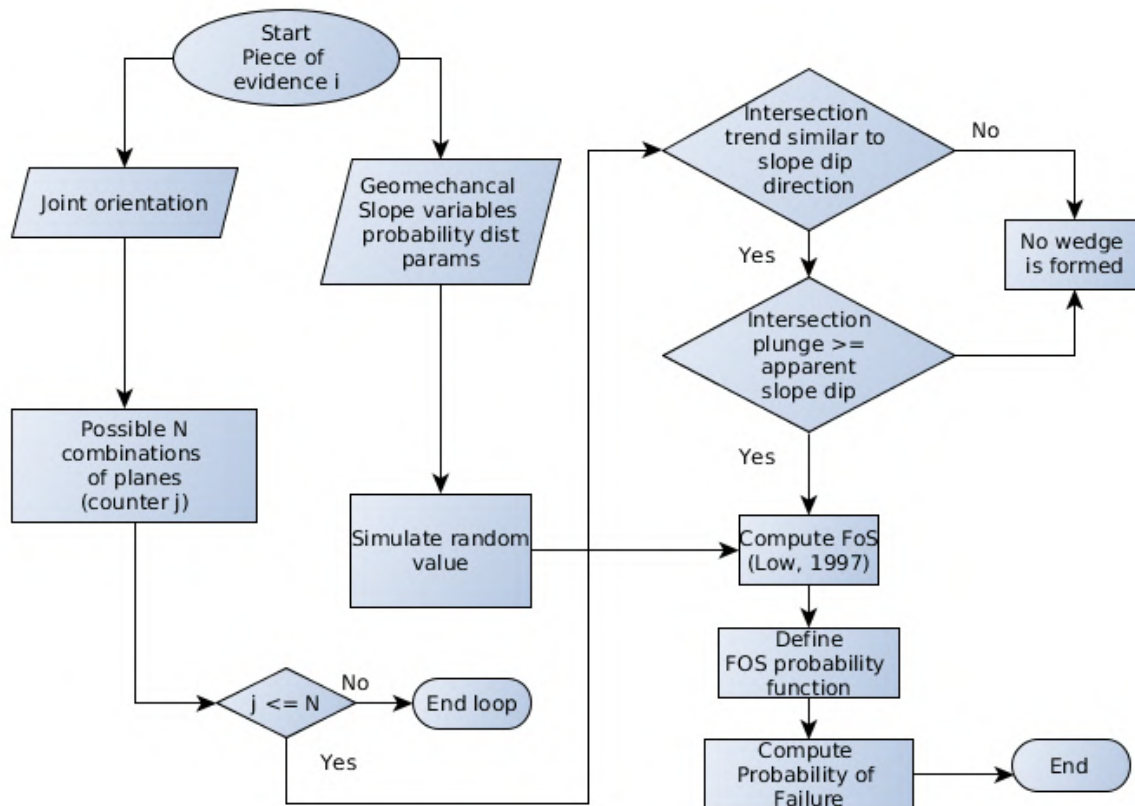
Figure 3-12.: Algorithm to calculate the probability of failure

Joint combination of measure data and other parameters modeled as probabilistic

3.4.3. Joints: Kent, slope: Kent or probabilistic, strength: probabilistic

Like Scenario 2, Scenario 3 is based on the information collected during the field mapping. Nevertheless, these data are utilized to estimate the probability distribution parameters, which will be the input for simulating planes, as the algorithm depicted in Figure **3-13**.

This scenario considers the slope face as an uncertain plane; consequently, it is simulated as a spherical random variable. This assumption differs from

Scenario 2, where slope dip and dip direction were considered independent variables, each with its own univariate probability distribution.

When both joints and slope planes are modeled as Kent distributions, the process for the probability of failure and slope computation is more resource consuming. Firstly, parameters of Kent distribution to model joints have to be estimated for each piece of available evidence. Then, a N number of simulation of joint planes and slope has to be carried out. This means to execute the A/R algorithm N times. Considering that the least number of realization in this chapter is 1000, this is the most time-consuming part of the process. Figure **3-13** shows the algorithm to work under these conditions.
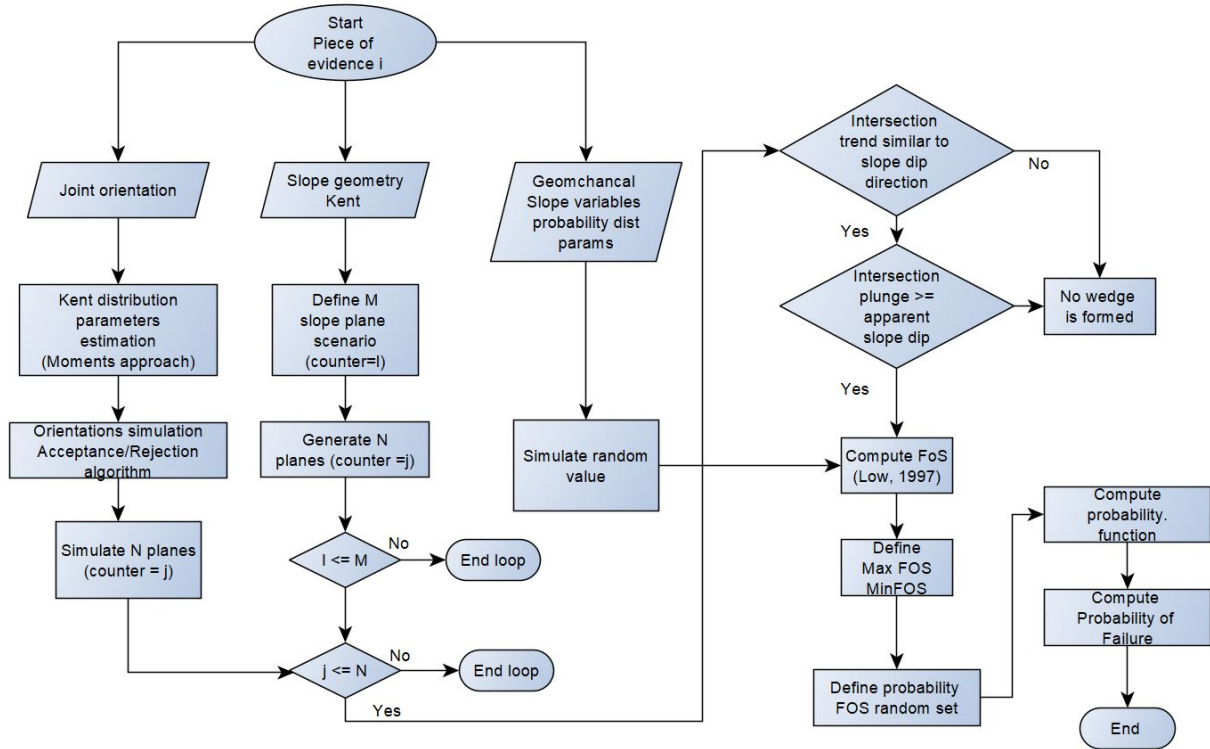
Figure 3-13.: Algorithm to calculate the probability of failure of a wedge

Joints and slopes orientation, as well as, strength parameters are considered as random variables

Again, the algorithm is computed for each available piece of evidence. In this case, several combinations of kappa and beta are evaluated. Hence, the algorithm iterates over each scenario. Next, the algorithm is described:

1. For a piece of information i

2. Estimate Kent distribution parameters for each plane, based on the measured planes

3. For a scenario l, defined by a combination of slope κ and β

4. For each block realization j

5. Simulate a slope direction

6. Simulate the joint planes according to Kent

7. Simulate the strength parameters, according to the assumed probability distribution

8. Verify the kinematic conditions for failure. If not removable go back to 2. If kinematically unstable update number of removable blocks $N_{removable} = N_{removable} + 1$.

9. Compute the factor of safety, FS. If $FS < 1{,}0$, update the number of unstable blocks $N_{unstable} = N_{unstable} + 1$.

10. Go back to step 4, update $j = j + 1$

11. Repeated while $j < N$

12. Repeat for scenario $l + 1$. Next combination of slope κ and β

13. Repeat for piece of evidence $i + 1/$

This algorithm implies the formulation of three nested loops. One for each piece of evidence, one for each slope scenario and one for each block realizations. Therefore, the total number of realization of the block model is:

$$N_{totalRealizations} = N_{sourcesInfo} * N_{slopeScenarios} * N_{realizations} \qquad (3\text{-}40)$$

The two cases just presented correspond to the most simple and most elaborated cases analyzed. In between, there are several possibilities such as combining deterministic slope with Kent distributed joints orientation. Some

of these scenarios are illustrated by examples in the following section and Appendix B.

In Appendix B a comparison between using Fisher and Kent distribution in the probability of wedge failure is assessed. As a result, an important influence of the ovalness factor in the probability of failure was detected, which provides evidence on the suitablity of using Kent distribution instead of Fiher's for modeling the variability of orientation datasets.

3.5.   Probabilistic stability analysis of El Pedregal Mine

Information on rock mass geometry and mechanical properties was collected in El Pedregal Mine as described in Appendix A.

Based on the information collected from different sources along the main slope, during the last 20 years, a probabilistic analysis of the stability of wedges generated for a slope with an average dip direction of 330.

Table **3-2** summarizes the values adopted for the deterministic computations. A truncated normal and uniform distribution were assumed for the data other than joints orientation.

The uniform distribution parameters (bounds) were extracted from the available information at each period, as shown in Table **3-3**. The truncated normal distribution has the same bounds, but with the mean and standard deviation included in Table **3-4**. Concerning the orientation, Table **A-3** summarizes the mean orientation computed for each piece of information for the planes involved in the wedge failure analyzed.

Joints simulation

This section shows results from the estimation of the probability distribution for joint orientation, i.e., spherical data. The Kent distribution parameters were computed by using the algorithm depicted in Figure **3-6**, based on the orientation measured at each period.

In order to illustrate the importance of considering the more general Kent distribution, over the Fisher distribution, Figure **3-14** compares the measured

<div align="center">Table 3-2.: El Pedregal mine. Input deterministic parameters</div>

Year	Material	Slope Dip Direction (°)	Slope Dip (°)	Friction 1 (°)	Cohesion 1 (kPa)
1997	Claystone	330	70	23	22
2011	Sandstone	330	70	32	29
2016	Claystone	330	70	23	51
2017	Claystone	330	70	27	78
	Sandstone	330	70	40	75

Year	Friction 2 (°)	Cohesion 2 (kPa)	Height (m)	Omega (°)	Unit weight (gr/cm^3)
1997	23	22	15	10	2.53
2011	32	29	15	10	2.42
2016	23	51	15	10	2.50
2017	27	78	15	10	2.59
	40	75	15	10	2.55

data with the planes simulated using, firstly a Kent distribution with the ovalness and concentration parameters estimated for the data, and secondly, the simulated orientation assuming a Fisher distribution, i.e. with an ovalness factor $\beta = 0$.

From a visual inspection of these plots can be concluded that the Kent distribution simulates better orientation where there is a higher variation in the azimuth than in the dip. Indeed, the Fisher cannot account for the entire range of measured dip directions, given the constraint imposed by the Gaussian symmetry.

On the other hand, from the comparison between Figure **3-14**a and Figure **3-14**c it can be inferred that there is a reduction in the dip interval, i.e., planes at low densities away from the set mean are not simulated. This is associated to the A/R algorithm for simulating the Kent planes. There, high concentration was assumed, in order to model the FB_5 as a bivariate normal distribution [Kent, 2012], concentrated near $x_1 = 1$, $x_2 = 0$ and $x_3 = 0$. Hence, high concentration of the simulated planes is obtained.

Table 3-3.: El Pedregal mine. Uniform distribution parameters

		Slope Dip Direction (°)		Slope Dip (°)		Friction 1 (°)		Cohesion 1 (kPa)	
Year	Material	Min	Max	Min	Max	Min	Max	Min	Max
1997	Claystone	325	335	67	73	18	26	3	20
2011	Sandstone	325	335	67	73	24	40	0	87
2016	Claystone	325	335	67	73	19	25	23	68
2017	Claystone	325	335	67	73	17	37	41	146
	Sandstone	325	335	67	73	29	49	19	183

	Cohesion 2 (kPa)		Height (m)		Omega (°)		Unit weight (gr/cm^3)	
Year	Min	Max	Min	Max	Min	Max	Min	Max
1997	3	20	14	16	0	20	2.41	2.66
2011	0	87	14	16	0	20	2.41	2.44
2016	23	68	14	16	0	20	2.59	2.6
2017	41	146	14	16	0	20	2.59	2.6
	19	183	14	16	0	20	2.5	2.61

Deterministic analysis

The analysis was performed for each piece of evidence, considering the information collected on geomechanical parameters. Moreover, the factor of safety was computed for the two defined materials, one for the claystone and one for the sandstone, with the same joint orientation parameters at each piece of information.

Deterministic results were selected to validate the results from the script developed in this project to compute the factor of safety of the wedge, based on Low's approach [Low and Einstein, 1991, Low, 1979, 1997]. Accordingly, Table **3-5** shows the factor fo safety obtained by the code compared to those obtained by using the software Swedge v4.0 from Rocscience (available at www.rocscience.com). As can be seen, both approaches yield the same results. This verification is crucial because this deterministic model is the reference case for the probabilistic approach in this chapter; hence, millions of realizations of this model are computed.

Moreover, these results validate the proposal for the computation of the horizontal angles presented in Section 3.3.1, which is also essential because it allows us to systematically assess and program the algorithm for the different modes of failure.

Table 3-4.: El Pedregal mine. Truncated normal distribution parameters

		Slope Dip Direction (°)		Slope Dip (°)		Friction 1 (°)		Cohesion 1 (kPa)	
Year	Material	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1997	Claystone	330	2.5	70	1.5	23	4	11	6
2011	Sandstone	330	2.5	70	1.5	32	6	29	9
2016	Claystone	330	2.5	70	1.5	23	5	51	26
2017	Claystone	330	2.5	70	1.5	27	10	78	30
	Sandstone	330	2.5	70	1.5	40	6	75	30

	Cohesion 2 (kPa)			Height (m)		Omega (°)		Unit weight (gr/cm^3)	
Year	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1997	4	11	8	15	0.5	10	4	2.53	0.06
2011	6	29	9	15	0.5	10	4	242	0.02
2016	5	51	26	15	0.5	10	4	2.5	0.02
2017	10	78	38	15	0.5	10	4	2.59	0.005
	6	75	30	15	0.5	10	4	2.55	0.03

Std: standard deviation

Table 3-5.: El Pedregal mine. Deterministic factor of safety for mean orientation 1997, 2011, and 2016

Year	Source ID	FS – Claystone		FS – Sandstone	
		Script	Swedge	Script	Swedge
1997	1	1.4	1.4	2.5	2.5
	2	1.5	1.5	2.6	2.6
	3	0.8	0.8	1.6	1.6
2011	1	0.9	0.9	1.7	1.7
	2	1.1	1.1	2.1	2.1
	3	1.0	1.0	1.9	1.9
	4	1.1	1.1	2.0	2.0
2016	1	1.1	1.1	2.1	2.1

On the other hand, since joint claystone strength parameters are lower than sandstone's, the wedge factor of safety at each piece of information is lower. Hence, subsequent probabilistic computations will be carried out with claystone strength parameters.

(a) 1997. Piece of information 0

(b) 1997. Piece of information 0 as Fisher

(c) 1997. Piece of information 0 as Kent

(d) 1997. Piece of information 2

(e) 1997. Piece of information 2 as Fisher

(f) 1997. Piece of information 2 as Kent
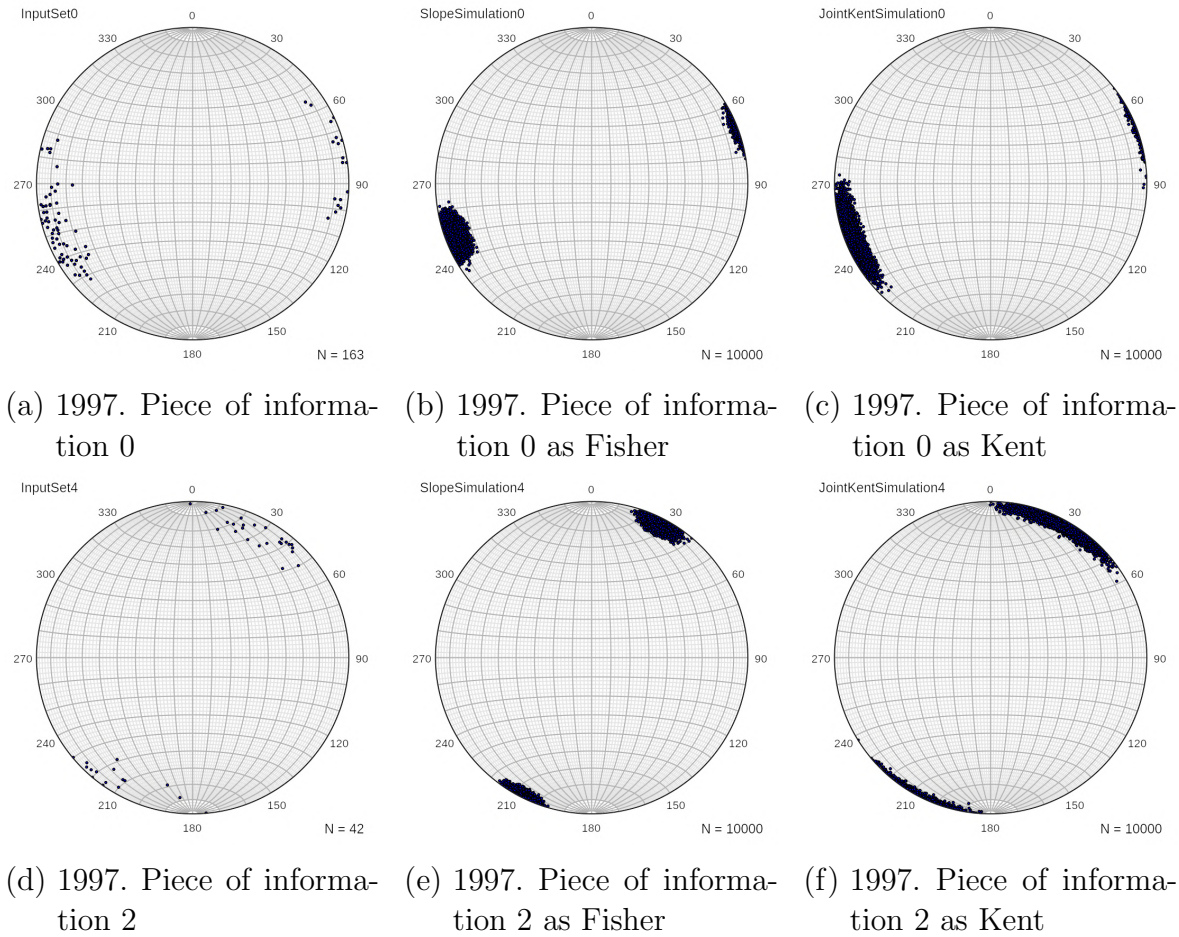
Figure 3-14.: Comparison among measured orientation and 10000 orientation planes simulated as Fisher and Kent distributions for sources of information 0 and 2

Table **3-6** summarizes results for the deterministic evaluation in 2017, for each piece of information (The location of each piece of information is included in Appendix A). A comparison at each station for each lithology of the information collected in 2017, between the mean orientation and the weighted main orientation. The latter corresponds to the orientation weighted by size, which is computed according to the size of the plane linked to each orientation. The weighted orientation was computed using ShapeMetriX3D (see GmbH [2010]).

These computations are the first evidence of the variability of the factor of safety. Even though the information was collected at the same slope and

Table 3-6.: Factor of safety. Analysis: Deterministic. Year: 2017.

Piece of evidence	Station	Mean		Weighted Mean	
		Claystone	Sandstone	Claystone	Sandstone
0	1	4.1	4.9	2.5	2.8
1	2	4.1	4.9	4.1	5.0
2	3	8.5	10.1	8.0	9.4
3	4	4.0	4.8	3.5	4.2
4	5	4.6	5.5	5.3	6.4
5	6	4.6	5.6	4.4	5.3
6	7	5.3	6.4	5.0	6.0
7	8	6.6	8.1	5.2	6.3
8	9	9.7	12.0	8.0	9.9
9	10	4.6	5.5	4.7	5.6
10	12 − Lower	4.6	5.5	4.2	5.0
11	12 − Upper	4.7	5.6	4.6	5.4
12	13	3.7	4.5	4.0	4.8
13	14	3.6	4.3	3.8	4.6
14	15	8.1	5.8	5.1	6.1
15	16	7.1	5.3	4.4	5.2

block, the factor of safety ranges from 2.5 to 8.5 for the claystone and between 2.8 and 12.0 for the sandstone.

Let $p = 1/N_{sourcesInfo}$ be the probability assigned to the factor of safety of each piece of evidence. Then, a probability function for the factor of safety can be computed according to the algorithm plotted in Figure **3-15**. There, the factors of safety are organized in ascending order, and the probability is assigned as $p_i = p_{i-1} + p$.

As a result, two probability functions are obtained, one for the mean orientation and one for the weighted mean orientation, as shown in Figure **3-16**a for the claystone and Figure **3-16**b for the sandstone. These probability functions are Dempster-Shafer structures.

This is a first approach to express the variability of the results, accounting for the averaged information at each station. A first approach to a measurement of the uncertainty is provided by the area between bounds. In this case, the areas are 0.58 and 0.72, for claystone and sandstone, respectively.

At this point, it is important to note that the Dempster-Shafer structures are a natural way to describe the variability of the information. Particularly,
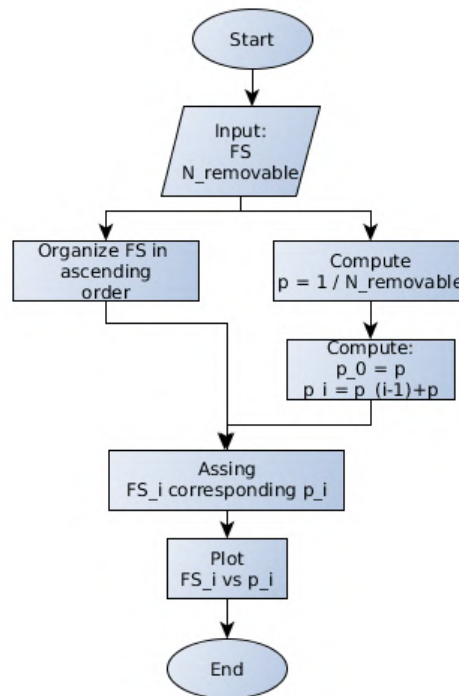
Figure 3-15.: Algorithm to calculate a probability function from the deterministic factor of safety of each piece of evidence

these structures represent the natural variability of the orientation data and the epistemic uncertainty associated with the computation of the factor of safety using the mean and the weighted mean orientations.

Joints: combination of planes; slope and strength parameters: probabilistic

In this section, the factor of safety of the wedge is computed considering the combinations of orientation planes measured using, either, the compass or ShapeMetrix3D, and the other inputs as probabilistic with the parameters included in tables **3-3** and **3-4**, for uniform and truncated normal distributions, respectively.

The assessment is carried out for every available source of information. As a result, for each piece of evidence, a probability function, along with the probability of failure in terms of the factor of safety, is obtained. At this point, it is essential to remind that the probability of failure accounts for the
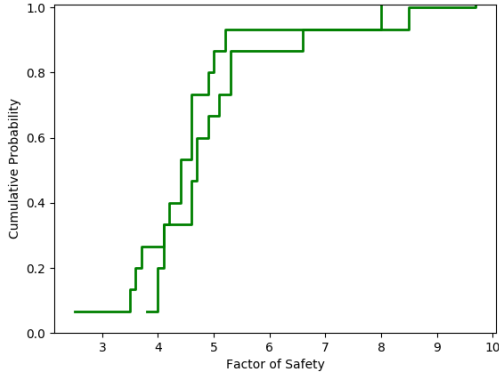
(a) Material: Claystone. Year: 2017   (b) Material: Sandstone. Year: 2017
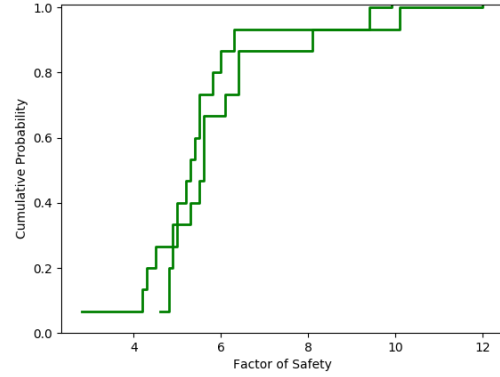
Figure 3-16.: Deterministic factor of safety computed expressed as a Dempster-Shafer structure

number of failed wedges over the total number of realizations. Conversely, the probability functions considering just the removable blocks.

Figures **3-17** and **3-18** illustrate the probability function resulting from the removable blocks, with orientation and geomechanical properties modeled as random variables with uniform and normal distribution. The first noticeable difference is the shape. Some of them are nicely smooth functions (e.g., piece of information 2 from 2017), while others are uneven (e.g., piece of information 2 from 2011). This is controlled by the available amount of measure planes and the total proportion of removable blocks. For example, in 2011 for the piece of information 2, just 56 and 17 planes were measured, leading to 952 combinations of poles. Whereas, for the piece of information 2 from 2017, 120 and 86 joint were measured for planes 1 and 2, respectively, turning into 10320 realizations of the model. This difference clearly explains the uneven and smooth behavior.
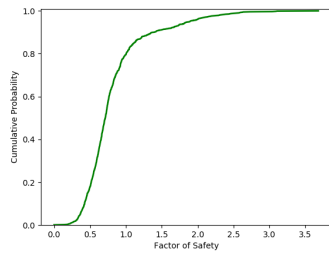
(a) 1997. Piece of information 2
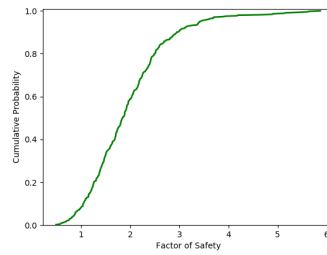
(b) 2011. Piece of information 2
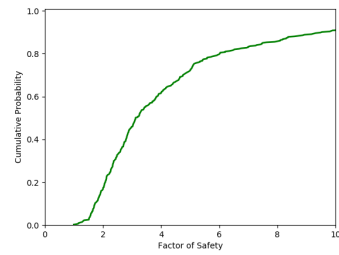
(c) 2016. Piece of information 1
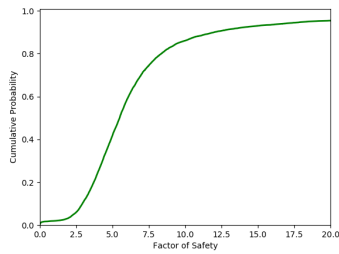
(d) 2017. Piece of information 2. (Station 2)
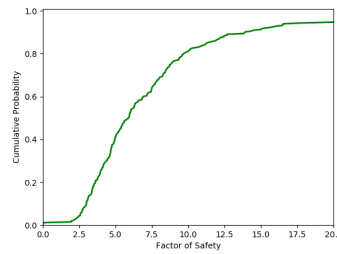
(e) 2017. Piece of information 3. (Station 3)
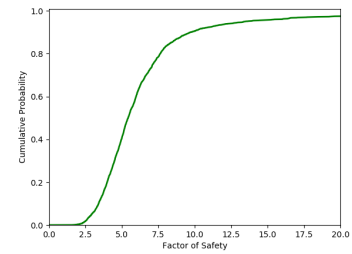
(f) 2017. Piece of information 10. (Station 10)
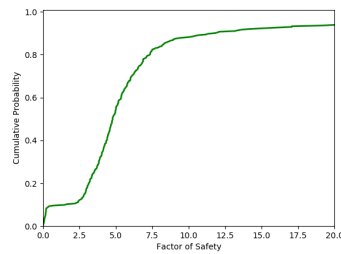
(g) 2017. Piece of information . (Station 12-upper)
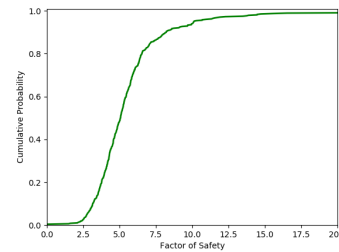
(h) 2017. Piece of information 13. (Station 13)

Figure 3-17.: Selected probability functions for model: Joints: Combination. Slope: Uniform distributed. Strength: Uniform distributed
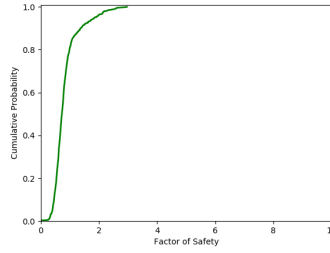
(a) 1997. Piece of information 2
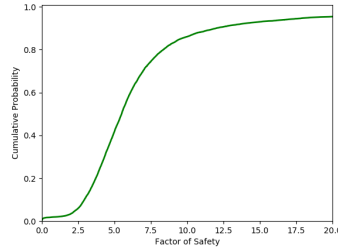
(b) 2017. Piece of informa-
tion 2. (Station 2)
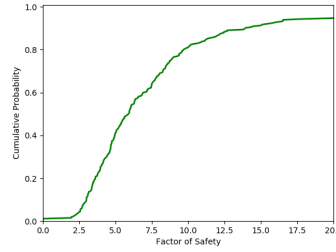
(c) 2017. Piece of informa-
tion 3. (Station 3)
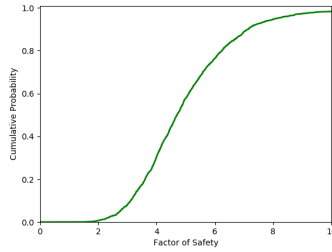
(d) 2017. Piece of informa-
tion 10. (Station 10)
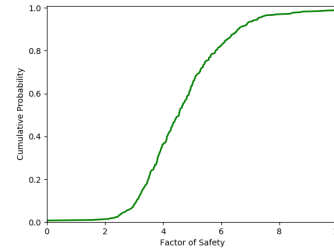
(e) 2017. Piece of informa-
tion 13. (Station 13)

Figure 3-18.: Selected probability functions for model: Joints: Combination.
Slope: Normal distributed. Strength: Normal distributed

Regarding the variability of the probability function, Figure **3-19** shows the
probability function for the factor of safety computed for the information
collected in 2017 utilizing the measurement system ShapeMetrix3D and the
other parameters as uniform distributed. These functions were calculated
using the same uniform distribution parameters, which means that the pro-
bability function's variation reflects just the change in the measured input
planes. This variation may be dramatic, only for illustration, according to
the information collected in 2017, the probability of having a factor of safety

lower than 5.0, ranges between 0.18 and 0.75. Likewise, the probability of getting a factor of safety lower than 10.0 is between 0.50 and 0.92.
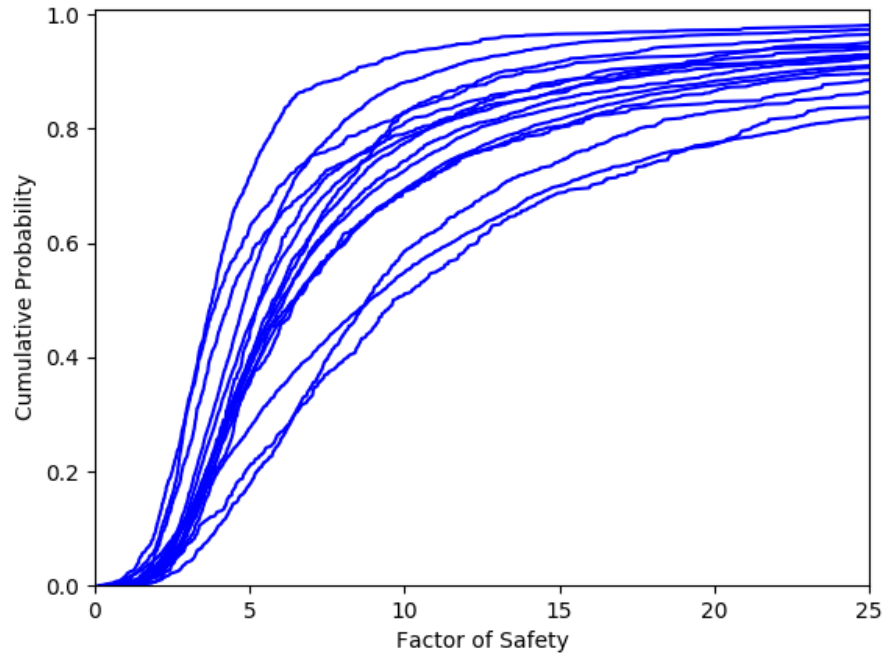
Figure 3-19.: Probability function for the factor of safety considering the information collected in 2017

Joints: Kent/Fisher distributed; slope and strength parameters: probabilistic

Results from Section 3.5 showed the impact of modeling the joint orientation using the Kent distribution over the Fisher distribution. In this sense, here, the probabilistic factor of safety is computed, considering the joint orientations as a spherical distribution. In this case, the joint 1 is preferably Kent-distributed, while the bedding is Fisher-distributed.

For each source of information, a total of 10000 planes orientation were simulated according to the acceptance-rejection algorithm A/R. Hence, the factor of safety was computed 10000 times for each source. The slope orientation was considered as deterministic, while the other inputs were assumed to have

a uniform distribution, with the bounds included in Table **3-3**. With this assumptions, Table **3-7** shows the probability of failure computed for the pieces of information collected in 2017.

Moreover, Table **3-7** shows the probability of failure ($P(FoS < 1,0)$)calculated for the combinations of measured joint planes, instead of Kent simulated data. Figure **3-20** depicts the results included in Table **3-7**, while Figure **3-21** shows the probability of having wedges with a factor of safety lower than 3.0. As can be seen, there is a noticeable difference between the two approaches, especially when the factor of safety is lower than 1.0.

Figure **3-22** shows the probability function constructed for the removable blocks for the two scenarios described above. First, **3-22**a presents the probability function for the combinations of planes and **3-22**b shows the same function, but with the planes simulated from the Kent distribution. Again, there is no only one probability function, but many, even for the same slope. It is important to highlight that the other input parameters are modeled as uniformly distributed in both cases.

Table 3-7.: Probability of Failure according to information collected in 2017

Source	Combinations	Kent
1	0.00E+00	0.00E+00
2	5.08E-03	1.60E-03
3	0.00E+00	0.00E+00
4	1.28E-03	0.00E+00
5	3.98E-04	2.00E-04
6	4.25E-03	1.20E-03
7	1.70E-03	1.00E-03
8	4.89E-04	0.00E+00
9	2.35E-03	4.20E-03
10	5.16E-03	2.00E-04
11	6.30E-03	1.04E-02
12	0.00E+00	0.00E+00
13	5.54E-03	2.97E-02
14	1.07E-03	0.00E+00
15	7.94E-04	2.00E-04
16	1.45E-03	0.00E+00

Analysis: Joints modeled as combinations of planes and Ken distributed and other parameters as uniform distributed. Slope modeled as deterministic
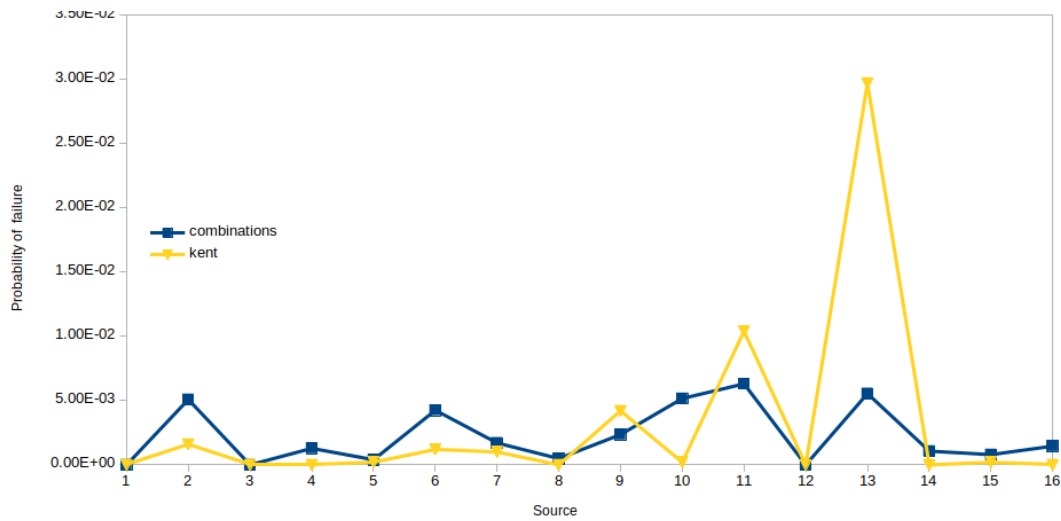
Figure 3-20.: Probability of failure for information collected in 2017

Stability computed with combinations of measured orientations and modeled as Kent distributed
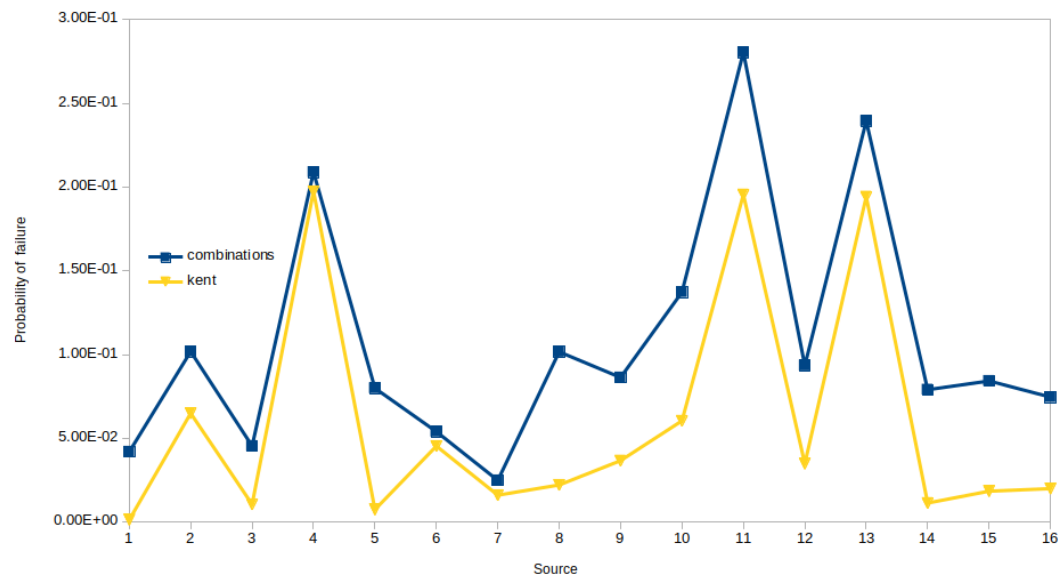
Figure 3-21.: Probability of factor of safety lower than 3.0 for information collected in 2017.

Stability computed with combinations of measured orientations and modeled as Kent distributed

(a) Combinations of measured planes

(b) Simulated orientations
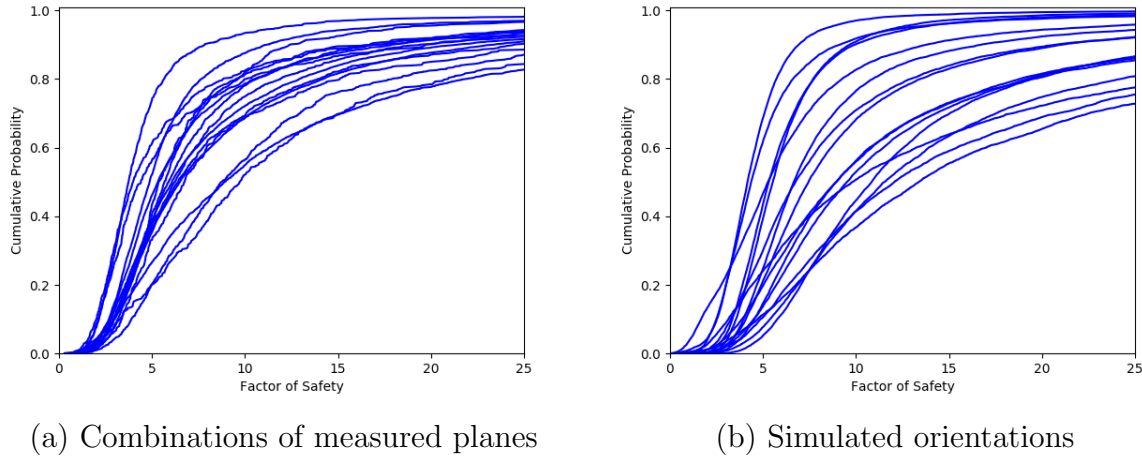
Figure 3-22.: Probability functions for information collected in 2017. Slope orientation modeled as deterministic

3.6. Discussion

Results presented in Appendix B showed an important difference between modeling the variability of orientations with the Kent distribution instead of the Fisher. Figure **3-23** presents an explanation for this; it illustrates the contours generated by Fisher and Kent distributions schematically.

The central assumption of these distributions is that orientations are distributed around a central value (the mean orientation). The probability density reduces as the orientation is farther from the mean.

Firstly, Kent distribution fits the measure data better, compared to the Fisher's, which is the case of many field measurements, as shown in the figure. From the graph, two regions can be distinguished. In region I, the Fisher distribution would lead to simulating some planes within this area, which are not likely to happen, according to the measured data. This will generate unrealistic combinations of planes and wedges that can trigger a failure.

On the other hand, Fisher distribution cannot simulate planes within region II, because they fall out of the symmetry assumption behind this distribution. In this case, several likely wedges are disregarded.

A first thought might suggest that these two regions compensate each other. However, this is not true, since the probability function with the simulation
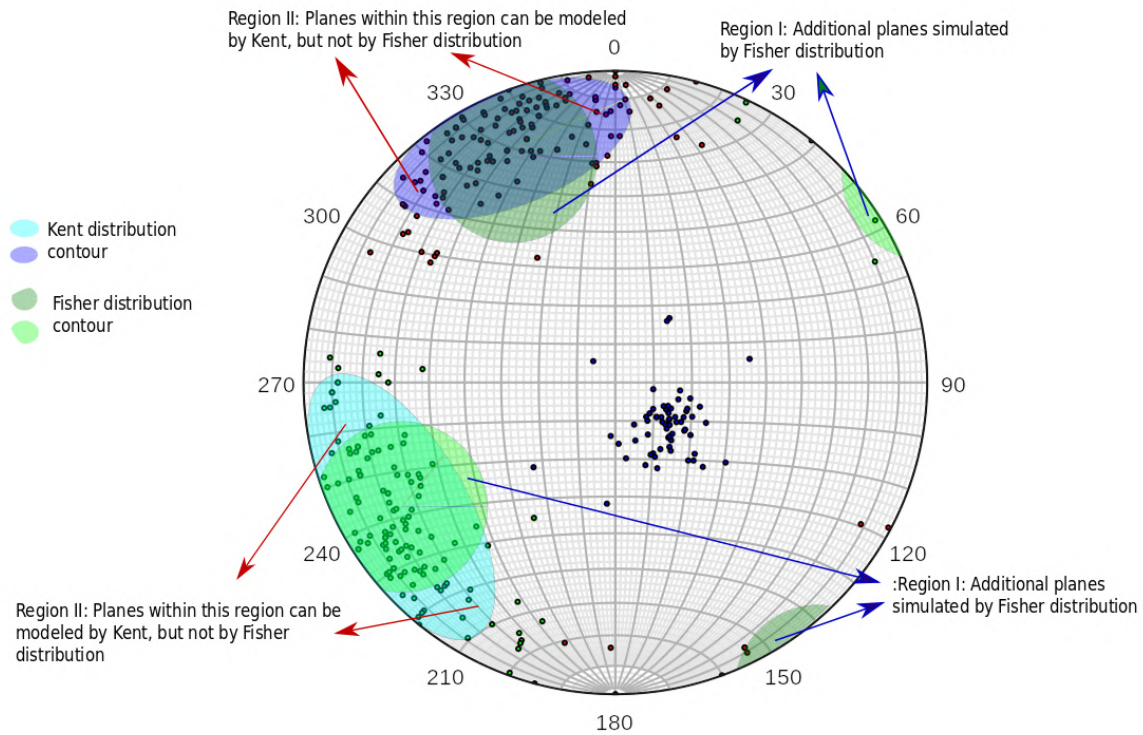
Figure 3-23.: Effect of simulating orientation by Fisher rather than Kent distribution

under Fisher distribution would include unlikely and exclude likely blocks. Besides, the deterministic problem is nonlinear and involves several modes of failure. Numeric results showed that the variation in the probability of failure and the probability functions are as high as 700%, for high ovalness parameter.

Several FOS probability functions were computed under different assumptions for different pieces of information collected in El Pedregal mine. As a result, different probability functions were obtained, which is also reflected in different probabilities of failure. This variability results from the aleatory and epistemic uncertainty of the inputs

It is important to highlight that when the target factor of safety is low, the probability of failure is low, which means that many realizations of the model are required to establish the probability of failure. Whereas, for a high target probability of failure, the probability of failure is higher, so a lower number of

realizations is required to get close results. This is because, when comparing the combinations and the simulations results, there is a higher difference in the probability of failure when the target factor of safety os 1.0 than when is 3.0.

The problem is highly nonlinear, in which several scenarios are involved. The change of geometry at each realization can lead to different mechanisms of failure, which makes it hard to predict the effect of κ and β on the final probability of failure. Sometimes it reduces, and sometimes it increases. However, the effect has to be considered.

3.7. Conclusions

This chapter presented a framework to perform a reliability assessment of the stability of rock wedges, considering the variability of joint planes orientation with the Kent distribution (5-parameters Fisher-Bingham). This is an alternative to the Fisher distribution, which is conventionally used to model de variability of joint planes.

The Kent distribution is expressed in terms of 5 parameters, which have to be calculated from the sampled planes. These parameters were computed with the moments' method, without resorting to an iterative process, which reduces the complexity of the algorithm. Once the 5 parameter are estimated, the simulation of planes is not straightforward since the distribution cannot be directly inverted. Hence, the iterative acceptance-rejection algorithm using a truncated bivariate normal envelope was implemented.

With the algorithms implemented in this chapter, in Appendix B it was demonstrated that there is a significant effect of the probability distribution selected to model the variability of joint planes in the probability distribution function of the factor of safety.

In summary, there is an impact on the selection of the input probability distribution on the factor of safety probability function, when joint sets fit better the Kent distribution. Hence, the approach proposed in this chapter will lead to a more suitable representation of the probability of failure, compared to the conventional approach adopted in rock mechanics, in which the variability of joints orientation is modeled by using the Fisher distribution.

Several assessments were carried out for different pieces of evidence; here the information is available and is used to estimate the input parameters for the subsequent Monte Carlo simulation. Each piece of evidence led to a different probability function, which means that there is no only one function, even within a small area. Instead, there is a family of probability curves that describe the variability of properties.

Therefore, a unique probability function based on a probabilistic description of the variability of the input parameters cannot fully represent by the uncertainty linked to the geomechanical and orientation parameters. This is because probability functions represent the aleatory uncertainty of joints orientations, but not the epistemic uncertainty related to rock structure geometry.

Based on the just mentioned, it makes sense to express the variability as intervals rather than a unique probability function. In the coming chapters, an alternative to evaluating the uncertainty with intervals of probability is presented.

4. Rock Wedge Stability Analysis by Dempster-Shafer theory of evidence

4.1. Introduction

In the last chapter, it was demonstrated that there is no unique probability function for the factor of safety. Instead, there is a family of functions that reflect the quality of the information available. In that chapter, the input information had these features:

- There were robust data sets on joints orientation. Hence, estimation of parameters and subsequent simulation were feasible, according to the moment estimates and acceptance-rejection algorithm, respectively.

- There were several sources of information available, defined by the stations where information was gathered.

- There was scarce information on joints mechanical properties. Hence, their probability distributions were assumed. Besides, the same values were utilized for several pieces of information on joint orientation.

These conditions are not ideal. However, this is the case in most rock engineering project, where technical and financial conditions constraint the amount of information on joint strength parameters.

Under this context, this chapter focuses on modeling the stability of structurally controlled failures in rock slopes. Both epistemic an aleatory uncertainty are accounted for by using a Dempster-Shafer Theory of Evidence. The analysis considers the limited information on geomechanical parameters, e.g., joints strength parameters. Besides, this chapter addresses the issue of combining

limited data on geomechanical parameters, with the more complete sets of information on geometrical properties of joints, i.e., joints orientation. This problem is tackled by resorting to the spherical statistics, modeling orientation with a 5-parameters Fisher Bingham (Kent) distribution. The latter problem has not been addressed for geotechnical application as described below.

In regarding the structure of this chapter, first, an approach for performing reliability assessment combining Dempster Shafer structures with Kent distributed orientations is presented. Then, the process is illustrated by an application example utilizing information collected in El Pedregal mine and reported in Appendix A.

This chapter aims at defining an alternative to perform reliability assessment of rock slope stability taking in to account the uncertainty of input parameters. Firstly, the nature of the input information should be considered. Hence, a wedge stability model as shown in Section 3.2 is selected. For that model, the input information can be divided into joints orientation and geomechanical parameters. This classification is based on the amount of information collected during current engineering projects. Usually, it is possible to gather a robust data set on joint orientation, primarily when techniques like short-range photogrammetry are utilized. On the other hand, the information of joints strength parameters is limited to a few results taken either from laboratory tests or back analysis.

The lack of knowledge induced by the limited amount of information brings an epistemic uncertainty to the problem, that already has an aleatory uncertainty linked to the variability of the joint strength given by its origin. Hence, the geomechanical parameters are suitable to be represented as a Dempster-Shafer structure.

In Chapter 3, an alternative to represent the rock mass joint orientation using Fisher and Kent probability distribution was presented. Based on that approach and, the concept of Dempster-Shafer structure (Chapter 2), a framework for performing rock wedges stability analysis is proposed, taking into account the following considerations:

- Only the epistemic uncertainty induced by the limited information on joint strength parameters is considered.

- Joint strength parameters are represented by DSS, which represents both the epistemic and the aleatory uncertainty. The earlier is expressed by the probability assignment, and the latter by the interval defined at each focal element.

- Joint orientations have an aleatory uncertainty.

- Wedge failure is considered as deterministic model.

The orientation of planes are accounted in two different ways. Firstly, the stability of the wedge is computed considering the combinations of joint planes from each joint set. Secondly, each joint set orientation is simulated as a Kent-distributed random variable.

4.1.1. Joint orientation as combinations

This project has included the use of the close-range photogrammetry to measure the joint planes on the slope face. The technique allows gathering a robust and reliable database on joints orientation, as shown in Appendix A. Based on this, the procedure described in Figure **4-1** is proposed as follows:

1. For a piece of evidence i.

2. For a simulated slope orientation l.

3. For each combination u of focal elements.
   a) Define the combination j of planes.
   b) Verify the kinematic conditions for failure. If not removable go back to a. If kinematically unstable update number of removable blocks $N_{removable} = N_{removable} + 1$.
   c) Compute the factor of safety, FS. If $FS < 1,0$, update the number of unstable blocks $N_{unstable} = N_{unstable} + 1$.
   d) Go back to step a, update $j = j + 1$.
   e) Repeat while $j < N$.
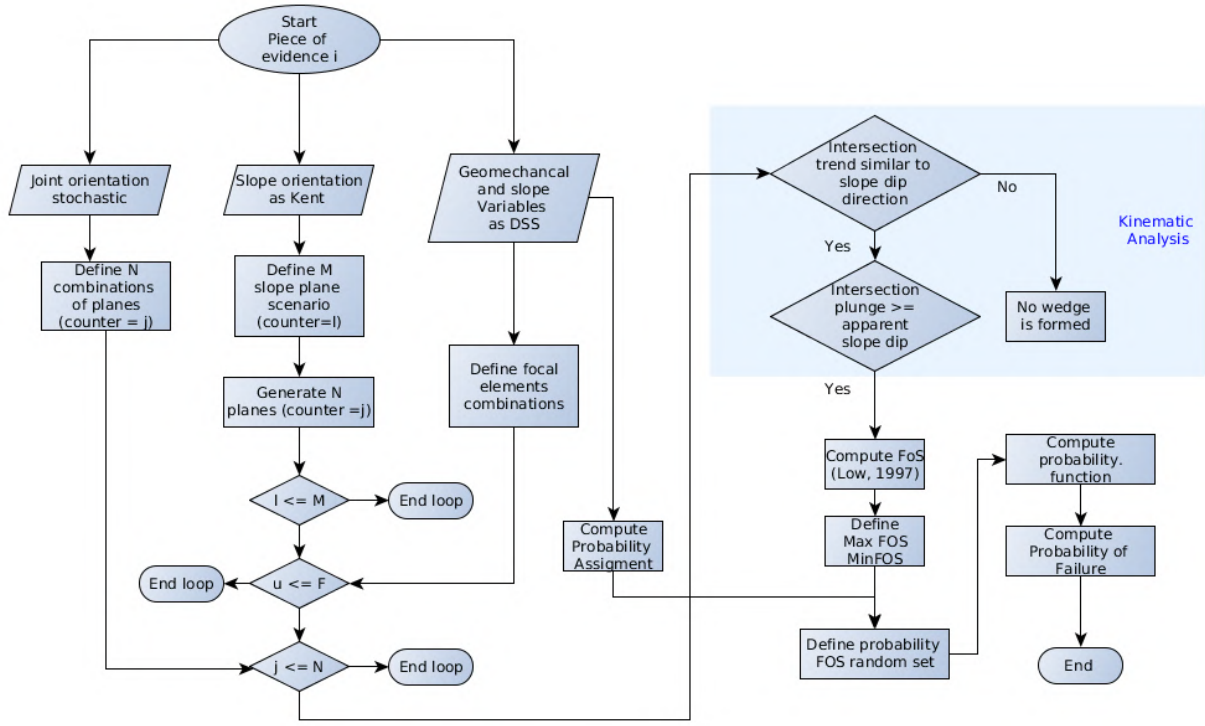
Figure 4-1.: Algorithm to perform reliability assessment of rock wedges considering the combination of joint planes and strength parameters as DSS

4. Go back to step 3, update $u = u + 1$.

5. Repeat while $u < F$.

6. Go back to step 2, update $l = l + 1$.

7. Repeat while $u < M$.

8. Go back to step 1, update $i = i + 1$.

9. Repeat for every available piece of evidence.

In this case the total number of realizations of the model is defined as follows:

$$n_{realizations} = N * n_c * M \tag{4-1}$$

where, N is the number of combinations and is computed as $N = \prod_{i=1}^{2} n_i$, n_c corresponds to the total number of combinations of DSS and, M is the number of simulated planes.

In the procedure presented here, the orientation of the slope face can be modeled either, as a deterministic variable or probabilistic Kent distributed variable.

4.1.2. Joint orientation simulated as Kent

Alternatively, the orientations of the planes are simulated as Kent distributed variables. The general procedure is depicted in Figure **4-2**. This approach is suitable for performing Monte Carlo simulations to be combined with the DSS of strength parameters as follows:
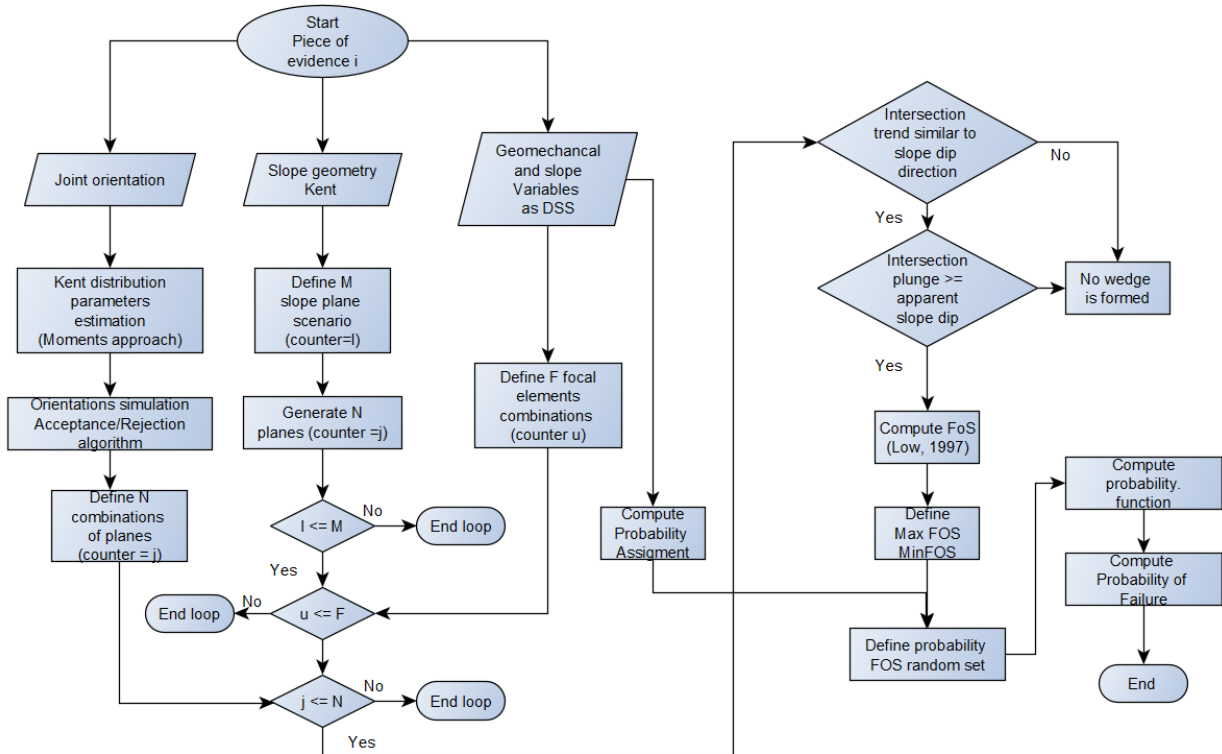
Figure 4-2.: Algorithm to perform reliability assessment of rock wedges.

The orientations area assumed as a Kent distributed random variable and strength parameters as DSS

1. For a piece of evidence i

2. For a simulated slope orientation l

3. For each combination u of focal elements

    a) Simulate plane orientations according to Kent distribution

    b) Verify the kinematic conditions for failure. If not removable go back to a. If kinematically unstable update number of removable blocks $N_{removable} = N_{removable} + 1$.

    c) Compute the factor of safety, FS. If FS¡1.0, update the number of unstable blocks $N_{unstable} = N_{unstable} + 1$.

    d) Go back to step a, update $j = j + 1$

    e) Repeat while $j < N$

4. Go back to step 3, update $u = u + 1$

5. Repeat while $u < F$

6. Go back to step 2, update $l = l + 1$

7. Repeat while $u < M$

8. Go back to step 1, update $i = i + 1$

9. Repeat for every available piece of evidence

Now the total number of realizations of the model is defined as follows:

$$n_{realizations} = H * n_c * M \tag{4-2}$$

where, H is the number of simulations of planes

In this case, the number of realization is n_c higher than a conventional Monte Carlo (MC) simulation, which means that a MC simulation is carried out for each combination of focal elements of geomechanical parameters.

4.2. Example

This section presents an application of the procedure just described. It involves the information collected in El Pedregal Mine and presented in Chapter 2. For illustration purposes, orientation parameters collected in 2017 by using ShapeMetrix3D were utilized, along with the joint geomechanical parameters for the claystone layers.

The amount of information on geomechanical parameters is limited to a few results. Hence, this information was expressed as DSS, as shown in Table **4-1**. Here, it was assumed that the fiction angle of the rock joints is between 20 and 35 degrees, hence the same probability was assigned to the three focal elements. It considers that the expected interval intersects the three focal elements. As for the the cohesion, the same assumption was considered with an expected interval between 50kPa and 60kPa.

Table 4-1.: Joint strength parameters expressed as DSS

Year	Friction			Cohesion (kPa)			γ (ton/m^3)		
	Lower	Upper	P.A.	Lower	Upper	P.A.	Lower	Upper	P.A.
2017	17	20	0.33	44	89	0.5	2.59	2.60	1
	17	37	0.33	41	146	0.5			
	32	37	0.34						

P.A. stands for probability assignment

In regarding the wedge geometry, two possibilities were taken into account. First, the stability for every possible combination of measured planes. Then, planes are simulated assuming that they follow a Kent distribution (See Chapter 3). In both cases, two wedge geometries with two different slope orientations were analyzed. The slope orientation was assumed as deterministic. This assumptions lead to the scenarios listed in Table **4-2**.

Figure **4-3** shows the major circles for the mean plane orientation for the mentioned scenarios. As can be see, the wedge formed by joint set 2 and bedding (scenarios 1 and 2) will have a failure mainly along bedding (Figure **4-3**a), while the wedge formed by joint sets 1 and 2 (Scenarios 3 y 4) will have a failure along the intersection line (Figure **4-3**b). These are the cases

Table 4-2.: Rock wedges slope orientation and involved joint sets

Scenario	Slope Orientation	Joint sets assumption	Joint sets involved	
1	330/70	Combinations of planes	Joint set 2	Bedding
2		Kent distributed	Joint set 2	Bedding
3	098/70	Combinations of planes	Joint set 1	Joint set 2
4		Kent distributed	Joint set 1	Joint set 2

for the mean direction, when variability is considered other modes of failure become relevant.

(a) Scenarios 1 and 2. Wedge formed by slope (red), joint set 2 (green) and bedding (green)

(b) Scenarios 3 and 4. Wedge formed by plane of slope (red), and joint sets 1 and 2 (green)
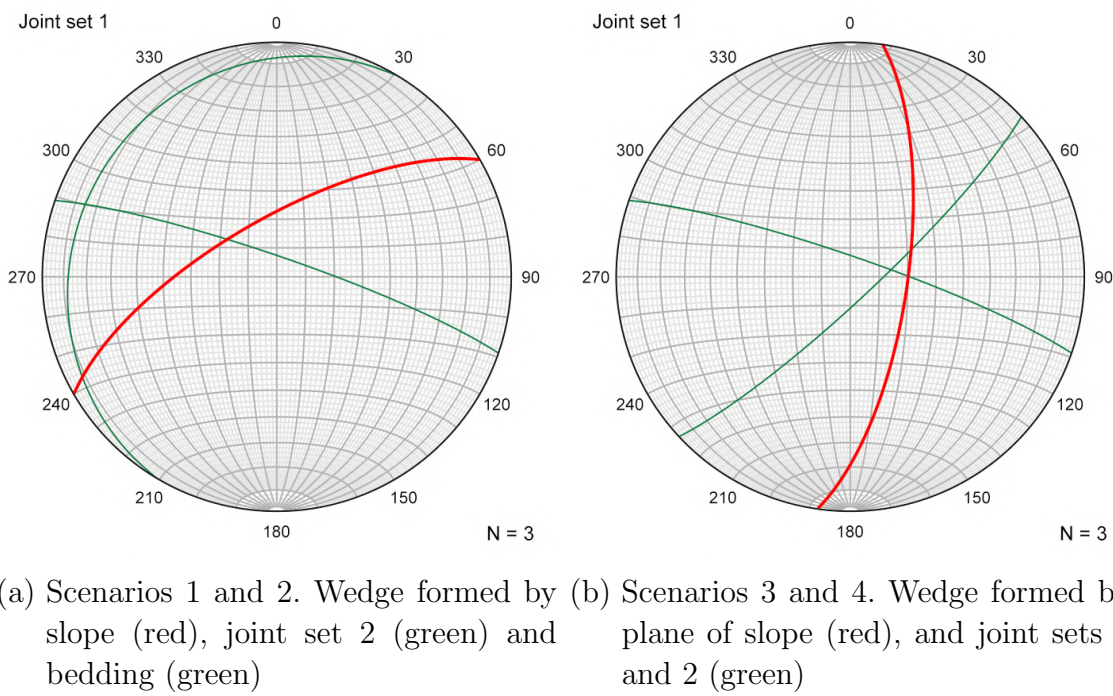
Figure 4-3.: Assessed scenarios. Stereographic projection of major planes for mean orientations

Sixteen pieces of evidence (numbered 0-15) on orientation parameters were collected in 2017 (i.e.regions were photographed and processed by using ShapeMetrix3D. See Figure **A-4** that presents the mean orientation at each piece). A DSS for the factor of safety of the wedge, for each scenario and for each piece of evidence was computed, which led to 64 factor of safety DSSs.

4.3. Results

Based on the algorithm just described, the reliability assessment for the factor of safety of rock wedges was computed. For illustration and comparison purposes, Figure **4-4** shows the DSS for the factor of safety of the rock wedge formed by bedding and joint set 2 planes. Figure **4-4**a depicts the DSS when the combination of planes is assessed, while Figure **4-4**b presents the DSS when orientations are simulated as Kent distributed random variable.

Each DSS consist of an upper left (upper) bound in green and a right (lower) bound in red. The area in between bounds represents the uncertainty linked to the input data.

In Appendix C the DSSs for the sixteen pieces of evidence is presented in figures **C-1** and **C-2**. For the wedge defined by slope, joint set 1 and 2 (Scenarios 3 and 4), results are plotted in figures **C-3** and **C-4** for the combinations and simulated cases, respectively.

(a) Combination of planes   (b) Simulated as Kent
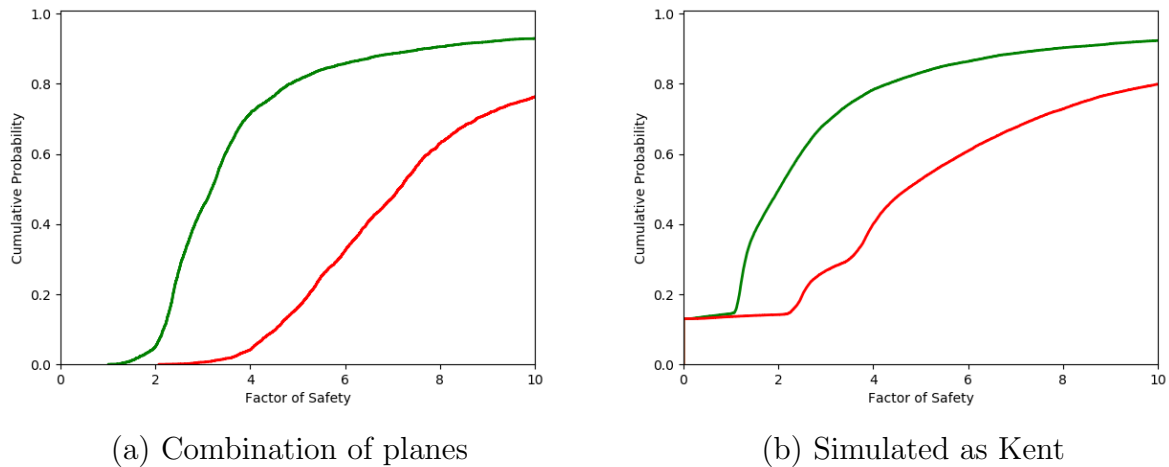
Figure 4-4.: DSS for wedge factor of safety considering the combination of collected planes

Piece of evidence 0. Wedge formed by bedding and joint set 2 planes (Scenarios 1 and 2)

4.4. Analysis of results

The first noticeable result is the variability of the calculated DSS of the factor of safety (FoS DSS). It reflects two facts. First, the variability of rock mass structure geometry, since in all cases, the focal elements were the same. Second, different mechanisms of failure are analyzed, e.g., Figure **C-1**n, for the piece of evidence 13, includes several wedges with a probability equals zero. Such a factor of safety is obtained because contact is lost on both planes, (see Chapter 3), while Figure **C-1**i, for the piece of evidence 8, does not report any factor of safety of zero. This means that throughout the computation none wedge lost contact on both planes. Indeed, the lowest factor of safety is 2.08.

On the other hand, in most cases, the probability function obtained is smooth, which is the result of the large number of computations carried out for each scenario. For the combination analysis, the factor of safety is computed for every possible combination of planes, for a given combination of cohesion and friction angle, i.e., focal elements. Subsequently, repeated for the next combination of focal elements. For the Kent distributed assumption, 10.000 random orientations simulated plane orientations are computed, for every focal element combination. If there are 36 focal element combinations, 360.000 computations are required for each piece of evidence, which turns 5.760.000 wedges assessed for scenarios 2 and 4.

From that analysis, it can be concluded that the amount of computations required to propagate the uncertainty under this framework is much higher than a Monte Carlo simulation. For instance, the number of computations for a conventional Monte Carlo simulation would be 10.000 while for computing the Dempster-Shafer structure is $10,000 * n_c$, i.e. In other words, n_c Monte Carlo simulations are required to compute a Dempster-Shafer structure for the factor of safety. In this case, 36 Monte Carlo simulations are performed for a single FoS DSS.

With a 32 GB RAM and an Intel Core i7 standard laptop, it takes 15 minutes to compute the 16 FoS DSS for the analyzed scenarios when combining the measured planes (the stability of 1.188.864 wedges was carried out). While, when planes are simulated as Kent, it took 64 minutes to compute 16 FoS DSS

for 5760000 wedges, which demonstrate the suitability of the methodology under the applied context.

In fact, this is one of the main limitations of Evidence Theory for propagating uncertainty through computation models. This makes many engineering problems impractical from the computational cost standpoint [Ferson et al., 2007]. However, for a wedge limit equilibrium assessment, the proposed methodology is practical.

When comparing the Fos DSS from Kent simulated planes against those from combinations of planes, there is a higher probability of failure when planes are simulated. For instance, for the piece of information 8, the lowest factor of safety reported, when only combinations of planes is considered, is 2.08, while for the probabilistic case is 0, see figures **C-1**i and **C-2**i. Similar results are calculated for pieces of information 0, 1, 6, 7, 9, 11, and 15. This difference reflects the fact that the combination of the collected data does not account for several planes that are still likely, but they do not daylight on the slope face. From this standpoint, the analysis of just the combination of planes underestimates the probability of failure, which evidently goes against the safety of an engineering slope design.

This issue on the combination assessment is reflected in one of the most relevant results of the evidence theory. The DSS does not provide a unique probability of failure, but a range within the actual probability lies based on the input DSS. Since for the scenario 1 (combinations of planes) the probability of failure is close to zero an underestimated, the size of the probability of failure interval tends to zero, while for the scenario 2 (probabilistic simulation of planes) it is higher than zero. This result is plotted in Figure **4-5**, on the left, the probability of failure for the probabilistic DSS is presented, on the right, the comparison with combination analysis is depicted. It is evident that the probability of failure interval is larger when orientations are simulated.
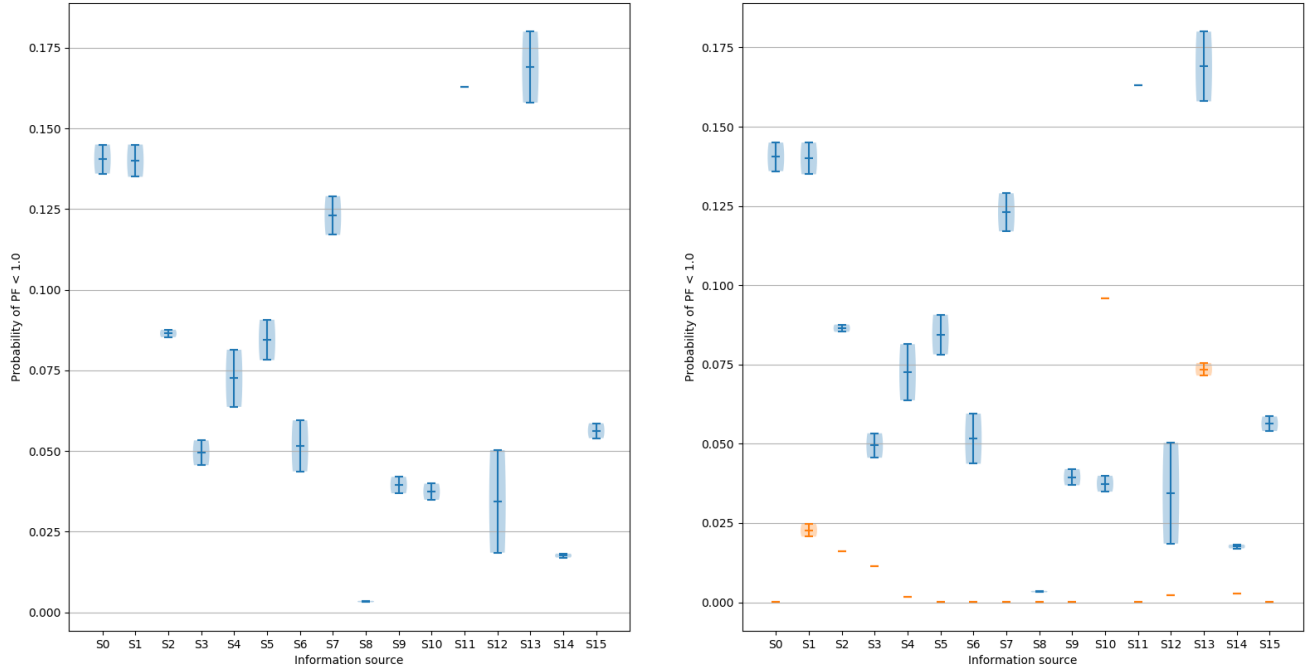
Figure 4-5.: Interval size of probability of failure ($FS < 1,0$) for a wedge formed by joint set 1 and bedding (Scenarios 1 and 2).

On the left, scenario 1. On the right comparison between scenario 1 (blue) and scenario 2 (orange)

Specifically, the largest interval size for the combination scenario is 0.004 for the piece of information 13. While in the probabilistic simulation is 0.022, which represents an increment of 240 % on the interval size. The largest interval size for the probabilistic simulation was computed for the piece of information 12 with 0.032 wide.

A similar result is obtained for the wedge formed by joint sets 1 and 2 as shown in Figure **4-6**. However, in this case, the interval size for the probability of failure of the combinations scenario is higher than for scenario 1, e.g., the largest interval size is 0.0642 corresponding to the piece of evidence 3. For scenario 1 such an interval measured is 0 because the upper and lower probability of failure are equal (0.014).

The probabilistic analysis gives the same trend. i.e., interval size is larger for the wedge formed by joint 1 and 2 rather than joint 2 and bedding. In fact, for the earlier, the size can be as large as 0.36 (36 %) for the piece of evidence 5.

In rock engineering, aside from the probability of failure (expressed as the
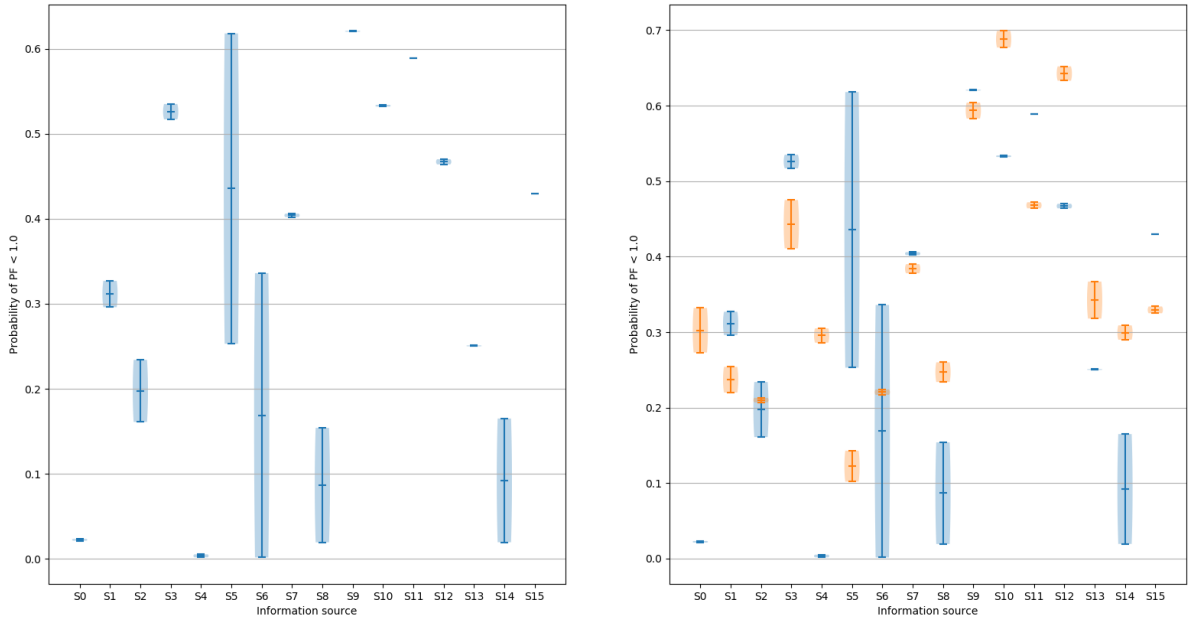
Figure 4-6.: Interval size of probability of failure ($FS < 1{,}0$) for a wedge formed by joint sets 1 and 2 (Scenarios 3 and 4). On the left, probabilistic simulated planes.

On the left, scenario 3. On the right comparison between scenario 3 (blue) and scenario 4 (orange)

probability of having wedges with a factor of safety lower than 1), there are other target factors of safety that lead the geotechnical design and recommendations. From the DSS the upper and lower probability of having any target factor of safety can be computed and can assist the decision-making process.

Hence, in figures **4-7** the upper and lower bound of the probability of having a factor of safety lower than N ($N = 1, 2, ..., 10$) for the scenario 1 for sources of information 0, 7 and 11 is depicted. Figure **4-7**a, for source 0, shows that for factors of safety higher than 1.0, the interval size is similar. Conversely, for source 7 (Figure**4-7**b), the probability intervals are larger for the combination analysis than for the probabilistic simulated orientations, but the upper bound keeps lower or similar values as the latter. Finally, **4-7**c presents, again, probability intervals from combinations larger than probabilistic simulated planes. However, here the probabilistic case would lead to lower probabilities of failure for $N > 1$.
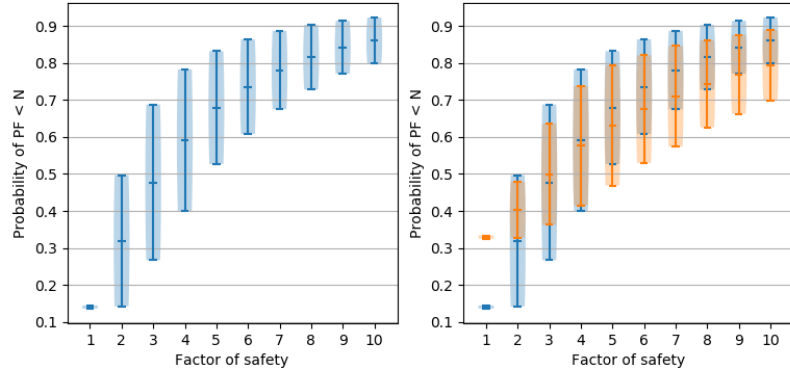
(a) 2017. Piece of information 0
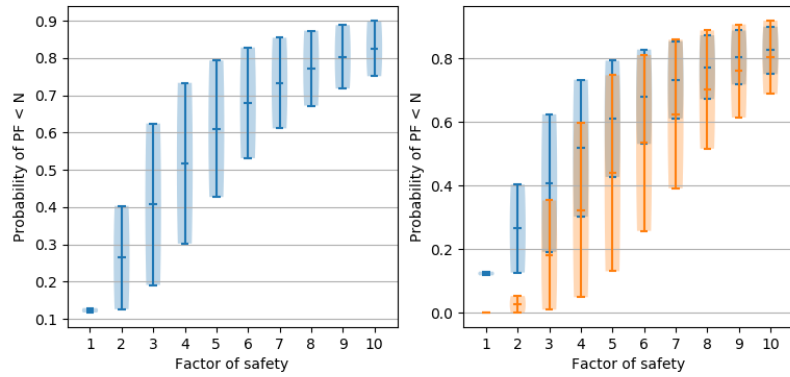
(b) 2017. Piece of information 7
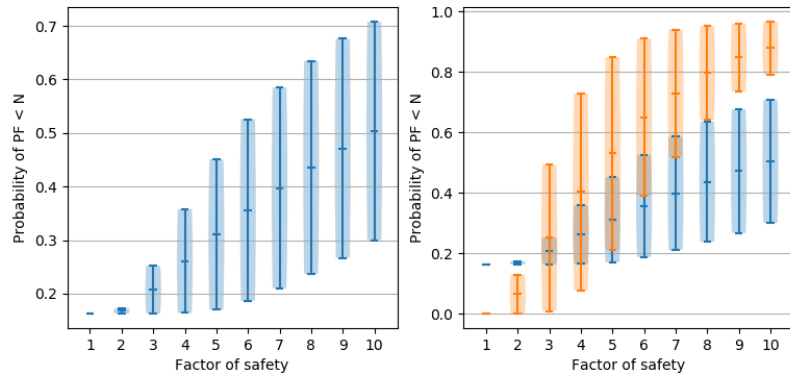
(c) 2017. Piece of information 11

Figure 4-7.: Interval size. Wedge formed by joint set 2 and bedding. Probability of having a factor of safety lower than N ($N = 1, 2, ..., 10$)

On the left, probabilistic simulated planes. On the right comparison between probabilistic simulated planes (blue) and the combination of planes (orange)

As can be seen, the interpretation of results is not straightforward, and it is not possible to define beforehand trends or expected behavior in terms of the size and shape of the resulting DSS, the analysis should be performed at each specific site, considering the particular uncertainty linked to the problem.

As mentioned above, the area between bounds of the DSS is interpreted as the uncertainty of such a DSS. Table **4-3** summarizes the areas calculated for each piece of evidence for each hypothetical scenario. Areas higher than 250 were identified as outliers and disregarded.

Table 4-3.: Area between bound of the factor of safety DSS

Piece of info	Scenario 1	Scenario 2	Scenario 3	Scenario 4
0	1.51E+01	9.14E+00	4.46E+01	1.07E+01
1	1.65E+01	8.40E+00	2.44E+01	1.15E+02
2	2.05E+01	1.37E+01	2.70E+01	8.11E+00
3	5.78E+00	8.66E+00	7.38E+00	2.68E+01
4	9.47E+00	1.05E+01	2.98E+03	4.36E+01
5	2.14E+01	1.63E+01	3.90E+01	2.06E+00
6	2.30E+01	8.32E+00	6.78E+01	2.83E+00
7	1.38E+01	1.47E+01	2.05E+01	3.92E+01
8	3.59E+01	1.55E+01	6.12E+01	4.64E+00
9	7.26E+00	1.13E+01	2.44E+01	2.78E+01
10	8.06E+00	1.11E+01	2.21E+01	2.33E+01
11	4.73E+00	5.15E+01	1.57E+01	6.22E+01
12	3.95E+00	2.12E+01	1.69E+01	2.98E+01
13	2.84E+02	1.40E+01	1.35E+01	3.73E+01
14	1.38E+01	9.17E+00	8.65E+01	4.81E+00
15	1.21E+01	9.92E+00	1.50E+01	2.81E+01

Figure **4-8** shows the area measurements summarized in violin plots. These plots show the distribution of data along with the median. From these charts, it can be inferred that the computed area is truncated at the lower side with median values very close.

Table **4-4** summarizes some descriptive statistics for the calculated area. As a relevant result, the mean area is very similar. For wedges formed by joint

set 2 and bedding, the area is 14.09 and 14.59, while for the wedge defined by joint sets 1 and 2 the areas are 22.23 and 27.21.
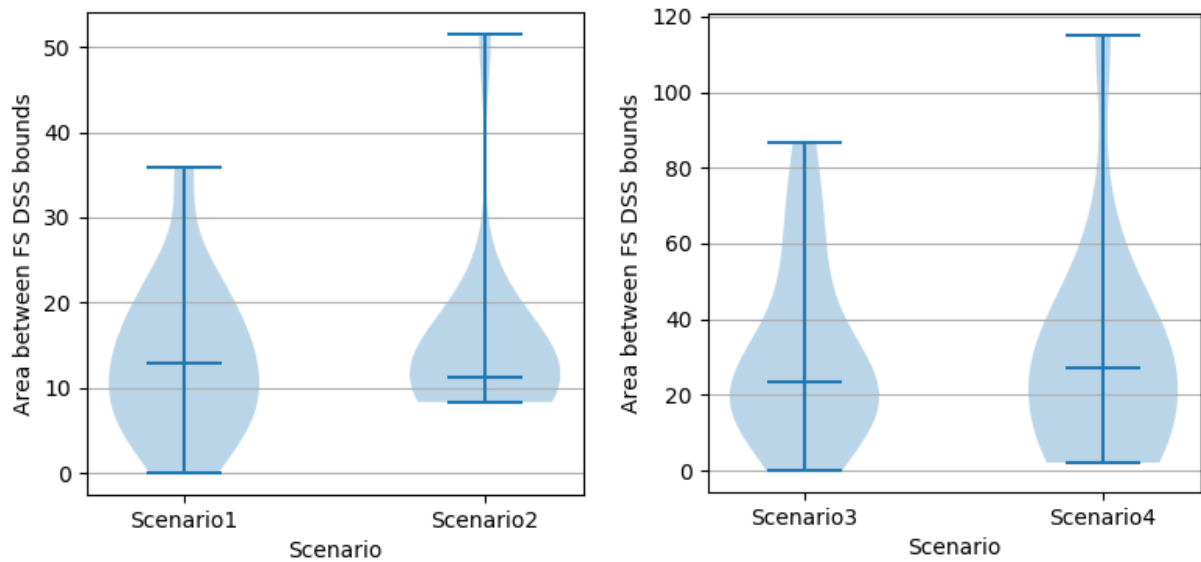
(a) Wedges formed by slope, joint set 2 and bedding plane

(b) Wedges formed by slope, joint sets 1 and 2

Figure 4-8.: Area between FoS DSS for different analyzed scenarios.

Left, wedge formed by joint set 2 and bedding. Right, wedge formed by joint sets 1 and 2

Table 4-4.: Descriptive statistics of area between bound of the factor of safety DSS

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Mean value	14.09	14.59	32.40	29.14
Standard deviation	8.26	10.14	22.23	27.71
Median	13.79	11.20	24.40	27.30

Finally, it should be highlighted that there is no correlation between the dispersion of the Kent distribution, represented by parameter κ and the area between FoS DSS. Figure **4-9** illustrates this point. This result can be explained by the fact that the area between bounds does not reflect the aleatory uncertainty of the input parameters expressed as a Kent distributed random variable, rather the uncertainty of the DSS structure.
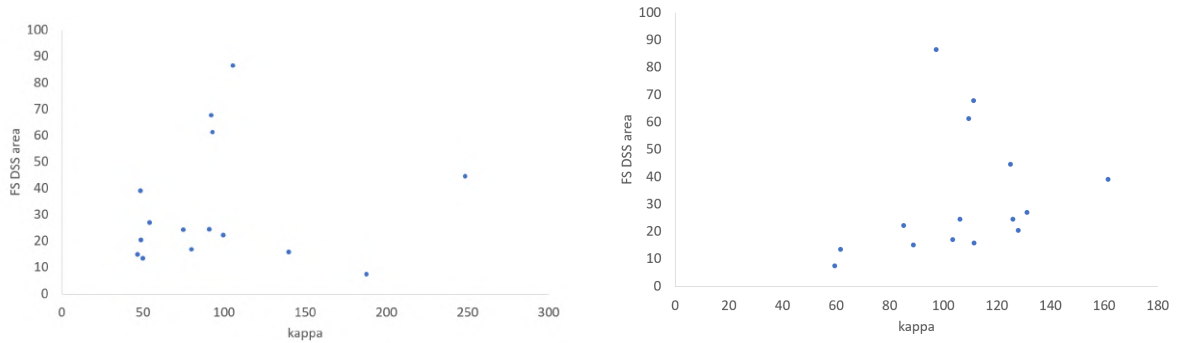
(a) Wedges formed by slope, joint set 2 and bedding plane

(b) Wedges formed by slope, joint sets 1 and 2

Figure 4-9.: Area between factor of safety DSS for different analyzed scenarios. Left, wedge formed by joint set 2 and bedding. Right, wedge formed by joint sets 1 and 2

4.5. Conclusions

This chapter proposes an alternative to assess the stability of rock wedges considering the orientation as a spherical Kent distributed random variable and the joint geomechanical parameters as DSS. This approach acknowledges the fact that in conventional rock mechanics applications, it is feasible to gather a robust database on rock joint geometry, either by using the traditional compass and tape measurements or remote sensing techniques, e.g., short-range photogrammetry.

The approach presented allows considering both the epistemic and the aleatory uncertainty leading to a bounded probability function, rather than a unique probability function. The obtained envelope bounds all the potential probability distribution feasible according to the information collected on joint strength parameters.

The proposed methodology demonstrated a high computational cost linked to the implementation of the DSS along with planes modeled as Kent distributed random variables. Nevertheless, this chapter showed that the application for the wedge model is still practical, from the computational time required. It is important to highlight that up to now, Evidence Theory analysis reported for geotechnical applications does not involve this amount of information and computations since the combination of robust datasets and limited informa-

tion has not been addressed within a geomechanical assessment framework. Results showed that for the extreme case defined by a low factor of safety (e.g., $FoS < 1,0$) the probability of failure is underestimated when only measured planes are measured. It happens because several possible orientations that lead to critical wedges do not daylight on the slope and consequently are not measured. This is overcome by performing several realizations of the orientation planes assuming a Kent distribution. Besides, as shown in Chapter 3, the Kent distribution guarantees that a more realistic simulation of the rock mass orientation is performed.

The FoS DSS calculated in this chapter showed to have a large variability of results, which was reflected in the variety of interval size for different target factor of safety (See section 4.3). This is the result of utilizing a relatively simple model (wedge stability), but under a high uncertainty context, with limited information. In other words, the stability of rock slopes is a complex problem with a high variability on input parameters so that no general result can be predicted and simulations and considerations on uncertainty should be performed for each specific case.

Despite the variability, the FoS DSS the mean and the median area within bounds calculated for different pieces of evidence had similar values. So, the mean area value might be used to compare the uncertainty linked to different wedges geometry. In this case, the uncertainty of FoS DSS formed by joint set 2 and bedding is lower than the one formed by joint sets 1 and 2.

The approach presented in this chapter yields a broader framework to decision-makers in comparison to the deterministic and probabilistic approach. The earlier does not account for explicitly for the variability of input information; while the second one does not involve the epistemic uncertainty linked to the lack of information. Consequently, although more computations are required, the DST based approach allows us to have a full overview of the model uncertainty adjusted to the quality of input information, which gives more confidence in the final design decision.

So far, the wedge geometry is explicitly defined by the joint and planes relative positions as input parameters. However, in rock slopes engineering the geometry is much more arbitrary. Hence, a more general approach should be implemented to identify such removable and unstable blocks, considering

variables as joint length and location. Therefore, in the next chapter, this analysis will be performed under an Evidence Theory framework.

A significant result stated here, is that the DSS analysis combined with Monte Carlo simulation of Kent distributed variables weights each Monte Carlo simulation by the probability assignment given to each focal element. In the following chapters and alternative for finding a weighting factor, based on the probability distribution of input information is presented. This approach might be used for updating the reliability analysis.

5. Rock slope stability with DSS and DFN

5.1. Introduction

Up to now, the stability of rock wedges has been analyzed considering the orientation variability by Kent distribution and the geomechanical information as Dempster-Shafer structures. Besides, a short-range photogrammetry system was introduced to collect information on joints geometry. However, information on the location of joints has not been accounted for, since the wedge model considers just orientation of joints and the maximum wedge based on the slope height.

The orientation, location, size, and separation of joint are uncertain in the rock mass because field measurements either by conventional surveying or image interpretation gather information only on the surface, and no information is provided on the variation with depth of these features. Therefore, the collected information can be fitted to different distributions to consider the epistemic variability of each parameter.

From this standpoint, each measurement captures only one realization of the joint setup. Consequently, it is not possible to know certainly the joints setup at a different location within the rock mass, only realizations of that setup can be considered.

The network of joints established for one realization is known as a discrete fracture network, DFN, and will be the result of a stochastic simulation of the joint orientation, separation, length, and location expressed as random variables, which parameters are established based on the collected information.

From the beginning, an essential point to highlight is that the 3D representa-

tion of fractures is performed from 1D (boreholes) or 2D(outcrop mapping). Since direct observation of the 3D fracture network is not possible at the moment. Consequently, an entirely realistic representation of the 3D fracture network is still an unresolved issue [Lei et al., 2017].

This problem has been addressed by stereological analysis of fracture networks [Berkowitz and Adler, 1998, Warburton, 1980]. The analysis consists of solving the inverse problem of finding the distribution of planes, based on the distribution of the measured traces [Berkowitz and Adler, 1998].

The main assumptions are that the geometrical parameters, as joints orientation, separation, and length are distributed as random variables. Hence, as for any probabilistic simulation, each realization of the fracture network will lead to different fracture networks. This will be explicitly shown in the next section.

For the generation of the fracture system, the three main geometrical features to be considered are [Baecher, 1983]:

1. Density: separation and frequency

2. Size: joint length, area

3. Joint orientation

Some of the simplifications are: the separation and the length of the joint are independent, which is arguably [Pahl, 1981] but contributes to the simplicity of the problem [Baecher, 1983].

The 3D distribution of size and spacing can be deducted from the traces, based on the stereological analysis. Hence, from the probability distributions of the density, size, and orientation, a stochastic realization of the discrete fracture network (DFN) can be carried out. This approach is known as the Poisson DFN model or Baecher model [Lei et al., 2017].

The seminal work developed by Einstein et al. [1978], and Baecher [1983] presents a starting point on the generation of DFN from stochastic simulation the geometrical parameters aforementioned. Early applications of fracture networks were presented by Balberg and Binenbaum [1983], Robinson [1983] to solve 2D percolation problems.

Since then, the model has been implemented for generation of fracture network for several problems, just for mentioning some examples: fractures connectivity [Bour and Davy, 1997, 1998], hydraulic properties of rock masses [Baghbanan and Jing, 2007, 2008, de Dreuzy et al., 2001a,b, Ebigbo et al., 2016, Min and Jing, 2004], modeling water flow through fractured media [Ahmed et al., 2014, Andersson et al., 1984, Berrone et al., 2018, Dershowitz and Einstein, 1988, Dershowitz and Fidelibus, 1999, Endo, 1984, Faille et al., 2016, Karimi-Fard et al., 2004, Long et al., 1982, Sandve et al., 2012], solute transport [Zhao et al., 2013], block identification and simulation [Zhang and Lei, 2013], identification of removable blocks [Zhang and Lei, 2014] and geomechanical upscaling [Elmo et al., 2014].

Even though the use of this sort of simulation is broadly used and accepted, it has some limitations linked to the simplifications during the generation process, such as homogeneous spatial distribution, planar fractures, and independence among geometrical parameters [Lei et al., 2017]. Hence, some comparisons with natural fracture networks have turned discrepancies [Belayneh et al., 2009, Lei et al., 2014].

Research to overcome these limitations focuses on the inhomogeneity of fracture spatial distribution [Billaux et al., 1989, Fadakar, 2014, Long and Billaux, 1987, Xu and Dowd, 2010]. A comprehensive state of the art of the discrete fracture network, including applications to geomechanics is presented by Lei et al. [2017].

The stochastic generation or rock wedges is based on the conceptual model of joint geometry developed by Baecher et al. [1977], built based on the following assumptions:

1. Joints are 2D circles

2. Joints center locations correspond to a Poisson process

3. Circles radii are lognormally distributed

4. Joint radius and dip are uncorrelated

5. Joint radius and spatial location are uncorrelated

The generation of the fracture pattern provides input for geotechnical analysis. However, this sort of analysis requires the identification of blocks as described below. Once blocks are defined, a first pass geotechnical analysis [El-mouttie et al., 2013] i.e., key block, rock fall, and volume estimation.
The polyhedral modelers provide the bridge between the fracture networks and the block identification required for geotechnical analysis. These modelers can be classified in [Jing and Stephansson, 2007, Zhang et al., 2010]:

1. Constructive Solid Geometry

2. Successive Space Division

3. Boundary Representation

4. Element-Block Assembling

Constructive solid geometry

Simple geometry solids (spheres, discs, or cubes) are combined by topological transformations and identification to form more complex shapes. This is the approach used by bonded particle methods.

Successive space division

Warburton [1983] presented one of the first studies on this method. In this method, a predefined geometry block is divided by adding successively new discontinuities. The procedure turns out conceptually and computationally straightforward. However, it has some limitations:

1. All generated blocks are convex

2. Artificial boundaries included to simulates excavations generates a more complex behavior than the actual one.

Hence, the method has essential limitations to model actual rock masses, primarily flow through joints [Jing and Stephansson, 2007]. Moreover, this method overestimates the amount of removable block, which bias the stability analysis result. Thus, a method capable of modeling convex and concave block shape formed by finite size joint should be presented.

Element block assembling

The method was originally proposed by Zhang et al. [2010] and Zhang et al. [2012]. This block identification method first treats all joint as infinite in the domain. Hence, at this stage, it identifies all block as convex. Subsequently, all convex blocks are analyzed further to account for the finiteness of the joints. In fact, the method identifies finite joints and merge all the convex blocks that form a concave block. Recently, the method has been extended to consider curved fractures [Zheng et al., 2015].

Boundary representation

The method is based on principles of closed surfaces and polyhedra in combinational topology to represent the block surface [Jing and Stephansson, 2007]. Conceptually and computationally this method is more complicated than successive space division. Nevertheless, it can model finite joints, which makes it more suitable for modeling actual rock masses.

Shi [1988] initially developed the method. Topological concepts explicitly introduced by Lin et al. [1987] to identify 3D rock assemblages. This proposal involves concepts from set theory. Subsequently, the method was extended to account for a block with interior holes, so that multiple connected blocks can be treated [Jing and Stephansson, 1994]. This approach is very similar to the original proposal of Shi [1988] but involves less topological concepts than Lin's approach.

The steps required to identify the block assemblage are:

1. Inputting individual joints and joint sets

2. Defining joint intersections

3. Performing edge regularization

4. Generating of appropriate data sets for block tracing

5. Block identification (block tracing)

Based on these concepts, Elmouttie et al. [2010] presents a robust and more efficient polyhedral modeler that allows modeling thousands of fractures and blocks, non-planar discontinuities, and arbitrary excavation shapes. Subsequently, the capability for handling topological errors is improved by Elmouttie et al. [2013].

Once the blocks are defined, it is possible to perform a first pass geotechnical analysis, for defining block size distribution [Elmouttie et al., 2010], key block identification [Shi, 1988] and the subsequent stability analysis by limit equilibrium.

The bottom line is that the generation of fracture network is subjected to several uncertainties, mainly because it is not possible to observe the actual 3D fracture network. Besides, given the geological origin of rock masses, both geometrical and mechanical properties are variable. Hence, stochastic DFNs are a suitable alternative for representing the rock mass fracturing pattern, to be included in the subsequent mechanical analysis of the rock mass response. Although, simplifications involve additional epistemic uncertainty to the models.

This means that a large number of realizations of the DFN is needed, considering the same probabilistic input parameters [Jing, 2003]. However, each realization is an input for the stability analysis, in which the geomechanical input parameters are uncertain as well.

Following the same approach, each realization of the DFN should be analyzed several times for different geomechanical parameters. This process can be described as a nested Monte Carlo simulation, which means that the total number of realizations would be the number of DFN realizations times the number of random simulations of the geomechanical parameters.

A direct application of this approach is not practical considering the computational cost involved for each realization of the DFN and subsequent geomechanical model.

In this context, this chapter presents an alternative to perform a stability analysis of rock masses, in which the rock mass structure is modeled by using DFN. Subsequently, the fracture system is used to perform limit equilibrium stability analysis, in which the strength parameters are considered as uncertain and modeled as Dempster Shafer structures.

In the following, a description of the methodology suggested to perform reliability assessment considering DFN and DSS is presented.

5.2.   Methodology

This section proposes a framework to explore the capabilities of performing stability analysis combining DFN along with DSS. The project resorts to the Open Pit Simulator, OPS-SIROMODEL [SCIRO, 2015]., developed by the Large Open Pit Mine Slope Stability Project. At the moment, the software is under development and is available for the project partners and researchers. The license was kindly provided by Marc Elmouttie from the Commonwealth Scientific and Industrial Research Organisation (CSIRO). For this research, OPS-SIROMODEL allowed generating the DFN, the identification of blocks, and the stability analysis of each block.

The proposed algorithm is depicted in Figure **5-1**. The general idea of the process is generating several realizations of the stochastic fracture network, then identifying the removable blocks and performing the stability analysis for a given combination of focal elements of the strength parameters, according to the input DSS. Hence, the probability of failure for a given realization of the fracture network is defined for every focal element combination. Finally, the DSS for the probability of failure is computed.

A description of the algorithm shown in Figure **5-1** is presented in the following:

1. Define the geometric parameters (separation, size, and orientation) as random variables.

2. Define the joint strength parameters as Dempster Shafer structures.

3. Establish the Q combinations of focal elements, counter m.

4. For each combination m:

5. Simulate the i realization of the discrete fracture network. K is the total number of realizations.
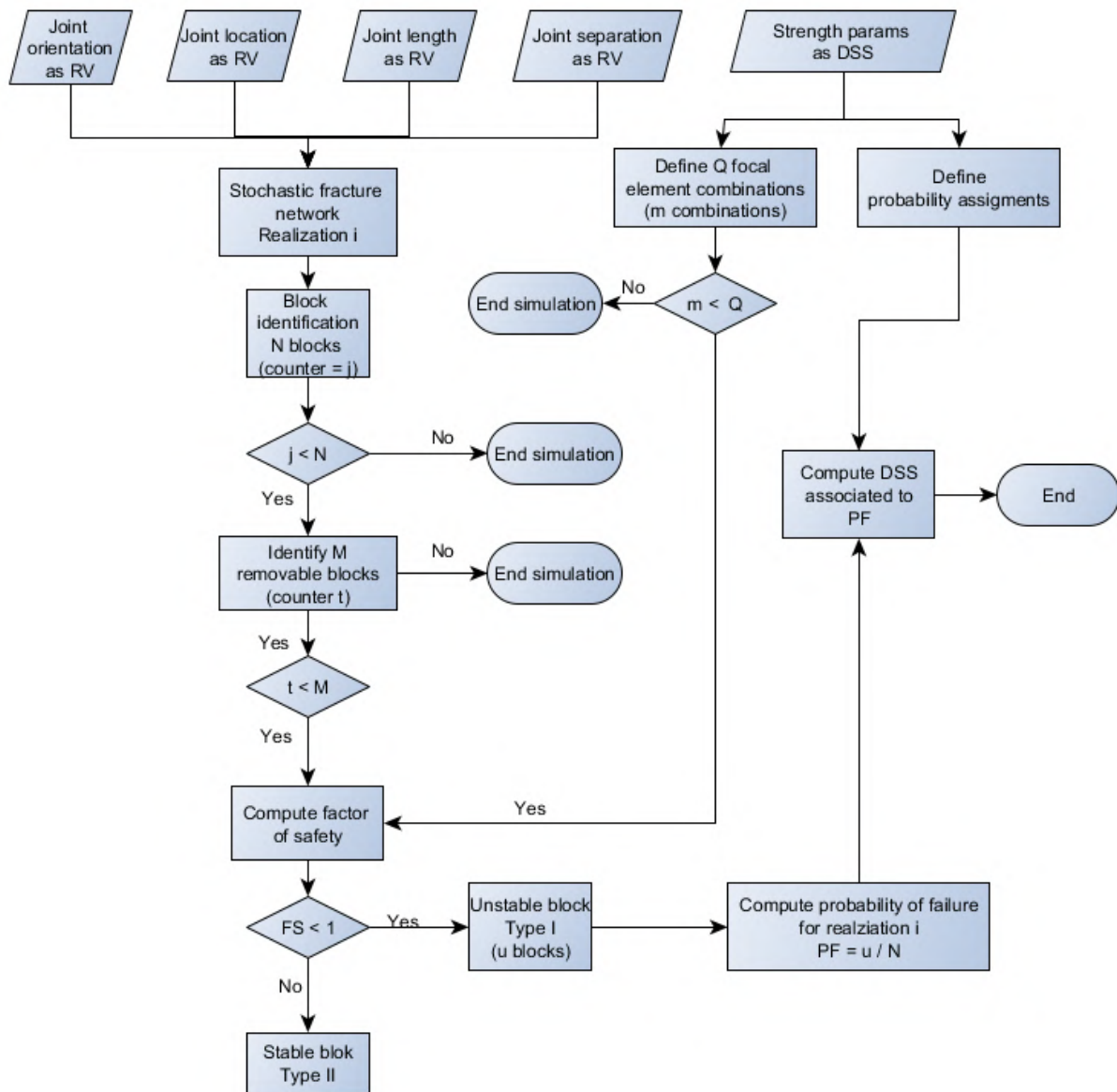
Figure 5-1.: Proposed algorithm to compute the probability function of the probability of failure

6. Identify the N blocks formed for this realization, counter j.

7. Loop through the N block to identify the M removable blocks, counter t.

8. Loop through the M removable blocks and compute the factor of safety, FS, for the corresponding combination m of focal elements.

    a) If $FS > 1$ block is stable.

    b) If $FS < 1$ the block is unstable. Update the number of unstable blocks, $u = u + 1$.

9. Compute the probability of failure PoF for DFN realization i as $PoF = u/N$.

10. Calculate the probability assignment for the PoF.

11. Go back to step 5 and update $i = i + 1$.

12. Repeat until $i > K$.

13. Go back to step 4 and update $m = m + 1$.

14. Repeat until $m > Q$.

15. Compute the Dempster Shafer structure for the PoF.

5.3.   Example analysis

Before applying the procedure described above, an example to compare different geometries by using a Fisher distribution against a Kent distribution is carried out. The process followed to compare results is the following:

1. Location of joint centroids is generated by using the DFN generator from SIROMODEL

2. Joint orientation is generated based on the algorithm implemented in this project, both for Fisher distribution and Kent distribution
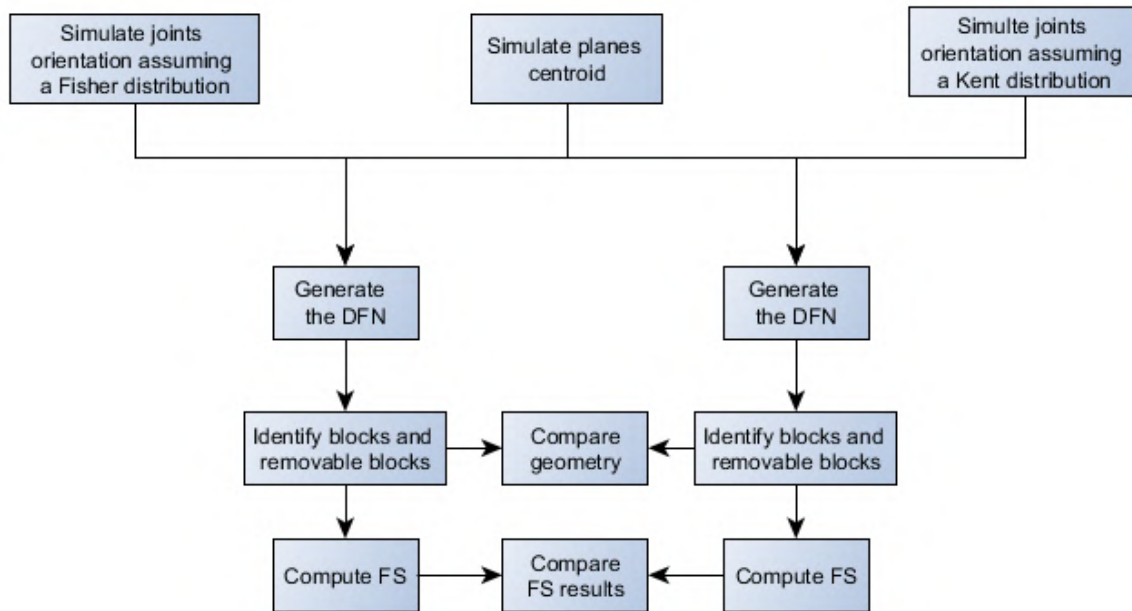
Figure 5-2.: Process followed to compare DFN geometry obtained from a
Fisher and Kent distribution

3. DFN are generated again, keeping the centroids generated in step 2.

4. Then, blocks are identified by the SIROMODEL polyhedral modeler.

5. Identify removable blocks Identify removable blocks

As mentioned above, the initial simulation process utilizes the joint planes
centroid coordinates generated by the DFN generator of SIROMODEL. Then,
those points are fixed, and the orientation is changed.

For the specific case of this example, joint centroids were modeled considering
orientation measured at Station 1 (piece of evidence 0) and included in Table
A-4. This simulation matches each centroid with a joint set. Subsequently,
joint orientations for each joint set were simulated and assigned to each cen-
troid according to the original matching. For these simulation the joint length
is also constant to guarantee that just the orientations are varying.

The model was built for a 40m long and 10m high slope a face with a strike
of 240° and a plunge of 70°. Then, 188 joint planes were simulated by the
DFN generator. Based on this, 40 DFN were simulated. The first 20 assumed

a Fisher distribution. Likewise, the next 20 simulations were carried out assuming that join set 1 and bedding follow a Kent distribution, while joint 2 keeps a Fisher distribution. These parameters were calculated as presented in Chapter 3.

Figure **5-3** shows two simulations of planes for the same Kent distributed parameters and with the same location of joints centroids. These two models have the same joint planes centroids and length, just the orientations changed, but using the same statistical parameters, κ, and β. However, both yield different DFN and consequently to different sets of removable blocks.
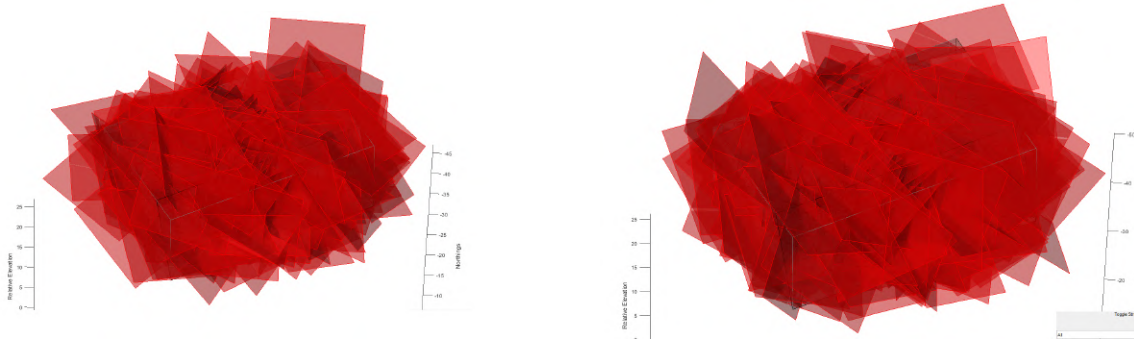
Figure 5-3.: Simulated planes with the same centroid coordinates but different orientations. Left: simulated as Fisher. Right: Simulated as Kent

Accordingly, each simulation would generate a different geometry, as shown in Figure **5-4**. There, removable blocks are highlighted in green and yellow. Key block theory [Goodman and Shi, 1985] defines finite blocks based on the intersection of semi spaces and divides them into nonremovable and removable. The latter are classified as Type I or unstable, Type 2 or stable with friction, and Type 3 or stable without friction.

From these definitions, in Figure **5-4** yellow and green blocks correspond to Type 2 and Type 3 blocks, respectively. As expected, blocks located on the horizontal upper slope correspond to Type 3, while those on the slope face are Type 2.

Beyond this classification, Figure **5-4** stresses the variability induced involving different simulations on the analysis. For instance, Figure **5-4**c has the largest type 3 blocks, while **5-4**d shows the smallest blocks on the slope.

Based on the context mentioned above, the total number of blocks along with the removable blocks were extracted from each simulation. Results are summarized in Table **5-1** and plotted in Figure **5-5**.

Table 5-1.: Kent and Fisher DFN simulation for station 1, keeping joint planes centroids

Simulation ID	Fisher		Kent	
	Number of removable Blocks	Volume of removable blocks m3	Number of removable Block	Volume of removable blocks m3
1	278	203	316	257
2	301	164	342	84
3	309	91	243	145
4	282	137	332	244
5	320	315	300	200
6	291	220	297	319
7	321	201	323	233
8	282	206	266	162
9	278	153	318	187
10	285	195	277	141
11	285	234	286	252
12	266	136	302	182
13	273	119	309	176
14	281	201	319	217
15	312	202	326	198
16	293	185	315	151
17	269	178	291	175
18	315	238	300	175
19	318	161	307	230
20	339	154	310	223
Mean	294.9	184.65	303.9	197.6
Standard deviation	20.5	49.2	23.3	51.7

The first inspection of these results shows that there is not a significant difference between the number of removable blocks and volume obtained by Kent and Fisher distributions. Indeed, the mean number of blocks are 295 and 304 for Fisher and Kent planes orientation distributions, respectively.

To verify if the mean of the 20 realizations of the number of removable blocks simulated as Fisher is the same as the simulated as Kent , a T-student test was performed. In this case, the null hypothesis is that mean samples are

(a) Simulation 1. Planes simulated as Fisher

(b) Simulation 10. Planes simulated as Fisher

(c) Simulation 1. Planes simulated as Kent

(d) Simulation 10. Planes simulated as Kent
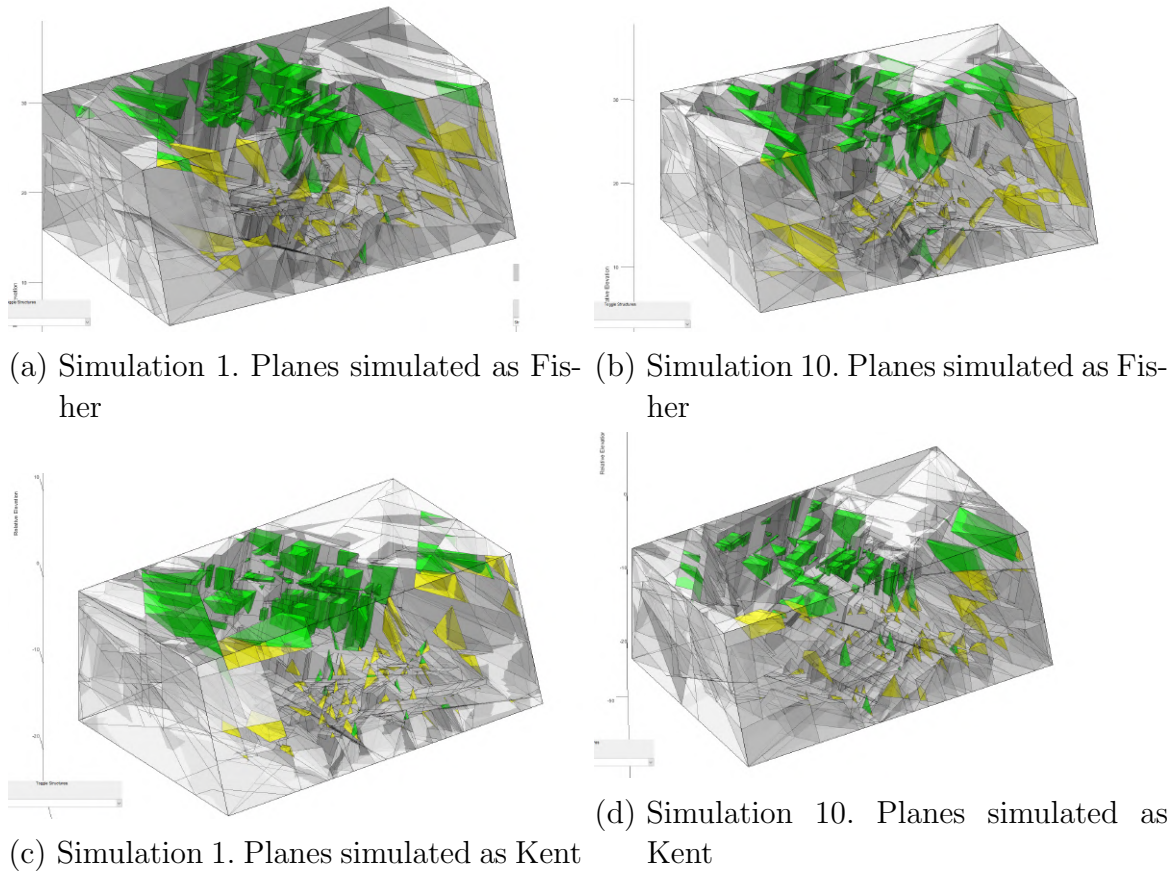
Figure 5-4.: Examples of removable blocks bounded by the DFN for Fisher and Kent distributed planes orientations. In green type III block and yellow Type II blocks

equal. Based on the information summarized in Table **5-1**, the test turned that the null hypothesis should be accepted. Hence, the samples have the same mean number of removable blocks.

This evidence allows concluding that there is no a significant difference in the amount of removable blocks resulting from simulating the orientations of the joint planes based on a Fisher or Kent distribution, for the information collected at El Pedregal Mine in station 1.

In Figure **5-6**, cross sections of the same realization of the discrete fracture network are presented. These figures illustrate the complexity of the rock joints simulation process and the uncertainty involved. In fact, each cross section might correspond to the information collected from a single measurement on a rock face. Hence, it is not possible to know for certain the distribution

(a) Planes simulated as Fisher distributed   (b) Planes simulated as Kent distributed
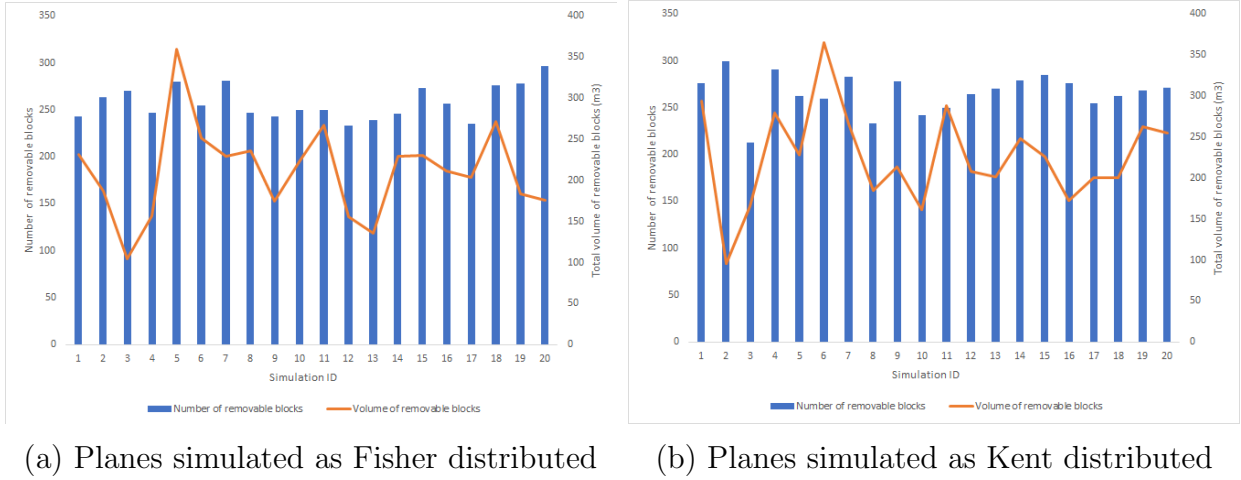
Figure 5-5.: Summary of removable blocks with the total volume for DFN simulation assuming a Fisher and Kent distribution for Station 1 at El Pedregal Mine

in depth of the rock joints by using surface mapping techniques.

Since only single measurements of the traces are possible, several realizations of the DFN are required to consider as many probable setups as possible.

For this example, and based on the results from the comparison between Fisher and Kent distribution, the procedure described in Figure **5-1** to involve the DSS with a DFN simulation process is applied for the information collected with ShapeMetrix3D (See Appendix A). The model utilizes the available information efficiently for the systematic generation of different scenarios defined according to evidence. The information on joints geometry is significantly larger than the joint strength parameters since the earlier was collected by using ShapeMetrix3D.

The orientation parameters were modeled as Fisher distributed. The mean and concentration parameters were computed as described in Section 3.2.1. Conversely, joint friction angle, and cohesion were inputted as Dempster Shafer structures as listed in Table **5-2**.

Figure **5-7** shows the procedure suggested in Figure **5-1** as two nested loops. The outer takes as input the geomechanical parameters (cohesion, friction angle, and unit weight) as DSS from and combines them as stated in Section 4.2.2.

Subsequently, the inner loop operates for a given combination of input DSS.
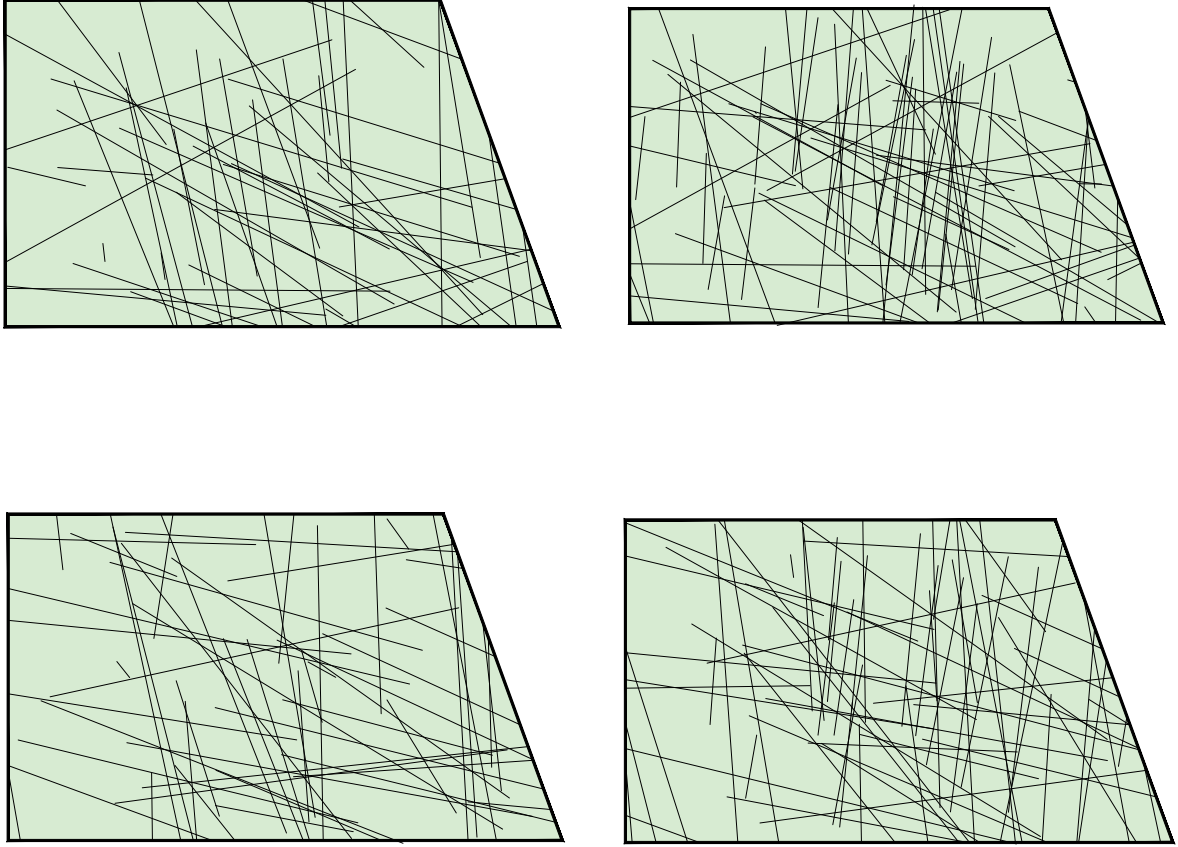
Figure 5-6.: Cross sections of the discrete fracture network

It generates a DFN from the input geometrical random variables (orientation, separation, and trace length). Then, the removable blocks are identified, and for these, the factor of safety is computed. The procedure keeps going until $N = 300$ DFN simulations were carried out.

The proportion of removable (Pr) and unstable blocks (probability of failure, PoF) is computed as follows:

$$Pr = M/N \qquad PoF = u/N \tag{5-1}$$

Here, M and N represent the number of formed and removable blocks for each realization of the discrete fracture network. u is the number of unstable

Table 5-2.: Joint strength parameters expressed as DSS

Year	Friction			Cohesion (kPa)			γ (ton/m^3)		
	Lower	Upper	P.A.	Lower	Upper	P.A.	Lower	Upper	P.A.
2017	17	20	0.33	44	89	0.5	2.59	2.60	1
	17	37	0.33	41	146	0.5			
	32	37	0.34						

P.A. stands for probability assignment

Figure 5-7.: Procedure to combine the probability of failure of several DFN realizations based on DSS

blocks (Type I).

Once the analysis is computed for all required realizations of the model, a DSS the probability of failure is built, PoF DSS, as follows.

For each combination of input parameters, defined according to the input DSS, 300 realizations of the DFN were simulated. Each one yields some removable and unstable blocks, from which the PoF is computed. Hence, 300 probabilities of failures are obtained, each one with a probability assignment, $m_{DFN}1/300$. Subsequently, the probability assignment, m, for each PoF, for this particular combination of focal elements (friction, cohesion, and unit weight) is:

$$m = m_{DFN} * m_\phi * m_c * m_\gamma \tag{5-2}$$

where m_ϕ, m_c, and m_γ are the friction angle, cohesion, and unit weight pro-

bability assignment, respectively.

The procedure mentioned above is repeated for every combination of focal elements defined by the strength parameters input DSS. For this example, 3600 DFNs were generated, which turned into the identification of 7.544.957 blocks, from which 7.063.808 were removable. Therefore, the stability of those 7.063.808 was computed. This whole computation process took about 24h of continuous work in a 32Gb RAM Intel i7 (7920) processor laptop.

Finally, a DSS of the PoF for the blocks defined by a systematic realization of the discrete fracture network is generated. The final DSS is built according to the procedure already described. Figure **5-8** shows 4 random selected DFNs out of the 3600 DFNs. In this case, green blocks are Type 3, yellow are Type 2, and red are Type 1 (Unstable blocks). Also, it is important to keep in mind that the joint planes are modeled as random variables, while in Figure **5-4**, such a length was constant.

Figure **5-9** shows the results DSS obtained for unstable blocks. It is essential to note that the DSS for the probability of failure is obtained from the combination of input DSS with different realizations of DFN.
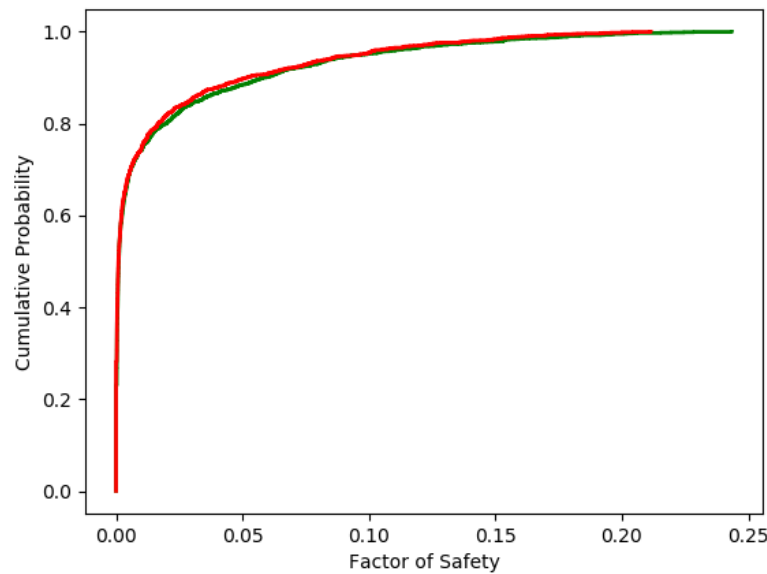
Figure 5-9.: Dempster Shafer structure built for the probability of having unstable blocks from DFN
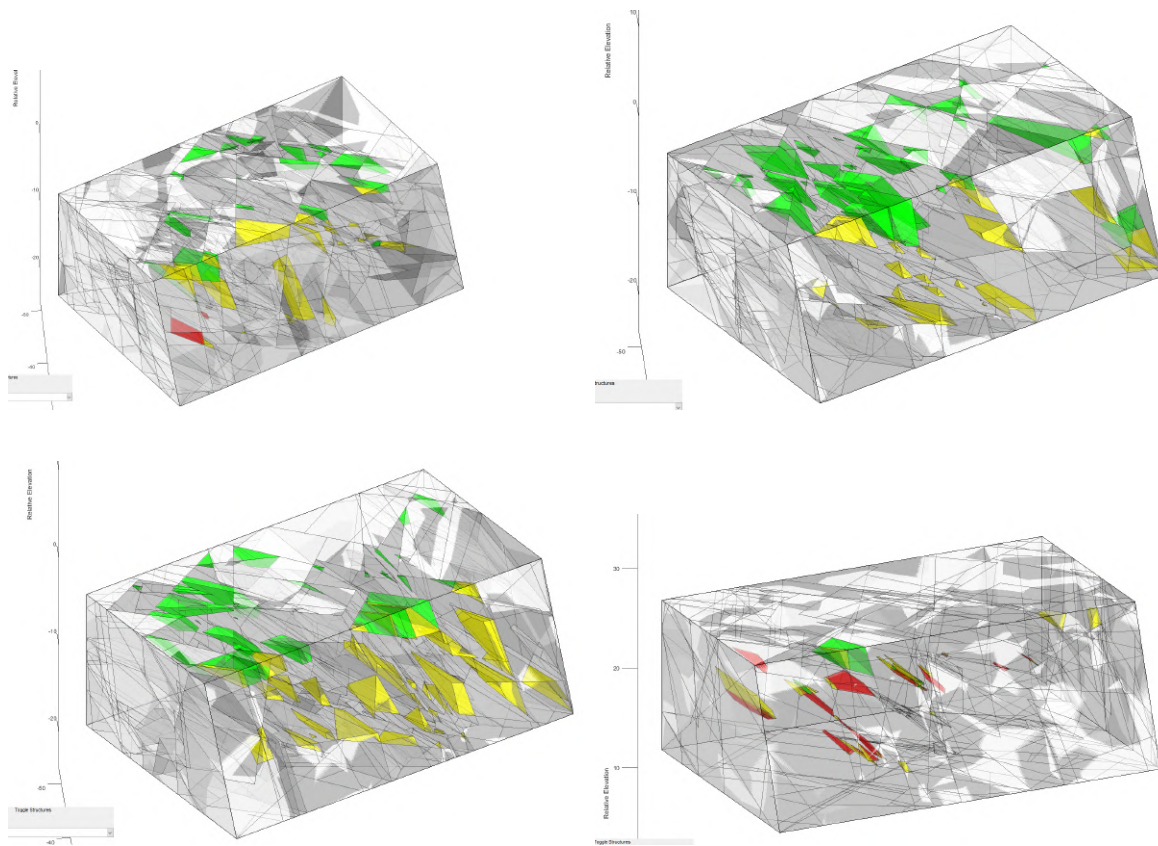
Figure 5-8.: Examples of removable blocks bounded by the DFN simulation for combination with DSS

A significant result has to do with the probability of failure computed for the different realizations. It ranges from 0 up to 25 %. This high variability is the result of the uncertainty of the rock mass geometry. However, 90 % of the simulations have less than 6 % of unstable blocks, which might provide valuable information for decision-making during the design process

In addition, the resulting PoF DSS has a very narrow band, which indicates that upper and lower bounds are very close to each other. This is an interesting result, since for this example, and after 3600 realizations of the model, the strip can be narrow enough to work with mean values instead of the DSS

5.4.   Conclusions

This chapter presented an alternative to performing a reliability analysis of rock slopes involving discrete fracture networks, DFN, and evidence theory. The integration is accomplished by modeling the geometrical parameters of the rock slope structure as random variables, including spherical distributions for joint orientations, this was possible because the short-range photogrammetry system, ShapeMetrix3D was utilized for measuring geometrical features on the slope face. On the other hand, rock joint strength parameters were modeled as Dempster-Shafer structures, DSS. The software OPS SIROMO-DEL was used to generate the DFNs, identify removable blocks, and perform the stability analysis of each block.

The comparison between assuming a Kent distribution instead of Fisher distribution showed for the example analysis that there is no statistical significance difference on the number of removable blocks between these distributions, which allowed to perform the subsequent DFN simulations using a Fisher distribution. This simplification contributed to reducing the computational cost linked to the computation process.

DFNs cross-section provided additional evidence on the complexity of the problem, given the inherent uncertainty of the rock joints geometry and the limitations to predict its evolution beyond the slope (or outcrop) face. Hence, the analysis methodology suggested here performs several realizations of the DFN for each DSS combination, to compute the response of the system, rather than 1 realization. The computation capacity restricts the number of

simulations. For the example, a total of 3600 DFN were simulated and the computation process took 24h.

A DSS for the probability of failure was built, from the number of unstable blocks from each simulation. Results from the example showed that upper and lower bounds are almost overlapping. Hence, mean strength parameters could be used instead of the input strength parameters DSS. However, it is not possible to generalize this conclusion in such a complex and uncertain framework. The shape of the resulting PoF DSS depends on the factors listed below, and each case should be evaluated carefully.

1. The uncertainty of input parameters (Focal elements of input DSS)

2. Mechanisms of block failures. i.e., if lost of contact on planes (lifting) dominates, the joint strength parameters are not relevant

3. DFN geometrical features

6. Updating the Reliability Assessment of Rock Wedges

As mentioned in Chapter 4, Shafer [1976] presented both a theory on evidence and a theory of probable reasoning. A theory of evidence, because it copes with weights of evidence. Also, a theory on probable reasoning because it allows the combination of evidence.

So far, the technique potential as a theory of evidence has been assessed in Chapter 4 (wedge stability analysis) and Chapter 5 (DFN models). However, the possibilities as a theory of probable reasoning for combining information coming from different sources have not been considered in this research. This capability of the evidence theory is very relevant for engineering projects since the information comes from different sources, i.e., expert opinion, laboratory tests, field tests. Moreover, the evidence is collected at different stages of the project.

Mining projects illustrate this point because they have project life cycles longer than 30 years and are subjected to strict supervision from public agencies. This control means that technical monitoring is performed regularly. Hence, new information is required and generated at different stages of the project. El Pedregal mine is an example of this monitoring process, which has provided several pieces of information on joints geometry and strength parameters over 20 years.

Engineers involved in the geotechnical field and specifically in rock mechanics are aware of the importance of considering different sources of information on the input parameters, given the technical and financial constraints associated with the characterization of rock masses in civil and mining projects.

A reference approach for dealing with geotechnical problems is provided by the observational method [Peck, 1969], developed for working under uncer-

tain conditions. Indeed, as shown in Chapter 5, the joints setup keeps unknown even as the excavation progress. Under this uncertain context, several scenarios should be accounted for, ranging from average conditions to the worst-case scenario. Hence, the selected approach should be able to deal with those scenarios. The probable reasoning offers such a flexible alternative to adjust the input parameters and have alternative measures when different input conditions appear.

In previous chapters, a framework to perform reliability assessment considering both epistemic and aleatory uncertainty of rock slopes has been presented within an Evidence Theory Framework, in which the input information is expressed as DSS and probability functions. Nevertheless, these inputs have been considered independent sources of information, and each one yields its independent FoS DSS. This means that the reliability analysis has not been updated as new information is available. Therefore, this chapter addresses the process of updating the reliability analysis by properly weighting and articulating the information from several pieces of evidence.

The main objective of combining information is to meaningfully summarize and simplify a corpus of data whether the data is coming from a single source or multiple sources [Sentz and Ferson, 2002]. Within the DST, there are more than ten alternatives to combine different pieces of evidence [Zargar et al., 2012] depending on the way they handle the evidence. However, none of those alternatives copes with the combination of uncertain spherical data regarding the orientation of fracture planes within the rock mass.

An expert should be able to translate the uncertainty into probabilities [Aven, 2010]. In DST, this is accomplished when focal elements and their corresponding probability assignments are defined. This definition is not straightforward for spherical data, even more, when data sets are robust.

Based on this framework, this chapter investigates the DST as probable reasoning for updating the reliability analysis of the stability of rock slopes. The chapter is divided into three parts. First, the concept of combination information is introduced, along with three rules for aggregating information. Second, an alternative to combine several pieces of evidence of spherical data on the orientation of joints is developed, given that they have an underlying probability distribution. Finally, three examples to illustrate the combination

process under DST and the proposed algorithm are presented.

6.1.   Updating the probability of failure of rock slopes

The combination of several pieces of information has been studied since the 1980s. Zadeh [1986] presents a relational view of the available combination technique back then, the Dempster's rule, attempting to clarify the concept for people familiar with artificial intelligence, AI, applications. This sort of description is handy for engineers, given the high-level background on probability theory required to understand the DST.

Subsequently, several applications have been presented. The Dempster's combination rule was utilized for clustering, using the k-nearest neighbor algorithm [Denoeux, 1995, Denœux, 2008]. The author highlights the capabilities of DST to handle ambiguity on the information, which can be seen as conflicting evidence. DST for the combination of information has been applied to sensor fusion architecture in computing [Huadong Wu et al., 2007, 2003], as a multi-criterion decision-making tool [Beynon et al., 2000b], for the selection of wireless networks [Wang and Jing, 2012], on the path followed by ships for maritime safety assessment [Talavera et al., 2013]

The problem of combining several pieces of information is addressed by Dempster-Shafer Theory of evidence and is suitable to consider the limited information. Besides, it allows handling both epistemic and aleatory uncertainty [Sentz and Ferson, 2002], even when expert opinion is involved [Torkzadeh-Mahani et al., 2018]. This type of information is assessed in this section, which deals with information from different sources, at different stages of the project.

In this regard, the information gathered at different stages or coming from different sources will be known as a piece of evidence or piece of information. As mentioned in Chapter 4, DST has not had received widespread attention in civil engineering, which leads to a few publications on the capabilities of the combination of evidence within this field. However, recently, some examples are reported in technical literature, as described below.

Several combination rules have been evaluated for aggregating two highly conflicting data sets of snow water equivalent in hydrological modeling [Zargar et al., 2012]. As for transportation, the Dempster's rule has been applied to assess different transportation modes under uncertainty from the user's standpoint. It allowed reducing the uncertainty linked to the subjectivity of expert judgment [Altieri et al., 2017].

As for structural engineering, new combination algorithms based on Dempster's rule have been introduced to assess structural damage; the technique named data fusion algorithm demonstrated to improve the structural damage diagnostic [Ding et al., 2019]. The combination of information using DST has also been used for combining the judgment of different experts on post-seismic structural damage, showing that DST is a suitable framework to represent ignorance and evidence-based assessments [Ballent et al., 2019a].

As for mining engineering, a modified combination rule, including the hierarchy process for underground water hazard assessment, was defined by Ruan et al. [2019].

Regarding geotechnical application, the mixing rule has been utilized for combining several sources of information on applications related to finite elements [Peschl, 2004, Schweiger and Peschl, 2005b].

The combination process under DST attempts simplifying or summarizing information that comes from several sources, into one set of evidence [Sentz and Ferson, 2002]. In a DST framework, each piece of information is expressed as a DSS and then combined with other pieces of information by redefining the focal elements and allocating different probability assignments. Figure **6-1** sketches the process. Here, three different DSS structures corresponding to three different pieces of information on the same input parameters are combined, and an updated DSS is obtained.

The above-described process summarizes the general idea of combining different pieces of evidence when they are expressed as DSS. The expression 'same input parameters' technically means that several sources offer a different assessment of the same discernment framework [Sentz and Ferson, 2002], assuming that those sources are independent. In this case, the discernment framework is the rock mass, and the sources correspond to information collected by different crews, or at a different location or time.

One source of
information: inputs as
random sets

Model or function.
Proper input
combinations should be
computed

Model output. Results
properly organized yield
a CDF of the model
response

Variable i

Variable i+1

Model

Stepped response (one
source)

New
Information

Updating

Updated variable i

Updated variable i+1

Model

Updated response
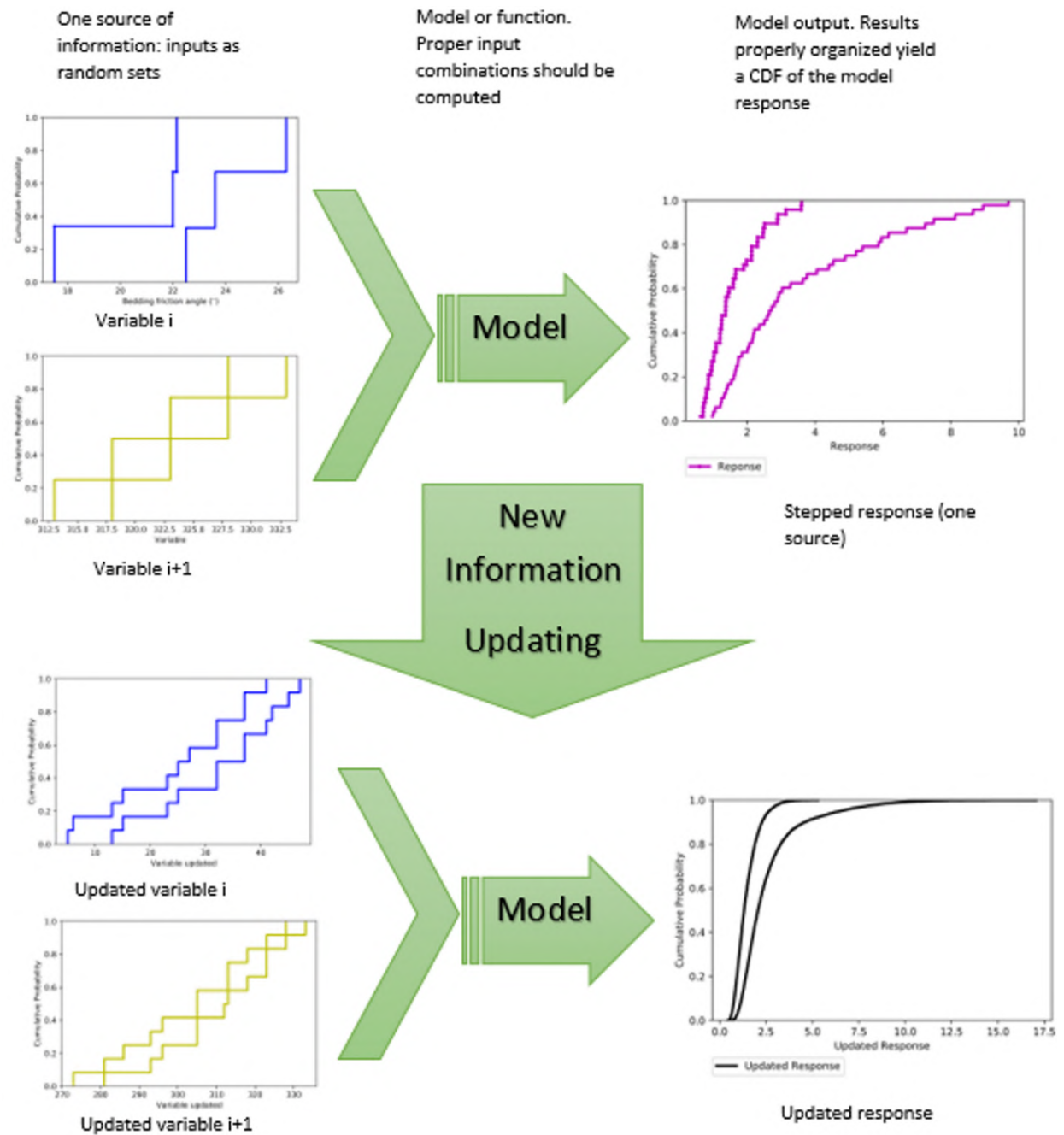
Figure 6-1.: Overview of input DSS updating process
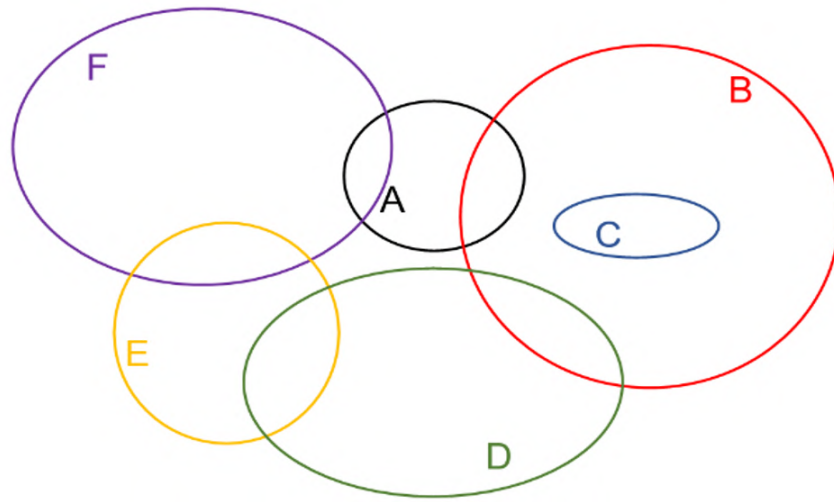
Figure 6-2.: Representation fo arbitrary evidence that consist of six pieces of evidence

The mathematical formalism behind the combination process relies on the reliability of the sources and the conflict among them. A straightforward explanation of this concept can be given in Figure **6-2**. There, six sets represent six pieces of evidence. The evidence is arbitrary since there is consistent and conflicting evidence. The piece of evidence C is consistent with B, but conflicting with the other four sources.

As for the reliability of the sources, when two samples are reliable, the conjunctive operation applies for the combination, while when one is more reliable, the disjunctive approach applies [Dubois and Prade, 1992]. Within this range, the combination rules under DST offer alternatives to combine the information; this operation was named tradeoff by Dubois and Prade [1992]. These alternatives are known as combination rules.

The original combination rule is the Dempster's rule [Dempster, 1967]; following this rule, several modifications of the rule have been proposed, depending on the way they handle and assign conflicting information. For instance, the Dempster's rule disregards the conflicting information. In the following, three of these rules are described. First, the original rule, then the Yager's rule that re-allocates the conflict and the mixing rule.

6.1.1. Dempster's rule

This technique takes out all the probability assignments linked to the conflicting evidence and distributes it among the intersecting sets by using a normalization factor K. In other words, Dempster's rule ignores the conflicting evidence and attributes it to the null set [Yager, 1987]. Accordingly, the combined probability assignment, m_{12}, of two pieces of evidence B and C, with intersection A, is computed as follows:

$$m_{12}(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - K} \tag{6-1}$$

and

$$m_{12}(\emptyset) = 0 \tag{6-2}$$

where

$$K = \sum_{B \cap C \neq \emptyset} m_1(B) m_2(C) \tag{6-3}$$

Eq. 6-1 indicates that two pieces of evidence, with both consistent (intersection) and conflicting evidence, turn into a combined set defined by the intersection of the original sets, with a probability assignment defined by the weighted product of the probability assignments of the initial sets. Hence, the conflicting information is disregarded. An example will be presented later to clarify this point.

6.1.2. Yager's rule

This rule is an essential modification of the Dempster's rule, since it does not disregard the conflicting evidence, but attributes it to the universal set. The rule was initially proposed by Yager [1987]. He stressed the importance that a combination rule allows us to update an already combined structure when new information is available [Sentz and Ferson, 2002]. This feature is

crucial in the problem addressed in this chapter since it is expected to have new information on geomechanical parameters as the project progress.

Yager's rule resorts to the concept of the ground probability assignment, $q(A)$. This ground probability assignment is attributed to the intersection A of two sets B and C, with basic probability assignments m_1 and m_2, as follows:

$$K = \sum_{B \cap C = A} m_1(B) m_2(C) \tag{6-4}$$

In order to account for the conflicting evidence, this rule allows the ground probability assignment of the null set $q(\emptyset)$ to be higher than 0. $q(\emptyset)$ is calculated in the same way as factor K in Dempster's rule (Eq. 6-4). Then, the value $q(\emptyset)$ is added to the ground probability assignment of the universal set. Hence, the probability mass of the conflicting evidence is assigned to the universal set.

6.1.3. Mixing rule

This technique generalizes the averaging operation frequently used for probability distributions, which is widespread in civil engineering. Hence, it is reasonable to consider this approach to average DSS. This combination rule has been applied to geotechnical problems, including the finite element method [Peschl, 2004, Schweiger and Peschl, 2004, 2005b]).

The method modifies the probability assignments, m_i, as follows:

$$m_i^*(A) = \frac{1}{n} \sum_{i=1}^{n} w_i m_i(A) \tag{6-5}$$

In which w_i are weights assigned according to the reliability of the sources and m_i^* is the probability assigned to the combined focal element.

In Section 6.3, three application examples will be presented to illustrate the application of these concepts.

6.2.   Proposed alternative for combining information

Above, three rules for combining evidence under the DST framework were described. The first two (Dempster's and Shafer's rule) account for the conflicting information and accordingly re-allocates the probability assignment in the aggregated DSS. The third (mixing rule) averages the probability assignments of the original DSS and assigns this average to the updated DSS. An overview of other alternatives for combining information can be found in Sentz and Ferson [2002] and Zargar et al. [2012].

Even though the presented rules consider alternatives to aggregate bodies of information, none of them addresses the issue of combining evidence of spherical data, i.e., combining several data sets on orientation planes collected from different sources or different locations. Hence, this section suggests an alternative to combine this information, based on the nature of the information and the findings reported in chapters 3 and 4. In this context, the main assumptions are:

1. The discernment framework is the rock mass

2. There are several independent pieces of information on the orientation of planes collected in such a framework at different locations and times, by different crews and by using different mapping techniques (compass, boreholes, and remote techniques)

In Chapter 3, it was shown that many sets of collected points fit better the Kent distribution instead of the Fisher distribution. Indeed, the Fisher distribution is a particular case of the Kent distribution. Besides, the probability distribution for the factor of safety of the stability of rock wedges is affected by selecting the Fisher distribution instead of the more realistic Kent distribution.

On the other hand, in Chapter 4, an algorithm for computing DSS of rock wedges, involving orientation parameters as Kent distributed variables and strength parameters as DSS. From this procedure, it was concluded that a nested reliability analysis was carried out. In the inner loop, a Monte Carlo

simulation is carried out for each combination of focal elements. Then in the outer loop, the resulting probability functions are weighted according to the DSS combination.

Within this framework, the proposed process for aggregating the sets on orientation proposes a weighting factor, w_i^* for each joint set, which is updated as new bodies of evidence are incorporated into the analysis. The weighting factor acknowledges that the orientation of the poles has a 'latent' Kent probability distribution.

Once w_i^* are computed, a DSS for the factor of safety of a wedge can be calculated and updated by systematically computing DSS for the aggregated sets of information adjusted according to the weighting coefficients.

6.2.1. Theoretical Framework

Since rock joints orientations follow a Kent distribution, each piece of evidence can be treated as a cluster with a 'latent' Kent distribution. When a new set of evidence (a new cluster with another 'latent' Kent distribution) is mapped, the two sets can be aggregated appropriately within a new cluster. The question is: how to combine those variables considering the latent probability distribution?.

Hence, the aggregation requires a mathematical technique capable of explaining pieces of evidence by combining probability distributions. The clustering based on the mixtures model provides such a capability.

When the mixture model was proposed, clustering the same feature for the input clusters, were restricted to Gaussian distributions and did not identify noise [Banfield and Raftery, 1993]. The mixtures model overcome these limitations and allows aggregating clusters with different probability distributions and recognizing outliers as noise.

These features make this approach suitable to be used along with DSS to combine sets of directional data with different 'latent' Kent distributions. This 'latent' distribution corresponds to the probability distribution of the components that will be aggregated [Kasarapu and Allison, 2015].

The mixture modeling has been used successfully for modeling random phenomena in astronomy, biology, ecology, engineering [Peel and Mclachlan, 2000],

radiation therapy [Bangert et al., 2010], and rock mass joint sets identification [Peel et al., 2001]. It has also been applied extensively in machine learning [Kasarapu and Allison, 2015]

Most of the applications deal with Gaussian mixtures as 'latent' probability distributions [Kasarapu and Allison, 2015]. Nevertheless, the model can involve different probability distributions for the components, and several algorithms have been proposed for different probability distributions [Kasarapu and Allison, 2015], e.g., t-distribution [Peel and Mclachlan, 2000], exponential [Seidel et al., 2000], and Weibull [Wang et al., 1996].

In regarding directional data, the Watson distribution has been used for clustering trough a mixture model as an alternative to the Fisher distribution [Sra and Karp, 2013]. Likewise, a clustering algorithm has been presented based on the angular Gaussian distribution and applied to fiber composites and ceramic foam [Franke et al., 2016].

As for rock mechanics applications, the mixtures model has been used for modeling rock fractures direction and paleo stress analysis using the Bingham distribution [Yamaji and Sato, 2011]. Besides, it has been utilized for clustering joint sets with the Kent distribution [Peel et al., 2001].

6.2.2. Mixtures model description

Let assume that the updated (combined) set of information consists of a p-dimensional random sample $x_1...x_n$ of size n taken from l groups. Based on this, the mixture model defines that each point is a realization of the random p-dimensional vector X, which has probability density function [Kasarapu, 2015, Peel et al., 2001]:

$$f(x : \Psi) = \sum_{i=1}^{l} w_j c_j(x : \Theta_j) \tag{6-6}$$

In which c_j corresponds to the probability function of the original data sets (pieces of evidence), w_j are the mixing proportions, $\Psi = (\pi_1...\pi_n, \theta_j^T)$, and θ_j represents the five parameters of the Kent distribution.

Traditionally, the parameters of Eq. 6-6 are estimated by minimizing the log-likelihood function of the data [Kasarapu, 2015], which requires and expectation-maximization algorithm (EM) [Dempster et al., 1977], which comprises these steps:

1. *Expectation*: The membership of each observation $x_i \in D$ in a mixture component (piece of evidence) i is defined by a responsibility index, r_{ij}, as follows:

$$r_{ij} = \frac{w_j * c(x_i; \theta_j)}{\sum_{k=1}^{l} w_k c(x_i; \Theta_k)} \qquad (6\text{-}7)$$

and

$$n_j = \sum_{i=1}^{N} r_{ij} \qquad (6\text{-}8)$$

In order to account for noise, the responsibility is modified as follows [Peel et al., 2001]:

$$r_{ij} = \frac{w_j * c_{(}x_i; \theta_j)}{\sum_{k=1}^{l} w_k c(x_i; \Theta_k) + w_0 f_0(x)} \qquad (6\text{-}9)$$

In which $f_0 = \dfrac{1}{4\pi}$ stands for a uniform distribution on the sphere.

The matrix r_{ij} is named the responsibility matrix.

n_j is the effective membership of the jth component Kasarapu [2015].

2. *Maximization*: At this step, the observations are updated by the maximum likelihood estimates Kasarapu [2015]. The log likelihood function for Ψ is:

$$logL(\Psi) = \sum_{j=1}^{n} \log \sum_{i=1}^{l} w_i * ci(x_i; \theta_i) \qquad (6\text{-}10)$$

The maximum likelihood requires to compute the root of the function Peel et al. [2001]:

$$\frac{\partial \log L(\Psi)}{\partial \Psi} = 0 \qquad\qquad (6\text{-}11)$$

As mentioned above, several alternatives for the original information probability distribution. Different alternatives are available to perform this maximization step using the Kent distribution [Kasarapu and Allison, 2015, Peel et al., 2001].

In the maximization step, the approximate moment method (Chapter 3) was utilized, as reported by Peel et al. [2001]. Hence, each observation is updated by the respective responsibility coefficient, and the mean orientation and rotation matrix are recomputed. Afterward, the parameters of the Kent distribution are recomputed as well, based on the updated parameters. With this, the responsibility matrix is recalculated.

As a result of the EM algorithm, the aggregated information is clustered, and the mixing proportion, w_i are defined. Hence, updated sets of information are considered, each one with its corresponding mixing proportion. This work proposes to use these clusters along with the mixing proportion to update the reliability assessment of rock wedges. The procedure is described below.

6.2.3. Updating the rock wedge reliability assessment by combining directional data

This section describes the methodology proposed in this research work to carry out an updating process of the reliability assessment of rock wedges, within a Dempster Shafer Theory of Evidence, taking into account the following considerations:

1. The amount of information is limited to a few data.

2. Robust databases on joints orientation are usually available and are collected either by compass or remote sensing techniques.

3. Joints orientation are random variables that follow a Kent distribution.

4. There are several pieces of evidence collected at different locations or time.

5. The discernment framework is the rock mass, and the pieces of evidence are assumed as independent.

6. Information on joints orientation is updated by using a mixture model fitted by the EM algorithm with the Kent distribution and the moment methods at the maximization step.

Let $\mathbf{c} = c_1, c_2, ..., c_j, ..., c_g$ represent a set of l pieces of evidence on fractures planes orientation collected on a rock mass. Each piece of evidence is a random variable with a Kent distribution.

On the other hand, let $\boldsymbol{\phi}$, $\mathbf{C}$, and $\boldsymbol{\gamma}$ represent DSS structures for joint friction angle and cohesion, and rock unit weight respectively.

The sets of information are combined into a mixture model by using the EM algorithm. For the expectation step, the responsibility matrix is computed according to Eq. 6-9. Consequently, 3-2 is replaced into 6-9. Eq. 3-2 is included below, for a detailed description of the Kent model and its parameters, see Section 3.2.

$$f(\boldsymbol{x}; \boldsymbol{\Theta}) = c(\kappa, \beta) exp\left\{\kappa\boldsymbol{\gamma}_1^T\boldsymbol{x} + \beta\left[(\boldsymbol{\gamma}_2^T\boldsymbol{x})^2 - (\boldsymbol{\gamma}_3^T\boldsymbol{x})^2\right]\right\} \tag{6-12}$$

The initial values for the mixing proportion are defined by the number of orientation measurements in c_j divided by the total number of orientation measurements, i.e., the portion of observation within the total number of observations, Q.

This step leads to a $Q+1$ by g responsibility matrix. It has $Q+1$, rather than Q, because it considers an additional component (cluster) for identifying the noise.

In the maximization step, for a fix cluster c_j all Q observations are updated by multiplying each unit cosine (Eq. 3-3) from each orientation by the responsibilities in column j form the responsibility matrix, hence:

$$x^*_{1_{ic}} = x_{1_{ic}} * r_{ic} \qquad x^*_{2_{ic}} = x_{2_{ic}} * r_{ic} \qquad x^*_{3_{ic}} = x_{3_{ic}} * r_{ic} \qquad (6\text{-}13)$$

Where x_1, x_2, and x_3 are the original unit cosines, while x^*_1, x^*_2, and x^*_3 are the updated unit cosines

As explained above, the responsibility coefficients measure the membership of a single observation to a cluster c. This means that if an observation is strongly explained by the cluster c, the responsibility value is high. On the contrary, if this cluster does not correctly explain an observation, the responsibility is low.

Based on the adjusted unit cosines, the Kent parameters are recomputed according to the procedure described in Section 3.2.1. This means that a new set of moment estimates $(\boldsymbol{\gamma^*_1}, \boldsymbol{\gamma^*_2}, \boldsymbol{\gamma^*_3})$.

Hence, a modified set of parameters for the Kent distributions are computed, based on the moment estimates (See section 3.2).

Then, the mixing coefficients are updated, for a cluster c_j as follows:

$$w_j = \frac{n_j}{Q} \qquad (6\text{-}14)$$

Once, both mixing proportion and Kent distribution parameters are adjusted, an iteration step h is over, and the next iteration $h + 1$ takes place, until:

$$\boldsymbol{w_h} - \boldsymbol{w_{h+1}} < t \qquad (6\text{-}15)$$

In which, $\boldsymbol{w_h}$ and $\boldsymbol{w_h + 1}$ are the mixing proportion vectors with $l + 1$ components, for iteration h and $h + 1$, respectively.

Once the EM algorithm iteration process is ready, a set of clusters with its corresponding mixing proportion is obtained. This set of information is considered the updated body of evidence on joints orientation, which will be included in the reliability analysis along with the DSS on geomechanical information.

Before describing the procedure to compute the factor of safety DSS, a remark on joint set clusters should be stated. As presented in Chapter 4, the basic probability assignment of a DSS meets three requirements:

$$m : \phi(U) \rightarrow [0, 1] \tag{6-16}$$

$$m[\emptyset] = 0 \tag{6-17}$$

$$\sum_{A \in \phi(U)} m(A) = 1 \tag{6-18}$$

In this research, the mixing proportion vector $\boldsymbol{w}$ is treated as a basic probability assignment for each updated piece of evidence on joints orientation (clusters). Hence, $\boldsymbol{w}$ is slightly modified to meet the probability assignment requirements.

The definition of the mixing proportion guarantees that:

$$\sum_{j=0}^{l} w_j = 1 \tag{6-19}$$

However, given that the number of initial clusters l is controlled by the number of available pieces of information, the EM algorithm might yield some empty clusters with very low mixing proportions. Besides, a w_0 was defined to account for the noise in the data. Hence, a residual w_r is defined as:

$$w_r = \sum_{A=\emptyset} w_j + w_0 \tag{6-20}$$

Then, w_r is proportionally assigned to the other mixing proportions, which yields a probability assignment-like weighting factor w_j^* that meets the basic probability assignment, according to DST.

$$w_j^* = w_j(1 + \frac{w_j}{1 - w_r}) \tag{6-21}$$

Under this assumption, the set of clusters defined by the EM algorithm under the mixture model for an underlying Kent distribution can be considered an updated DSS, generated from the aggregation of pieces of evidence on joints orientation that follow a Kent distribution. Each of these DSS consists of a set of orientations (clusters) with a probability assignment w_j^*.

Those DSS of directional data are combined with conventional DSS of real numbers that represent the geomechanical parameters. The procedure to perform the reliability assessment by computing the DSS of the factor of safety is the following:

1. Two updated orientation clusters j and t are selected, then the factor of safety of the wedges formed by the combination of all joint planes, according to the procedure depicted in Section 4.3.1, considering the strength parameters as DSS.

2. Compute the probability assignment as the product of the probability assignments of the corresponding focal elements of the DSS on geomechanical parameters. This step leads to a FoS DSS for clusters j and t. Each focal element of this DSS has a probability assignment PA_{FoS}

3. The probability assignment PA_{FoS} is updated as:

   $$PA_{FoS}^{jt} = w_j^* * w_t^* * PA_{FoS} \qquad\qquad (6\text{-}22)$$

   Where $w_j^* *$ and $w_t^* *$ are the weigthing coefficients for cluster j and t. Therefore, PA_{FoS}^{jt} is the updated probability assignmet for the combination of cluster j and t.

4. Repeat step 1 and 2 for every possible combination of two clusters on joints orientation

5. Construct a list of focal elements (ranges of the factor of safety) with their corresponding PA_{FoS}^{jr}

6. Build an updated DSS for the factor of safety with the information generated in the previous step. This will lead to the updated DSS on FoS accounting for the updated orientation because the probability assignments were modified to meet the requirements of the DST.

6.3. Application Examples

In this section, some practical applications of the combination of several pieces of evidence applied to the stability of rock blocks are presented, based on the information collected in El Pedregal Mine. This mine is in a sedimentary rock mass, formed mainly by alternating layers of sandstone and shales. The mine has been operated since the 1990s until nowadays. A description of the geology and collected information is presented in Appendix A.

6.3.1. Updating the stability of rock wedges by the mixing rule

This example presents a reliability assessment of wedge stability with information collected in 1997, 2011, and 2016. Table **6-1** shows the results of strength parameters along discontinuities in claystone layers. Regarding these results, some values of cohesion were measured in the laboratory. As for the samples tested in 2016, the cohesion is apparent and associated with the texture of the joint, rather than any filling within the joint. No information is provided related to cohesion from samples collected in 1997 and 2011.

Table **6-2** summarizes joint planes dip and dip direction and slope geometry. Sources of information listed in tables **6-1** and **6-2** are assumed to be independent. In this case, this assumption is reasonable, since the information was collected at different times, by different groups of people and with different tools.

Once the input information is defined, the steps followed to perform and update the probability of failure are depicted in Fig. **6-3**.

The wedge failure model selected has been developed for wedges delimited by two joint sets, the slope face and the upper slope. Based on this information, the model has 13 variables, including geomechanical and geometrical parameters. The wedge model is described in Section 3.3.1.

Table 6-1.: Mechanical properties measured on rock joints

Year	Sample	Peak cohesion (kPa)	Residual Cohesion (kPa)	Peak friction angle (°)	Residual friction angle (°)	Unit weight (kN/m3)
1997	1	32.5	2.9	22.5	22	26.2
		8.2	-	29.5	-	-
	2	-	-	26.3	17.6	23.6
		7.8	-	23.6	-	-
2011	1*	-	18	-	18	26
		-	6	-	20	-
	1	87.3	39.6	29.2	23.9	24.3
	2	89.2	25.5	34.8	32.8	23.6
	3	17.6	-	40.4	36.6	23.9
2016	1	48	-	31.8	-	24.2
	2	69	-	22	-	24.8
	3	97	-	18.5	-	24.8
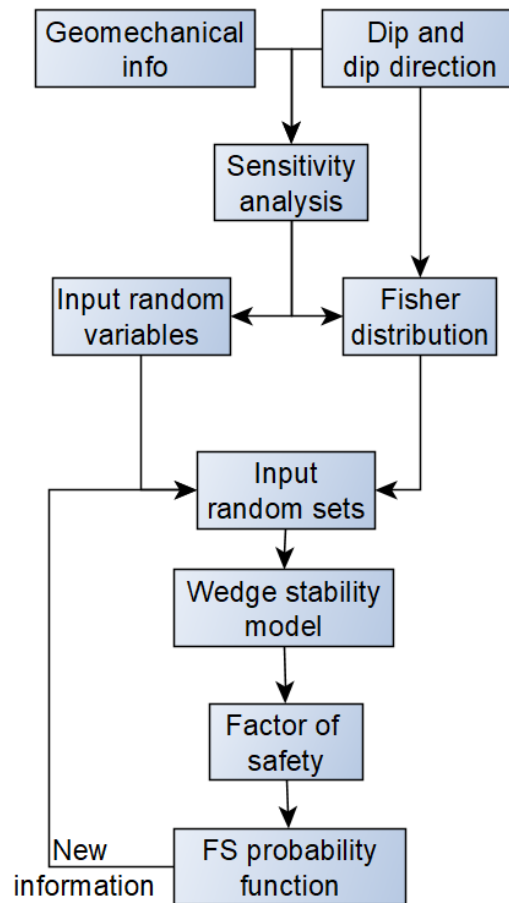
Figure 6-3.: The methodology applied to compute and update the probability of failure

Table 6-2.: Joint plane and slope average parameters

Plane	Dip direction (°)	Dip (°)	Height (m)
Bedding	306	26	-
Joint set 1	57	82	-
Slope	329	70	25

Before performing the reliability assessment, a selection of the most influencing variables is required. These variables are modeled as DSS, while the others as deterministic, which is a crucial step since the number of DSS defines the number of realizations of the model. For instance, if there are N DSS, each one with k focal elements, the number r of realizations of the model would be:

$$r = 2^N * k^N \tag{6-23}$$

The sensitivity analysis also allows knowing the variation of the response of the model with the different variables. With this information, the proper combination of input variables can be defined at each N-dimensional box to define the maximum and minimum results of the model, without computing every combination at that box. Hence, the number of computations reduces to:

$$r = 2 * k^N \tag{6-24}$$

To perform reliability assessment in geotechnical problems applying the DST, Peschl [2004] adopted a methodology based on a central difference approach, in which a sensitivity ratio (η_{SR}) is computed [EPA, 2002].

$$\eta_{SR} = \frac{\dfrac{f(x_{L,R}) - f(x)}{f(x)}}{\dfrac{x_{L,R} - x}{x}} \tag{6-25}$$

Where x is the reference value, and $x_{L,R}$ and $f(x)$ and $f(x_{L,R})$ are the outputs of the functions at those points. The sensitivity ratio is local, varying x_L a

small amount from the reference value, and general changing x_R across the whole range. Therefore, if there are N variables,(η_{SR}) should be computed $4N + 1$ times.

The sensitivity ratio is normalized, according to Eq. 6-26, then a total relative sensitivity index, $\alpha(x_i)$ is computed for each input variable (Eq. 6-27). The index varies between 0 and 1 and measures the influence of a given variable in the function. The higher $\alpha(x_i)$, the more the influence of the variable in the output.

$$\eta_{SS} = \eta_{SR}\frac{max(x_R) - min(x_R)}{x} \tag{6-26}$$

$$\alpha(x_i) = \frac{\sum \eta_{SS,i}}{\sum_{i=1}^{N} \sum \eta_{SS,i}} \tag{6-27}$$

The sensitivity analysis also provides a few realizations of the model, within a representative interval. Hence, it is possible establishing the local variation of the function for each input variable. In other words, the sensitivity analysis allows establishing if the model is increasing or decreasing concerning a variable within an interval, which reduces the number of computations to define the upper and lower bounds for combinations of focal elements.

For this example, the factor of safety was computed with Eq. 3-28, and the weighted sensitivity indexes, α_i, were computed and plotted in Figure **6-4**. The three variables with the highest α_i were selected as DSS and the rest as deterministic. Hence, the DSSs are the dip and dip direction of the bedding, as well as its friction angle. Besides, the factor of safety reduces as the dip direction increases, as shown in Figure **6-5**.

The next step is defining the DSS. For the information collected in 1997 and 2011, the DSS were defined assuming that the actual value of the friction lies somewhere between the peak and the residual friction. These values are conservative compared with the focal elements selected for 2016, in which only peak friction angle was considered, as there are no residual values reported. Table **6-3** shows the selected DSS.
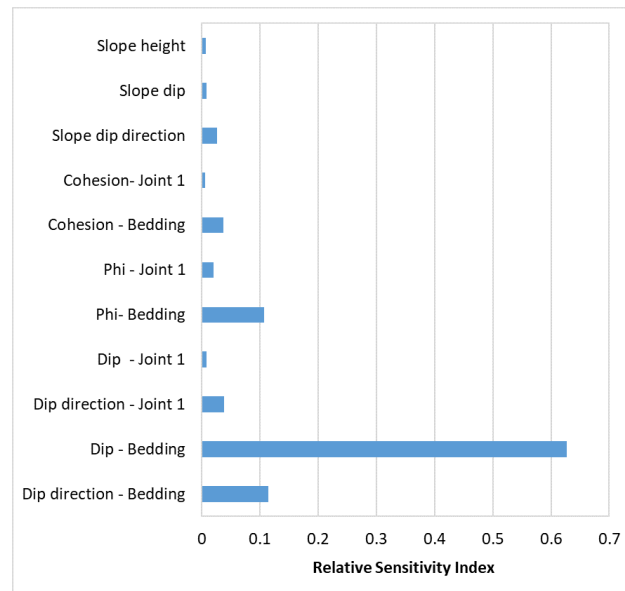
Figure 6-4.: Relative Sensitivity Index for Rock Wedge Failure

Table 6-3.: Input random sets on bedding plane friction angle

Year	Bedding plane friction angle (ş)		
	Lower	Upper	PA*
1997	22	22.5	0.33
	22	26.3	0.33
	17.5	23.6	0.34
2011	23.9	29.2	0.33
	32.8	34.8	0.33
	36.6	40.4	0.34
2016	22	31.8	0.5
	18.5	22	0.5

On the other hand, the dip direction and dip were assumed as independent variables. Hence, a DSS for the bedding dip and dip direction was defined, and the same probability assignment was given to all of them, as shown in Table **6-4**.

Subsequently, the factor of safety was computed for each piece of evidence for the combinations defined by the focal elements of each random set.

The next step is combining different sets of evidence to update the bounded probability of failure computed according to random sets. The pieces of evidence were combined as follows:

- The mixing or averaging rule was applied
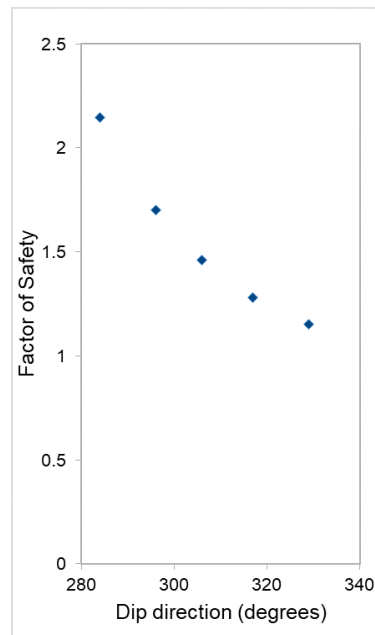
Figure 6-5.: Effect of bedding plane dip direction on the factor of safety

- At each period, the information was combined sequentially

- Then, the information at each period was combined until all sources of information were aggregated

Figure **6-6**a shows the DSSs associated with the bedding dip collected in 1997. From that, it can be concluded that the information is very conflicting since source 1 yields dip measurements up to three times higher than those reported by sources 2 and 3. The aggregated information combined by the mixing rule (plotted in black) re-allocates the probability assignment of the original focal elements. As a result, a more robust (higher number of focal elements) set of inputs is available. However, this combination rule does not account for the conflicting information; it averages the original probability assignments.

Figure **6-6**b presents a step forward in the evidence combination (updating). Here again, the bedding dip is depicted, but in this case, the updating sequence is shown. First, in black is the already 1997 combined information. Then, in green, the information from 1997 aggregated with the data collected in 2011 is shown. Finally, in red is the latter set, now updated with the

Table 6-4.: Input random sets on bedding dip and dip direction

Year	ID	Bedding					
		Dip direction (°)			Dip (°)		
		Lower	Upper	PA	Lower	Upper	PA
1997	1	313	318	0.25	27	32	0.25
		318	323	0.25	32	37	0.25
		323	328	0.25	37	41	0.25
		328	333	0.25	41	47	0.25
	2	273	281	0.25	6	13	0.25
		281	293	0.25	13	25	0.25
		293	305	0.25	25	37	0.25
		305	312	0.25	37	45	0.25
	3	286	296	0.25	5	15	0.25
		296	305	0.25	15	23	0.25
		305	313	0.25	23	32	0.25
		313	323	0.25	32	42	0.25
	4	300	305	0.25	11	16	0.25
		305	312	0.25	16	23	0.25
		312	319	0.25	23	29	0.25
		319	324	0.25	29	35	0.25
2011	1	295	300	0.25	10	16	0.25
		300	307	0.25	16	22	0.25
		307	314	0.25	22	29	0.25
		314	319	0.25	29	34	0.25
	2	281	286	0.25	11	18	0.25
		286	293	0.25	18	24	0.25
		293	299	0.25	24	31	0.25
		299	304	0.25	31	35	0.25
2016	1	288	293	0.25	18	20	0.25
		293	299	0.25	20	22	0.25
		299	304	0.25	22	24	0.25
		301	310	0.25	24	26	0.25
	2	313	318	0.25	27	32	0.25
		318	323	0.25	32	37	0.25
		323	328	0.25	37	41	0.25
		328	333	0.25	41	47	0.25

evidence collected in 2016. With this last robust random set, the updated wedge factor of safety can be computed.

Regarding the friction angle, the amount of information is much smaller, since there is only one piece of information at each period. Because of this, a smaller updated random input set is obtained as can be seen in Figure **6-7**

Figure **6-8** depicts the DSS for the wedge factor of safety computed with the evidence collected in 1997. First, results with the information considered separately are plotted in red, green, and magenta for the sources 1, 2, and 3, respectively. These curves reflect the conflict linked to the input parameters, since the factor of safety is highly variable, depending on the considered source of information. The lowest factors of safety are obtained from the first source 1997-1 (red) that has the highest dip and dip direction measurements.
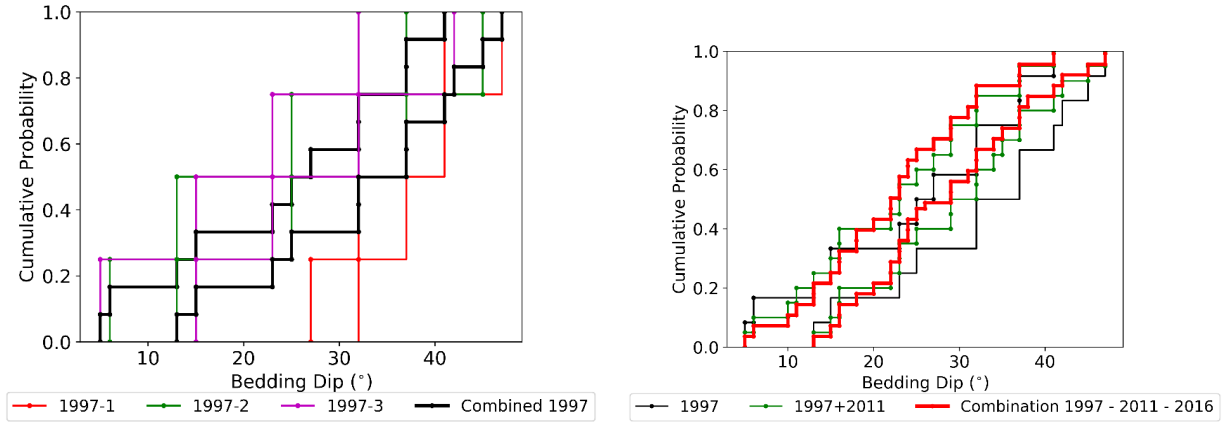
(a) Bedding dip. Aggregation of 1997 infor-
mation by mixing rule

(b) Bedding dip. Step by step updating

Figure 6-6.: Combined bedding dip angle

On the contrary, the highest factor of safety comes from the piece of evidence
1997-2 (green), which combines the lowest dip with the lowest bedding dip
direction.

Considering the evidence collected in 1997 separately might lead to a biased
conclusion on the actual stability condition of the rock wedge. If the informa-
tion first collected (1997-1) is involved, a conservative conclusion is drawn,
while 1997-2 source could be riskier. This issue justifies the need for combi-
ning information, to get a more reliable representation of the actual stability
condition.

In Figure **6-8**, the black curves represent the factor of safety obtained when
the combined input DSSs are considered. These lines balance the biased in-
formation provided by independent sources and a result in between the most
conservative and the riskiest DSS is obtained. This sort of response is directly
linked to the combination rule utilized to aggregate (update) the pieces of
evidence, which averages the probability assignments of the original DSS.

The averaging effect of the mixing combination rule is much more evident
when the input pieces of evidence are more consistent (less conflicting) than
those from 1997, which is the case of the information collected in 2011 and
2016. The resulting DSS are plotted in Figs. **6-9**a and **6-9**b, respectively. As
can be seen, the combined curves (in black) are located entirely between the
curves obtained from the original sources. Information collected in 2011 yields
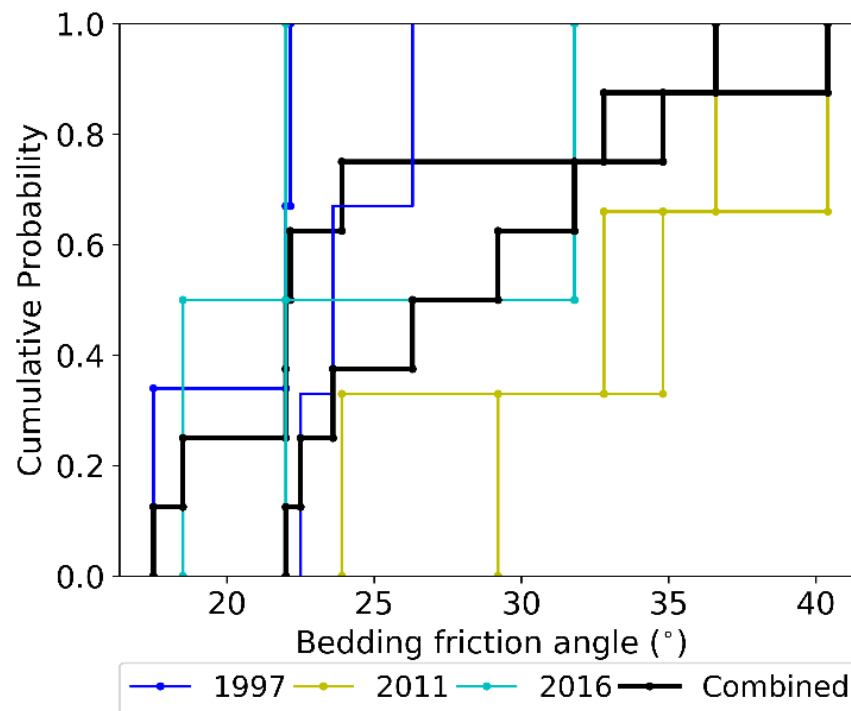
Figure 6-7.: Bedding friction angle. Aggregation of information, only one source per period
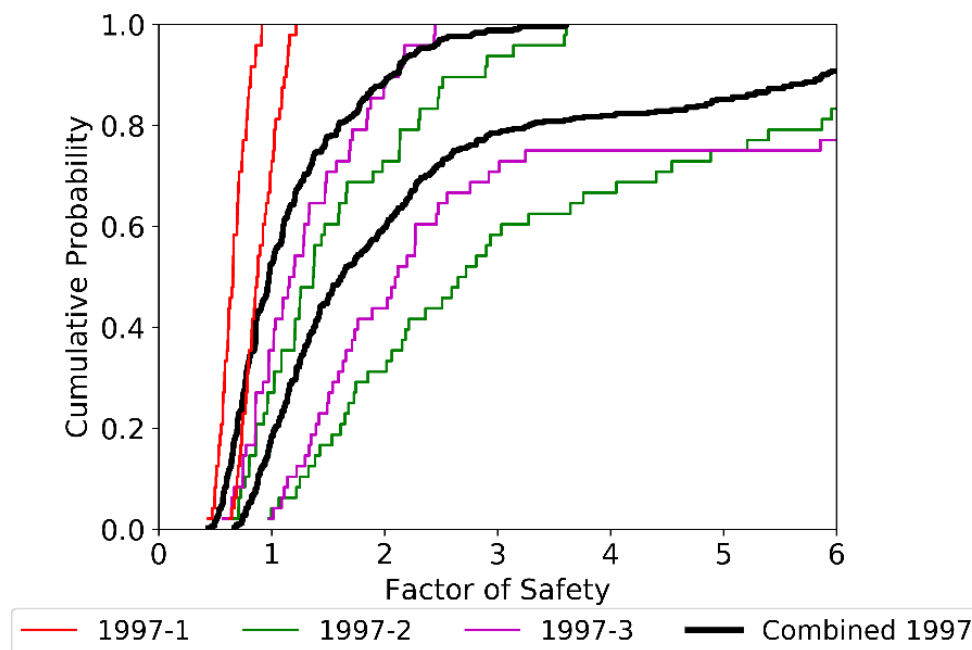
Figure 6-8.: Factor of safety DSS computed with the evidence collected in 1997
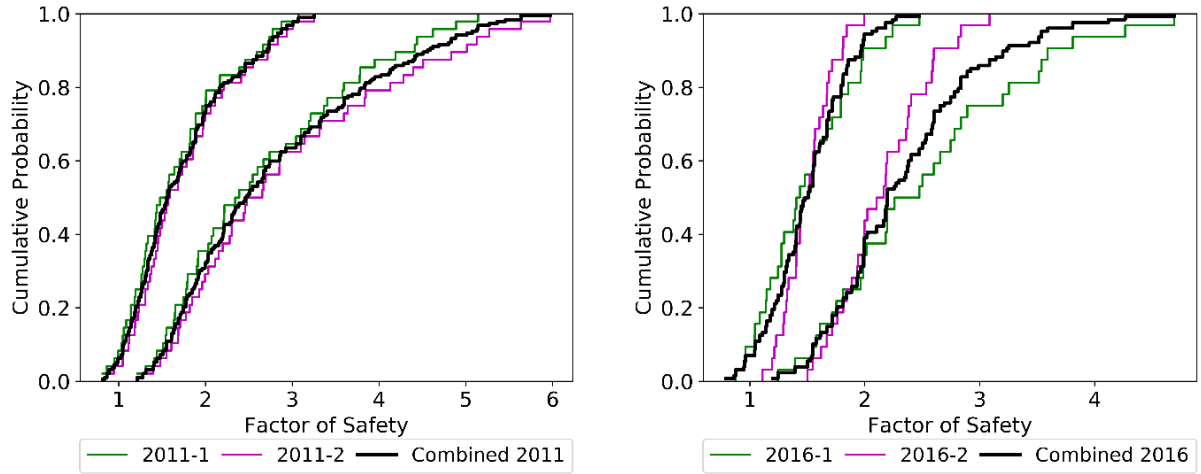
(a) Factor of safety DSS computed with the
    evidence collected in 2011

(b) Factor of safety DSS computed with the
    evidence collected in 2016

Figure 6-9.: DSS for the factor of safety

the best agreement between BDF computed from individual and aggregated pieces of evidence. These pieces of evidence are the least conflicting (See tables **6-3** and **6-4**).

So far, a noticeable difference in the cumulative probability function of the factor of safety has been presented. Now, to make it more objective, the probability of failure when considering different sources of information has been computed and included in Table **6-5** for $FS < 1,0$. This chart shows different probabilities of failure as different pieces of evidence are considered, e.g., for the information from source 1 in 1997, the probability of failure ranges between 67 % and 100 %, therefore failure is almost sure. On the other hand, the source 2 of the information collected in 1996 yields a probability of failure of 0, which is not valid, since a slope instability was already reported in 2000. When combining all available information, a probability of failure between 4 % and 25 % is computed.

6.3.2.   Updating the probability of failure of keyblocks

In this section, three combination rules for aggregating information from different sources are applied to the stability of a block by using the block theory to identify the key block and then assess its stability.

Table **6-5**.: Computed probability of failure

Year	Probability of failure (%)		
	Source	Lower	Upper
1997	1	67	100
	2	7	28
	3	5	35
	Combined	18	52
2011	1	0	8
	2	0	6
	Combined	0	6
2016	1	0	9
	2	0	0
	Combined	0	7
Combined		4	25

Goodman and Shi [1985] originally proposed the block method. The main objective is identifying key blocks from a given joint arrangement on an excavation face. Key blocks are those in a position to move, and as soon as they have done so, other blocks that were previously restrained will be liberated Goodman and Shi [1985], see Figure **6-10**a.

(a) Key blocks on a tunnel walls

(b) Half-space concept

(c) Block pyramid
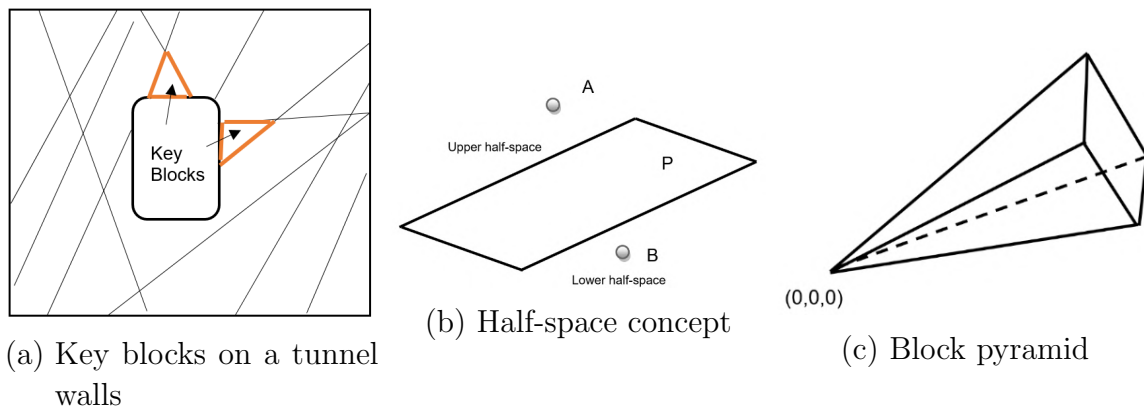
Figure 6-10.: Concepts related to the formulation of the block theory model

The method assumes planar and infinite discontinuities and rigid blocks [Haztor, 1992]. The approach distinguishes between non-removable and removable

blocks and states a systematic way to define removable blocks. First, the finite blocks are identified, and from these, removable blocks are established, which are the potentially unstable blocks.

Two concepts are essential to identify unstable blocks. Firstly, any plane in the space defines an upper and a lower half-space. In Figure **6-10**b, point A is located in the upper half-space of plane P, while point B belongs to the lower half-space. With this concept, a block can be seen as the intersection of half-spaces linked to a given joints arrangement. Secondly, a block pyramid (BP) is defined by the set of block planes, but shifted to the origin, as shown in Figure **6-10**c.

Based on these definitions, a finite block is a convex block, which BP is empty. If JP denotes the joint pyramid and the shifted excavation half-spaces are represented by EP (Excavation pyramid), then the block pyramid (BP) is defined as:

$$BP = JP \cap EP \tag{6-28}$$

And the block is finite when:

$$JP \cap EP = \emptyset = BP \tag{6-29}$$

A block is removable when it is finite, and its joint pyramid is not empty [Goodman and Shi, 1985]. Once the removable blocks have been identified, their stability is assessed. The block theory also offers a generalization of the limit equilibrium method to any block shape. The method computes the factor of safety as the ratio between resisting and acting forces. A very detailed explanation of the block theory is presented in Goodman and Shi [1985] and Haztor [1992].

Based on the three joint sets reported in El Pedregal mine, block models were built to define the key block for a given slope dip direction and dip (329/70). The model was built utilizing the software VisKBT (available at www.ddamm.org), which presents a graphical solution according to Goodman and Shi [1985].

With slope orientation defined, the key block was defined based on the mean orientation of the three joint sets reported in el Pedregal mine. As a result, the block defined by the intersection of the upper half-spaces of bedding and joint 2, with the lower upper space of joint 1 is the key block. The solution is presented in Figure **6-11**. As can be seen, the key block is defined by the half-space intersection (block pyramid) that falls entirely within the circle that represents the slope (excavation pyramid). Hence, the block is removable.

Based on this block geometry, the relative sensitivity indexes plotted in **6-12** allowed to define the joint 2 dip direction, the bedding dip, as well as, its friction angle as a random variable to be inputted as DSS. For all of them $\alpha(x_i) \geq 0,20$.

Following the sensitivity analysis, the information sources collected on these variables were utilized to define the input DSS for the key block stability model. The DSS for the friction angle is included in Table **6-3**. Table **6-6** presents the DSS for the joint 2 dip direction and the bedding dip. For the orientations, two focal elements with the same probability assignment were selected for the analysis.

DSSs for the bedding dip and friction angle are shown in Figure **6-13**. These plots were included to illustrate the difference between highly conflicting pieces of evidence (friction angle in Figure **6-13**a) and more consistent sets of information (dip direction in Figure **6-13**b). Shadowed areas on the plots attempt to show the intervals were the information is consistent (green) and conflicting (red). It is important to mention that the conflict and agreement among pieces of information cannot be read directly from this cumulative probability mass function.

The DSSs for the factor of safety are included in Figure **6-14**. The first evident result is that there is no agreement among the factors of safety computed at different years. For instance, the left bound (higher factor of safety) predicted by the evidence collected in 2011 ranges between 3.5 and 5.9, while those computed with the information from 1997 ranges between 0.9 and 5.6. The narrowest stripe between the left and right bounds is given by information from 1997. This high variability of the factor of safety comes from the variability of parameters collected at different stages of the mine excavation, which leads to a high uncertainty on inputs.
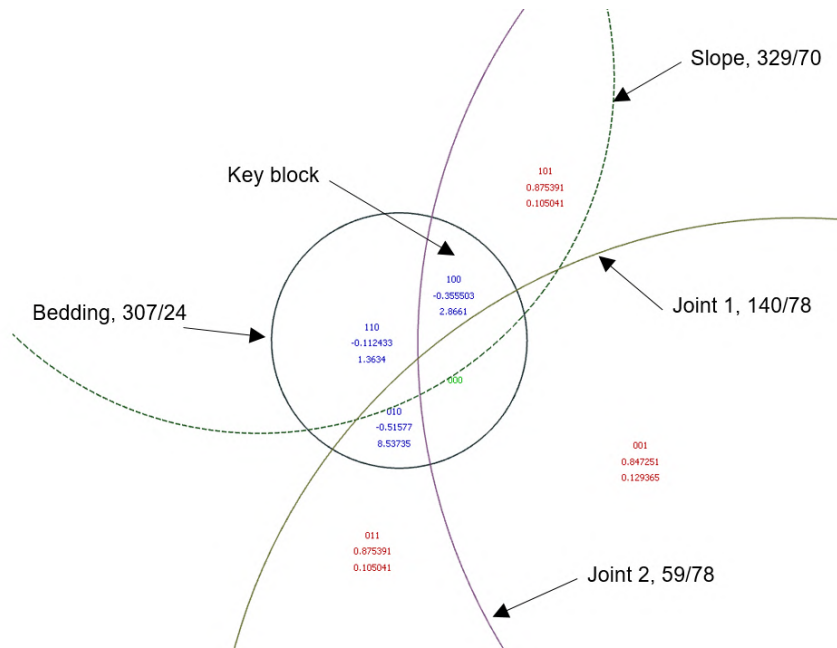
Figure 6-11.: Factor of safety DSS computed with the evidence collected in 2016

However, more important questions arise from these results: which one is the best result? What piece of information is more reliable? For this specific project, it is not possible to know which information is the most reliable, since the author was not part of the mine project since its beginning. Hence, a proper combination of the available information can give a better overview of the expected stability condition of the key block.

Therefore, the evidence was combined following the Demspter, Yager and mixing rules. The resulting DSSs are included in tables **6-7**, **6-8** and **6-9**. From that combination, it is important to mention:

1. The resulting combined DSS corresponds to the aggregation of the three pieces of information available

2. Mixing rule has the highest number of focal elements for the resulting combined structure, since it just averages the probability assignments of the original sources, and does not account for the conflicting evidence

3. Since Dempster rule disregards the probabilities linked to the conflicting evidence, it takes out all the corresponding focal elements. Hence, the
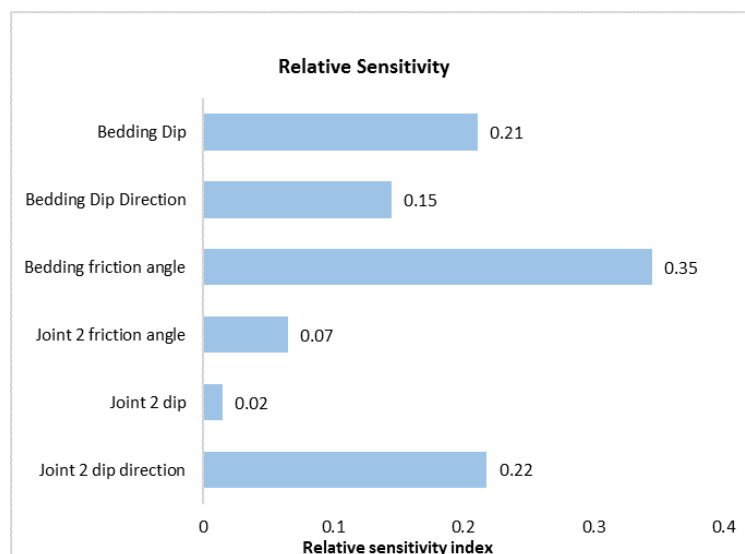
Figure 6-12.: Factor of safety DSS computed with the evidence collected in 2016

bedding friction angle combined DSS consist of only one focal element that has a probability assignment of 1. This means that information is lost during the updating process, which is an issue when computing the factor of safety since many likely combinations of inputs would not be considered.

In order to illustrate the effect of the way that the conflicting evidence is taken into account, Figure **6-15**a shows the friction angle DSSs for the Demster and Yager rule, along with the independent sources (before combining). This information is very conflicting. The aggregated DSS are entirely different. While the Dempster rule yields only one interval, the Yager rule yields a more complete DSS, since it does not disregard the conflicting evidence. Instead, it assigns that probability to the universal set. It means that there is no significant loss of information on inputs.

On the other hand, when the arbitrary information is more consistent, as the bedding dip, the difference between the Dempster's and Yager's combination rules are less dramatic. An example of this is shown in Figure **6-15**b, where the combined structures for the bedding dip are depicted. As a result, there is only a slight difference in the probability assignments, and the focal elements are the same. In that figure, the results of the mixing rule are included as

Table 6-6.: Input DS structures, defined for the orientation of planes data

Year	Source ID	Joint 2 dip direction (°)			Bedding dip (°)		
		Lower	Upper	m	Lower	Upper	m
1997	1997-1	30	48	0.5	27	47	0.5
	1997-2	20	35	0.5	5	42	0.5
2011	2011-1	54	65	0.5	11	35	0.5
	2011-2	51	68	0.5	10	34	0.5
2016	2016-1	58	88	0.5	13	38	0.5
	2016-2	35	51	0.5	18	26	0.5

(a) Bedding dip   (b) Bedding friction angle
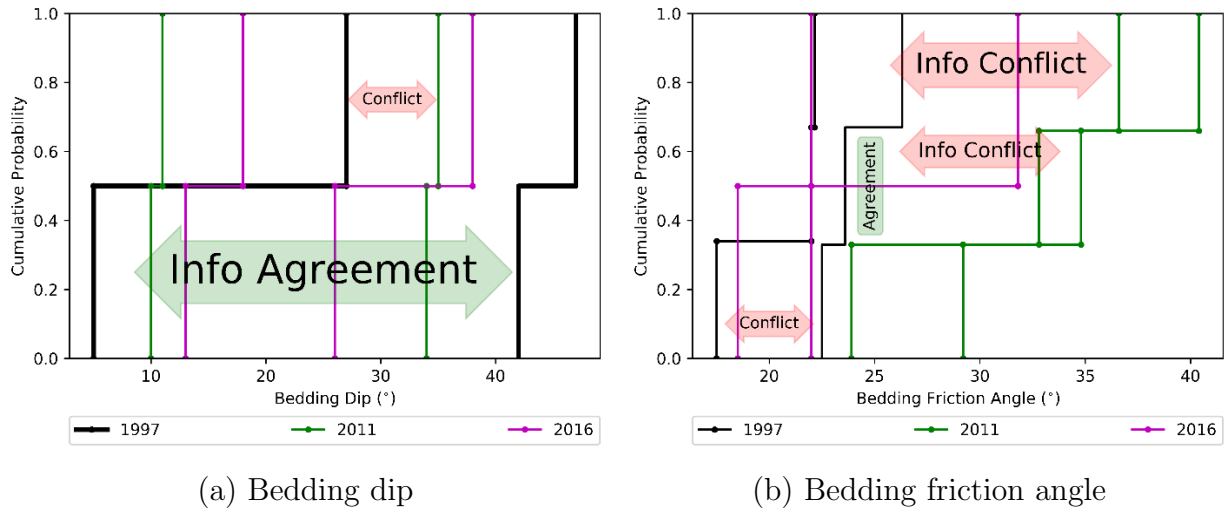
Figure 6-13.: Original DSS, before combining

well. Since it does not deal with the conflicting evidence, it consists of the same focal elements of the inputs, with a weighted probability assignment.

With this new combined input DS structures, the factor of safety DDSs were computed for the identified key block. Now, the number of computations required to reproduce the bounded probability distributions functions is higher since there are more focal elements involved. For instance, when combining with mixing rule, there are 6 focal elements for the bedding dip and joint 2 dip direction, as well as 8 for the bedding friction angle. It means that the factor of safety should be calculated 2304 times. However, since the sensitivity analysis allows to identify the trends of the factor of safety concerning
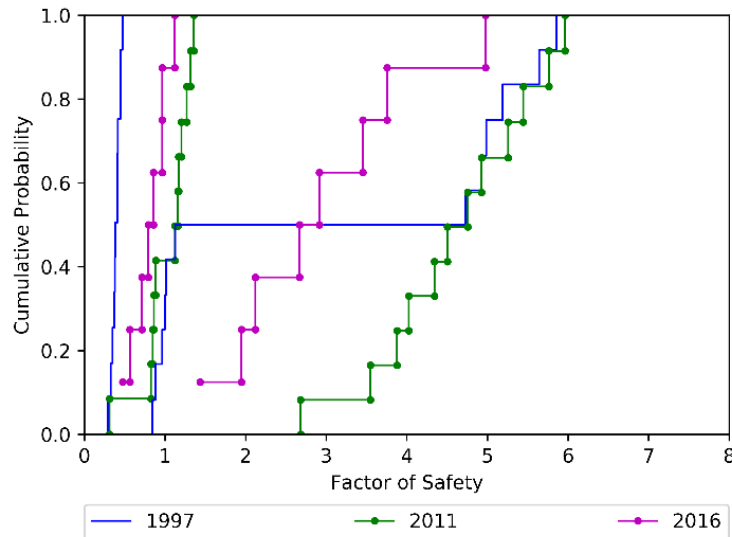
Figure 6-14.: Cumulative probability distribution function computed from information sources before aggregation

Table 6-7.: Combined DS structures according to mixing rule

Joint 2 dip direction (°)			Bedding dip (°)			Bedding friction angle (°)		
Lower	Upper	m	Lower	Upper	m	Lower	Upper	m
30	48	0.166	27	47	0.166	22.15	22.5	0.125
20	35	0.166	5	42	0.166	22	26.3	0.125
54	65	0.166	11	35	0.166	17.5	23.6	0.125
51	68	0.166	10	34	0.166	23.9	29.2	0.125
58	88	0.166	13	38	0.166	32.8	34.8	0.125
35	51	0.17	18	26	0.17	36.6	40.4	0.125
						22	31.8	0.125
						18.5	22	0.125

the variation of inputs, the number of computations reduces to 576. For the Dempster and Yager rules, the factor of safety was computed 42 and 192 times, respectively.

As for the results, first in Figure **6-16**a is depicted the DSS resulting from the mixing rule, along with the results computed from independent sources. Below 80 %, the mixing rule results attempt to find a balance among the highly variable factors of safety computed from separated sources. However, above 80 % factors of safety are much higher than those computed from independent sources. This is because more favorable combinations of parameters are possible when considering the whole body of mixed information, which

Table 6-8.: Combined DS structures according to Dempster rule

Joint 2 dip direction (°)			Bedding dip (°)			Bedding friction angle (°)		
Lower	Upper	m	Lower	Upper	m	Lower	Upper	m
54	58	0.33	27	35	0.143	23.9	26.3	1
51	54	0.33	26	27	0.143			
51	58	0.34	27	34	0.143			
			13	35	0.143			
			13	34	0.143			
			18	26	0.285			
			27	35	0.143			

Table 6-9.: Combined DS structures according to Yager rule

Joint 2 dip direction (°)			Bedding dip (°)			Bedding friction angle (°)		
Lower	Upper	m	Lower	Upper	m	Lower	Upper	m
54	58	0.25	27	35	0.125	23.9	26.3	0.0544
51	54	0.25	26	27	0.125	22	31.8	0.3911
51	58	0.25	27	34	0.125	18.5	22	0.3911
35	88	0.25	13	35	0.25	17.5	40.4	0.1634
			13	34	0.125			
			18	26	0.25			

yields higher factors of safety.

In Figure **6-16**b, results of the probability functions for the three combination rules are presented. The first interesting result is that the Yager's and Dempster's rules narrow DSS. This is a favorable result, since the narrower the stripe, the lower the uncertainty of the DSS.

The main reason for this effect is the treatment of conflicting information because the number of focal elements is reduced as the conflict increases. In fact, the narrowest stripe is given by the Dempster's rule, that disregarded most of the focal elements of the bedding friction angle, due to the highly conflicting information. As a result, the highest factor of safety computed was 2.52, which is 80 % lower than 9,9 comported with the mixing rule. In between the Dempster and the mixing rule is the Yager rule. This one has a more honest way to deal with conflict [Sentz and Ferson, 2002], since it does not disregard conflicting information, but allocates its probability assignment to the universal set. It means that even when some focal elements are missing during the combination process, the extreme values remain because these define the universal set.
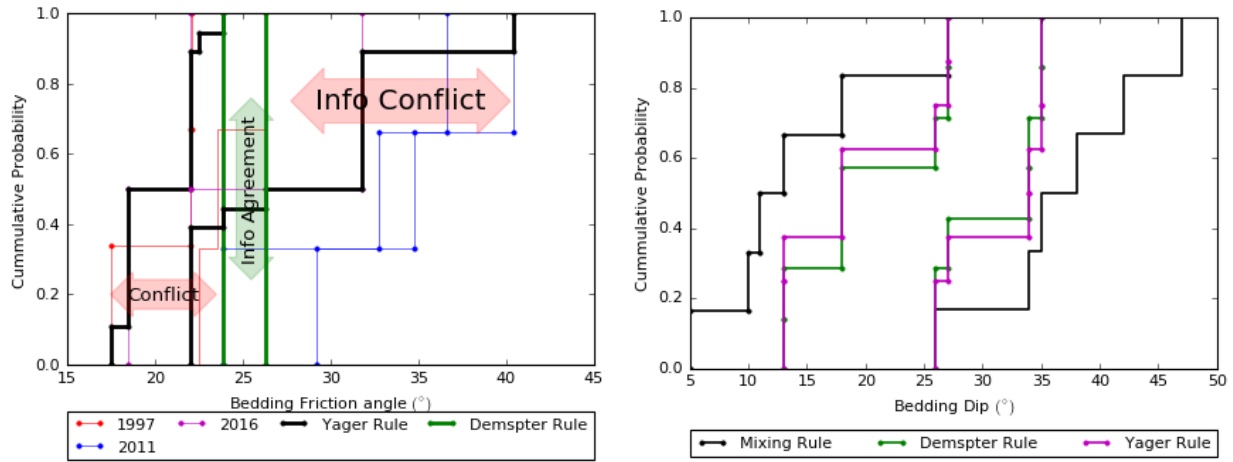
(a) Comparison between Yager and Demps-ter aggregation rules for a highly conflic-ting evidence

(b) Comparison among Mixing, Yager and Dempster for more consistent information sources

Figure 6-15.: DSS on inputs after combining

This result is evident in Figure **6-16**b at the right bound of the DSS. There, most of the results for Yager's rule are between 1 and 3. However, there are some results higher than 5 and even up to 6.6.

6.4. Updating clusters information

So far, the evidence on joint planes orientation has been considered independently, i.e., the dip and dip direction as separated variables and the combination process and reliability analysis have been performed consequently.

Therefore, in this section, two examples of the updating process proposed in this project are presented. Here, the orientation of joint planes is considered as a random variable that follows the Kent distribution. Then, the information is clustered utilizing the EM algorithm, as stated in Section 6.2. Subsequently, the nested reliability analysis described in Section 6.2.3 was followed to come up with the probability assignment.

For the first example, the rock wedge model presented in sections 3.6 and 6.4.2, for the information collected in 1997, 2011 and 2016. Again, the bedding dip, dip direction, and friction are expressed as non deterministic variables. However, unlike that example, the orientation planes are modeled as
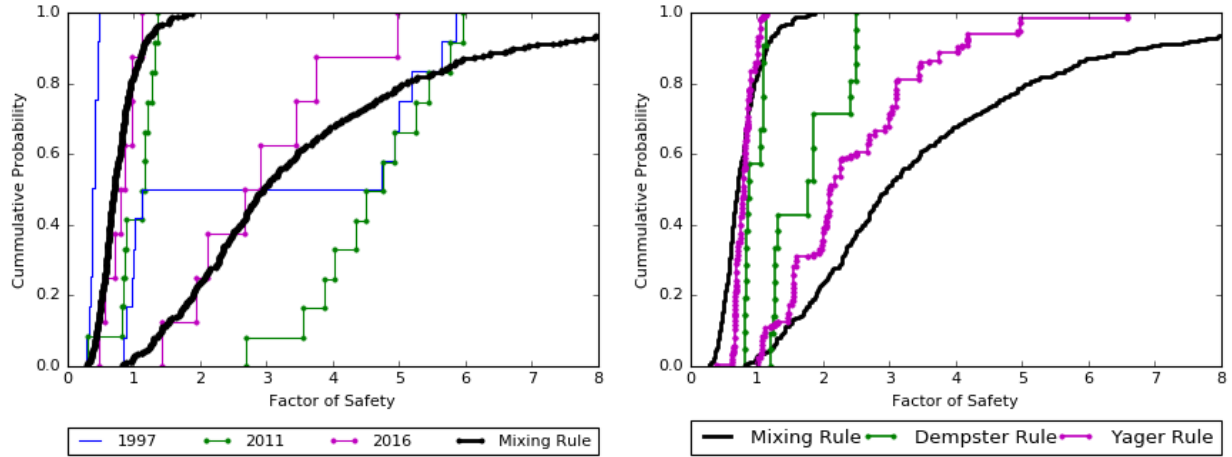
(a) Comparison between independent sources (b) DSSs computed from the combined in-
    and mixing rule   puts DSS

Figure 6-16.: DSS of the block factor of safety

Kent distributed variables. The clusters after applying the mixture model by
the EM algorithm are presented in Figure **6-17**. As can be seen, the EM al-
gorithm threw three clusters, with two outliers (white points). The weighting
coefficients w^*, adjusted according to Eq 6-21, are included in Table **6-10**.
Cluster 3 (in orange) carries most of the weight since 102 of the 136 measured
poles belong to that cluster.

Table 6-10.: Weighting coefficients for the bedding planes orientation mea-
sured in El Pedregal mine in 1997, 2011 and 2016

Cluster	w*
1	0.17
2	0.15
3	0.68

With this updated set of information on joints orientation, the factor of sa-
fety DSS is computed with the friction angles combined according to the
mixing rule (see Table **6-7**). This computation is performed according to the
procedure described in Section 6.2.

As a result, the DSS showed in Figure **6-18** was obtained. Besides, in Figure
6-19, this DSS is compared with those obtained when dip and dip direction
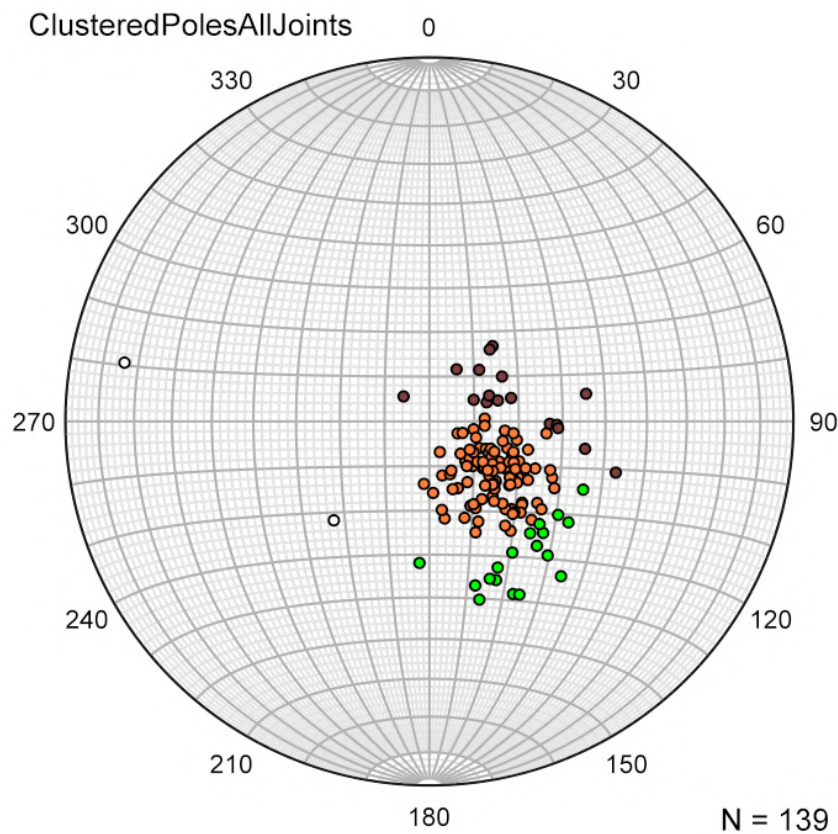
Figure 6-17.: Clustering resulting from applying the mixtures model to the information collected in 1997 for the bedding plane

were considered as independent variables.

Evidently, the proposed model reduces the area between DSS bounds. Therefore, the uncertainty linked to the FoS DSS is reduced, compared to original pieces of evidence (dotted lines in Figure **6-19**), and the combination of orientations by the mixing rule (in blue). Table **6-11** summarizes these areas. The proposed alternative means a reduction of 50 % in the uncertainty expressed as this area.

Likewise, for the updated orientation set, the FoS DSS was calculated for the friction angle DSS combined (updated) according to the Dempster's and Yager's rules, as included in tables **6-8** and tables **6-9**, respectively. The corresponding FoS DSS are depicted in Figure **6-20**.

In this case, the Dempster rule yielded the lowest area, as a consequence of the information lost during the combination process. On the contrary, Yager's
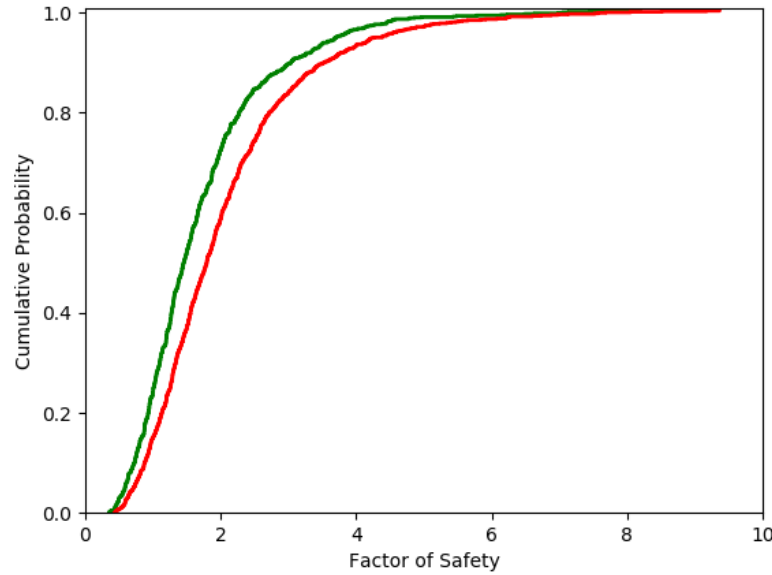
Figure 6-18.: DSS for the wedge factor of safety, obtained after updating the orientation by the EM algorithm and the friction angle by the mixing rule
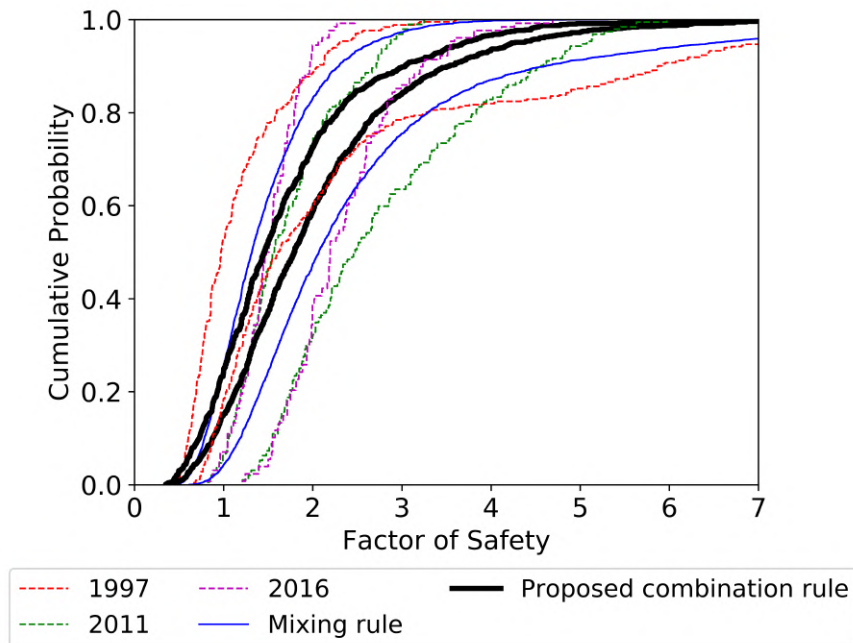
Figure 6-19.: Comparison among DSS combination rules

DSS generated from the sources before updating (dotted lines), the mixing rule assuming that dips and dip direction are independent (blue) and the proposed combination rule (black)

Table 6-11.: FoS DSS area for the evidence collected in 1997, 2011 and 2016, along with its aggregation

Evidence	DSS area
1997	1.3
2011	1.07
2016	0.83
Combined (orientation by mixing rule)	1.14
Combined (Proposed algorithm)	0.57

rule turned the most significant area, i.e., the highest uncertainty. Table **6-12**, shows these results.
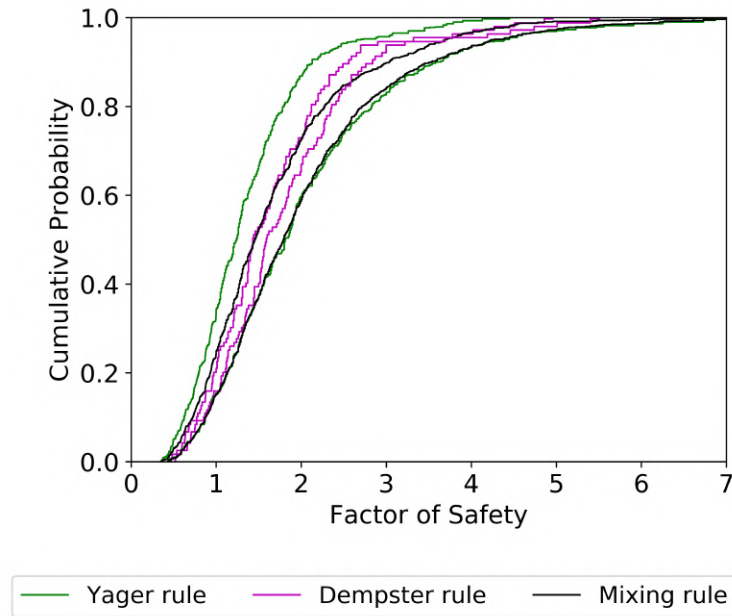
Figure 6-20.: Comparison of the wedge DSS FoS for different combination rules

Comparison among strength parameters updated according to Dempster, Yager and mixing rules when orientation is combined by the algorithm proposed in this work. Information collected in 1997, 2011 y 2016

Table 6-12.: Wedge FoS DSS area

Combination rule	DSS area
Dempster	0.18
Yager	0.724
Mixing	0.57

Comparison among strength parameters updated according to Dempster, Yager and mixing rules when orientation is combined by the algorithm proposed in this work. Information collected in 1997, 2011 y 2016

On the other hand, the proposed updating algorithm was applied to the information collected in 2017, for the pieces of evidence of joint orientations that yielded the highest uncertainty for the FoS DSS, according to Table 4-5. The assessed wedge is defined by joint set 2 and bedding plane, according to Table 4-6.

This updating process means the combination of information from different locations along the main slope, at the stations mapped by ShapeMetrix 3D. As explained in Chapter 4, each location yielded a different DSS for the factor of safety depending on the variability of the mapped planes, since the geomechanical input DSS were the same at each location.

Hence, this example attempts to systematically combine the different sets of information, acknowledging that they have a Kent distribution. This is accomplished by applying the combination algorithm proposed in this research. Hence, the updated sets of orientation data, for the wedge failing along the bedding plane (See Scenario 1 in Section 4-4), by using the EM algorithm are plotted in figures **6-21**a and **6-21**b. Again, outliers are shown in white. The corresponding weighting factors (adjusted mixing proportion) are included in Table **6-13**.

Based on these updated pieces of evidence, along with the DSS for the strength parameters collected in 2017, the updated DSS for the factor of safety was computed, as shown in Figure **6-22**.

In this figure, the updated FoS DSS (in black) is included, along with the DSS resulting from computing the DSS for each piece of evidence separately. When information is considered separately, it is evident the variability of the DSS at different locations, as stated in Chapter 4. Hence, combining the

information systematically into a representative DSS is a challenging task, but also because orientation data follow a Kent distribution.

This task is accomplished by the suggested algorithm, which leads to the FoS DSS in black in Figure **6-22**. This DSS attempts to represent the stability of wedges along the slope, accounting for both, the aleatory and the epistemic uncertainty linked to the complexity of rock masses.

(a) Clustering resulting from applying the mixtures model to the information collected in 2017 for the Joint 1

(b) Clustering resulting from applying the mixtures model to the information collected in 2017 for the bedding
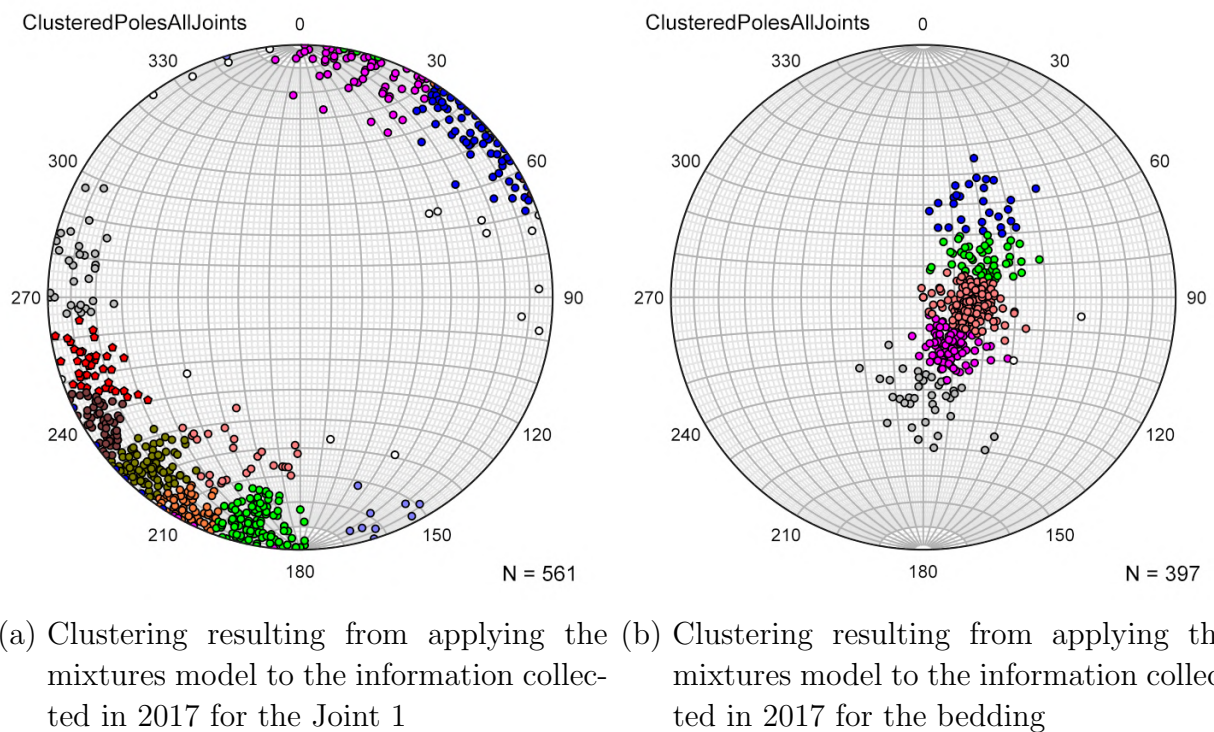
Figure 6-21.: DSS of the block factor of safety

6.5.   Conclusions

The capabilities of DST for combining evidence as a tool for updating the reliability assessment of the stability of rock wedges were studied in this chapter. First, existing alternatives for combining evidence under DST, based on the distribution of the conflict were considered. Then, an alternative for combining orientation data expressed as Kent distributed random variables, within mixture models aided by the EM algorithm was suggested. The

Table 6-13.: Weigthing coefficients for pieces of information on joint 2 and bedding orientations. Computed according to the algorithm proposed in this reserach

Joint 1		Bedding	
Cluster	w*	Cluster	w*
1	0.0156	1	0.146
2	0.18	2	0.077
3	0.115	3	0.261
4	0.115	4	0.089
5	0.106	5	0.427
6	0.111		
7	0.0806		
8	0.048		
9	0.168		
10	0.0607		

latter is the main contribution of this chapter. Finally, the concepts and algorithms presented were illustrated by practical examples with the information gathered at El Pedregal Mine. From this analysis, the following conclusions were drawn:

- When the pieces of evidence were considered separately, noticeable differences in the FoS DSS were obtained, which justified the need for aggregating the information into a representative set of updated evidence.

- Very conflicting pieces of evidence have a noticeable effect on the cumulative probability functions obtained. Indeed, Dempster and Yager combination rules lose information during the combination. This effect is dramatic in Dempster's rule because it disregards the focal elements from the conflicting evidence.

- The mixing rule balances the results obtained from individual sources, without dealing with conflicting information.
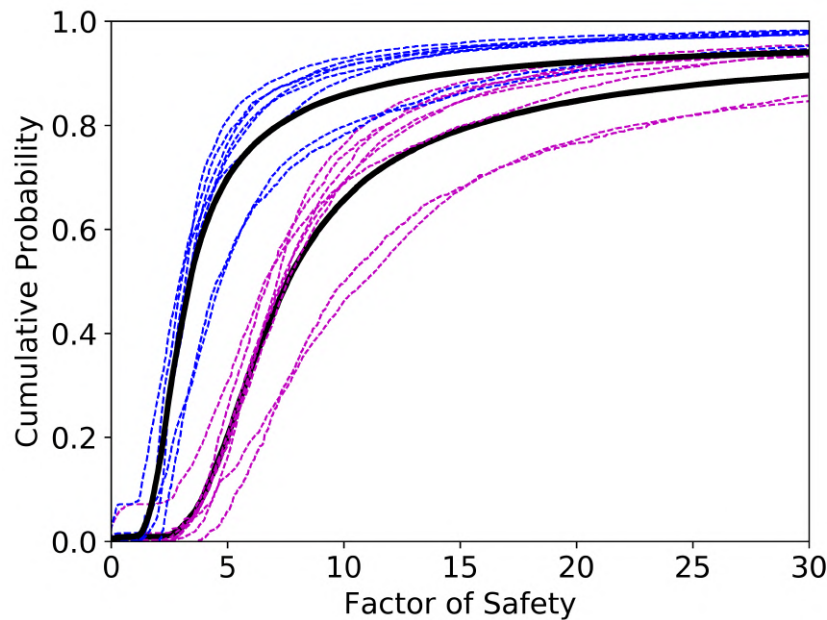
Figure 6-22.: DSS for the factor of safety for the information collected in
2017.

Information from independent sources is included, along with the updated DSS, where the evidence is combined. Left bound
in blue, lower bound in magenta and updates DSS in black

- When combining several sources of information by Yager's rule, the
  model is giving more importance to the consistent information. This
  evidence is expected to be the closest to the mean value and measured
  by different sources. Nevertheless, this approach does not disregard the
  possible extreme values that might affect the rock mass response, which
  is done by assigning the conflict to the universal set.

- The approach presented in this chapter is an alternative to systemati-
  cally include new information in the design process as it is available.
  This procedure allows us to update the model predictions and to assist
  the decision making at different project stages. Nevertheless, sound en-
  gineering judgment is required to define the input DSS and the proper
  models to be utilized.

- The proposed approach for combining information successfully subsu-
  med information collected at different times. Besides, since this method

considers the orientation as a Kent distributed variable, it reduces the uncertainty of the FoS DSS, compared to the one resulting from assuming dip and dip direction as independent variables.

- Besides, this algorithm combined highly conflicting sets of orientation data collected in 2017 by ShapeMetriX3D, into a single DSS that represents the stability of rock wedges, considering the aleatory and epistemic uncertainty inherent to the rock mass and the mapping techniques.

7. Concluding remarks

Fulfilling the proposed objectives, this research developed a framework to deal with uncertainty in rock slope problems, in which the presence of joints discontinuities controls the stability. This approach overcomes the limitations of deterministic and probabilistic techniques to deal with uncertainty, since it handles both epistemic and aleatory uncertainty through the Evidence Theory. In fact, the conventional probabilistic analysis is a particular case of the proposed framework, in which the epistemic uncertainty is not considered. Accordingly, the main contribution of this research are:

1. A methodology for combining robust datasets on joints geometry with limited information on geomechanical parameters within a rock wedge stability model. The variability of planes orientation is modeled as a Kent distributed random variable, while strength parameters are expressed as DSS. As a result, a DSS of the factor of safety is computed. This DSS reflects both the epistemic and the aleatory uncertainty of the input parameters.

2. A novel approach for updating sets of directional data that follow a Kent distribution. This approach resorts to the mixture models and the EM algorithm for clustering and defining weighting coefficients. These factors were treated as probability assignments for the updated evidence on orientation. The algorithm allowed to compute an updated FoS DSS, with less uncertainty than the DSSs obtained from the evidence considered separately.

The framework allows us to consider systematically the whole available information to compute a DSS of the factor of safety, without disregarding information or missing likely scenarios. This DSS provides a broader frame of

discernment since it presents an honest overview of the problem uncertainty, adjusted by expert sound engineering judgment. Besides, it explicitly shows the effect of including new information by updating the resulting DSS on the uncertainty of the model output.

The developed approach is supported by the following algorithms, that are contributions of this project as well:

1. A procedure to perform Monte Carlo simulation within a rock wedge stability analysis, modeling the planes' orientations as Kent distributed variables.

2. A modification of the Low's model to compute the horizontal wedge angles accounting for any mode of wedge failure.

3. An alternative for combining DFN with strength parameters expressed as DSS.

Further work

As the project was developed, the following opportunities for further research were identified:

1. Explore alternatives to adjust the parameters extracted from the remote sensing tool utilized in this project to improve its potential to mimic the actual rock mass structure by DFN.

2. Assess the suitability of algorithms other than the moments' method to estimate the Kent distribution parameters and perform the maximization step in the EM algorithm, within the stability analysis of rock wedges.

3. Study the spatial variability of rock mass structures, considering additional information extracted from the three-dimensional images by image processing techniques.

4. Explore alternatives for communicating to the stakeholders, the results as intervals of probability

Bibliografía

3GSM GmbH (2011). ShapeMetrix3D. Measurement and assessment of rock and terrain surfaces by metric 3D images. Technical report, Graz, Austria.

Agencia Nacional de Mineria (2019). Estadisticas de accidentalidad 2005-2019. Technical report, Agencia Nacional de Mineria, Bogota.

Ahmadabadi, M. and Poisel, R. (2016). Probabilistic Analysis of Rock Slopes Involving Correlated Non-normal Variables Using Point Estimate Methods. *Rock Mechanics and Rock Engineering*, 49(3):909–925.

Ahmed, R., Edwards, M., Lamite, S., and Mayur, P. (2014). Control-volume distributed multi-point flux approximation coupled with a lower-dimensional fracture model. *Journal of Computational Physics*.

Al Machot, F., Mayr, H. C., and Ranasinghe, S. (2018). A Hybrid Reasoning Approach for Activity Recognition Based on Answer Set Programming and Dempster–Shafer Theory. pages 303–318. Springer, Cham.

Altieri, M. G., Dell'Orco, M., Marinelli, M., and Sinesi, S. (2017). Evidence (Dempster – Shafer) Theory-Based evaluation of different Transport Modes under Uncertainty.: Theoretical basis and first findings. *Transportation Research Procedia*, 27:508–515.

Andersson, J., Shapiro, A. M., and Bear, J. (1984). A Stochastic Model of a Fractured Rock Conditioned by Measured Information. *Water Resources Research*, 20(1):79–88.

Arango Velez, I. F. (2014). *Desprendimiento de rocas en laderas: una guía para la evaluación del riesgo en vías*. PhD thesis, EAFIT.

Aven, T. (2010). On the need for restricting the probabilistic analysis in risk assessments to variability. *Risk analysis : an official publication of the Society for Risk Analysis*, 30(3):354–60; author reply 381–4.

Baecher, G. B. (1983). Statistical analysis of rock mass fracturing. *Journal of the International Association for Mathematical Geology*, 15(2):329–348.

Baecher, G. B. and Christian, J. T. (2003). *Reliability and Statistics in Geotechnical Engineering*. John Wiley & Sons, Chichester, England.

Baecher, G. B., Lanney, N. A., and Einstein, H. H. (1977). Statistical description of rock properties and sampling. In *The 18th US Symposium on Rock Mechanics (USRMS)*, volume 1, Golden, Colorado. American Rock Mechanics Association.

Baghbanan, A. and Jing, L. (2007). Hydraulic properties of fractured rock masses with correlated fracture length and aperture. *International Journal of Rock Mechanics and Mining Sciences*, 44(5):704–719.

Baghbanan, A. and Jing, L. (2008). Stress effects on permeability in a fractured rock mass with correlated fracture length and aperture. *International Journal of Rock Mechanics and Mining Sciences*, 45(8):1320–1334.

Balberg, I. and Binenbaum, N. (1983). Computer study of the percolation threshold in a two-dimensional anisotropic system of conducting sticks. *Physical Review B*, 28(7):3799–3812.

Ballent, W., Corotis, R. B., and Torres-Machi, C. (2019a). Dempster–Shafer Theory applications in post-seismic structural damage and social vulnerability assessment. *Sustainable and Resilient Infrastructure*, pages 1–13.

Ballent, W., Corotis, R. B., and Torres-Machi, C. (2019b). Representing uncertainty in natural hazard risk assessment with Dempster Shafer (Evidence) theory. *Sustainable and Resilient Infrastructure*, pages 1–15.

Banfield, J. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3):803–821.

Bangert, M., Hennig, P., and Oelfke, U. (2010). Using an infinite von Mises-Fisher Mixture Model to Cluster Treatment Beam Directions in External Radiation Therapy. In *Ninth International Conference on Machine Learning and Applications*.

Belayneh, M. W., Matthai, S. K., Blunt, M. J., and Rogers, S. F. (2009). Comparison of deterministic with stochastic fracture models in water-flooding numerical simulations. *AAPG Bulletin*, 93(11):1633–1648.

Ben-Haim, Y. (1994). A non-probabilistic concept of reliability. *Structural Safety*, 14(4):227–245.

Berkowitz, B. and Adler, P. M. (1998). Stereological analysis of fracture network structure in geological formations. *Journal of Geophysical Research: Solid Earth*, 103(B7):15339–15360.

Bernardini, A. and Tonon, F. (2010). *Bounding Uncertainty in Civil Engineering - Theoretical Background*. Springer Science & Business Media.

Berrone, S., Canuto, C., Pieraccini, S., and Scialò, S. (2018). Uncertainty Quantification in Discrete Fracture Network Models: Stochastic Geometry. *Water Resources Research*, 54(2):1338–1352.

Beynon, M., Curry, B., and Morgan, P. (2000a). The Dempster–Shafer theory of evidence: an alternative approach to multicriteria decision modelling. *Omega*, 28(1):37–50.

Beynon, M., Curry, B., and Morgan, P. (2000b). The Dempster–Shafer theory of evidence: an alternative approach to multicriteria decision modelling. *Omega*, 28(1):37–50.

Bhreasail, Á. N., Pritchard, O., Carluccio, S., Manning, J., Daly, T., Merritt, A., and Codd, J. (2018). Remote Sensing for Proactive Geotechnical Asset Management on England's Strategic Road Network. *Infrastructure Asset Management*, pages 1–40.

Billaux, D., Chiles, J., Hestir, K., and Long, J. (1989). Three-dimensional statistical modelling of a fractured rock mass—an example from the Fanay-Augères mine. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 26(3-4):281–299.

Birch, J. (2006). Using 3DM analyst mine mapping suite for rock face characterization. In Tonon, F. and Kottenstette, J., editors, *Laser and Photogrammetric Methods for Rock Face Characterization. Proc. 41 st U.S. Rock Mechanics Symp*, Golden, USA.

Bonilla-Sierra, V., Scholtès, L., Donzé, F. V., and Elmouttie, M. K. (2015). Rock slope stability analysis using photogrammetric data and DFN–DEM modelling. *Acta Geotechnica*, 10(4):497–511.

Boomsma, W., Kent, J. T., Mardia, K. V., Taylor, C. C., and Hamelryck, T. (2006). Graphical models and directional statistics capture protein structure. *Interdisciplinary models and Satistics and Bioinformatics*, 25:91–94.

Booth, P. and Meyer, G. (2013). Quarry wall stability and design optimisation using photogrammetric mapping and analysis techniques. In *2013 International Symposium on Slope Stability in Open Pit Mining and Civil Engineering*, pages 935–948, Brisbane. Australian Centre for Geomechanics.

Bour, O. and Davy, P. (1997). Connectivity of random fault networks following a power law fault length distribution. *Water Resources Research*, 33(7):1567–1583.

Bour, O. and Davy, P. (1998). On the connectivity of three-dimensional fault networks. *Water Resources Research*, 34(10):2611–2622.

Campbell, J. B. and Wynne, R. H. (2011). *Introduction to remote sensing*. Guilford Press.

Casagrande, A. (1965). Role of the Calculated Risk in Earthwork and Foundation Engineering. *Journal of the Soil Mechanics and Foundations Division*, 91(4):1–40.

Chen, J., Li, K., Chang, K.-J., Sofia, G., and Tarolli, P. (2015). Open-pit mining geomorphic feature characterisation. *International Journal of Applied Earth Observation and Geoinformation*, 42:76–86.

Chen, X.-Y., Fan, J.-P., and Bian, X.-Y. (2017). Theoretical analysis of non-probabilistic reliability based on interval model. *Acta Mechanica Solida Sinica*, 30(6):638–646.

Cigna, F., Bianchini, S., and Casagli, N. (2013). How to assess landslide activity and intensity with Persistent Scatterer Interferometry (PSI): the PSI-based matrix approach. *Landslides*, 10(3):267–283.

Couso, I., Dubois, D., and Sánchez, L. (2014). *Random Sets as Ill-Perceived Random Variables*. Springer International Publishing.

de Dreuzy, J.-R., Davy, P., and Bour, O. (2001a). Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 1. Effective connectivity. *Water Resources Research*, 37(8):2065–2078.

de Dreuzy, J.-R., Davy, P., and Bour, O. (2001b). Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 2. Permeability of networks based on lognormal distribution of apertures. *Water Resources Research*, 37(8):2079–2095.

Dempster, A. P. (1967). Upper and Lower Probabilities Induced by a Multi-valued Mapping. *The Annals of Mathematical Statistics*, 38(2):325–339.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data Via the <i>EM</i> Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE transactions on systems, man, and cybernetics*, 25(5):804–813.

Denœux, T. (2008). A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory. In *Classic Works of the Dempster-Shafer Theory*

of Belief Functions, pages 737–760. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dershowitz, W. S. and Einstein, H. H. (1988). Characterizing rock joint geometry with joint system models. *Rock Mechanics and Rock Engineering*, 21(1):21–51.

Dershowitz, W. S. and Fidelibus, C. (1999). Derivation of equivalent pipe network analogues for three-dimensional discrete fracture networks by the boundary element method. *Water Resources Research*, 35(9):2685–2691.

Ding, Y., Yao, X., Wang, S., and Zhao, X. (2019). Structural damage assessment using improved Dempster-Shafer data fusion algorithm. *Earthquake Engineering and Engineering Vibration*, 18(2):395–408.

Dong, L., Sun, D., Li, X., and Zhou, Z. (2017). Interval Non-Probabilistic Reliability of a Surrounding Jointed Rockmass in Underground Engineering: A Case Study. *IEEE Access*, 5:18804–18817.

DPN, D. N. d. P. (2013). Proyectos Viales bajo el esquema de Asociasiones Publico - Privadas: Cuarta Generacion de Concesiones Viales. Technical report, Bogota, Colombia.

Dubois, D. and Prade, H. (1992). On the Combination of Evidence in Various Mathematical Frameworks. pages 213–241. Springer, Dordrecht.

Dutta, P. (2018). An uncertainty measure and fusion rule for conflict evidences of big data via Dempster–Shafer theory. *International Journal of Image and Data Fusion*, 9(2):152–169.

Ebigbo, A., Lang, P. S., Paluszny, A., and Zimmerman, R. W. (2016). Inclusion-Based Effective Medium Models for the Permeability of a 3D Fractured Rock Mass. *Transport in Porous Media*, 113(1):137–158.

Einstein, H., Baecher, G., and Veneziano, D. (1978). Risk Analysisof Rock Slopes in Open Pit Mines. Technical report, U.S. Bureau of Mines.

Einstein, H. H. and Baecher, G. B. (1982). Probabilistic and Statistical Methods in Engineering Geology I. Problem Statement and Introduction to Solution. In *Ingenieurgeologie und Geomechanik als Grundlagen des Felsbaues / Engineering Geology and Geomechanics as Fundamentals of Rock Engineering*, pages 47–61. Springer Vienna, Vienna.

Elmo, D., Rogers, S., Stead, D., and Eberhardt, E. (2014). Discrete Fracture Network approach to characterise rock mass fragmentation and implications for geomechanical upscaling. *Mining Technology*, 123(3):149–161.

Elmouttie, M., Krähenbühl, G., and Poropat, G. (2013). Robust algorithms for polyhedral modelling of fractured rock mass structure. *Computers and Geotechnics*, 53:83–94.

Elmouttie, M., Poropat, G., and Krähenbühl, G. (2010). Polyhedral modelling of rock mass structure. *International Journal of Rock Mechanics and Mining Sciences*, 47(4):544–552.

Endo, H. (1984). *Mechanical transport in two-dimensional networks of fractures*. PhD thesis, University of California, Berkley.

EPA, U. E. P. A. (2002). Total Risk Integrated Methodology (TRIM) - TRIM.FaTE. Technical report, EPA-453/R-02-011a September 2002 TRIM Total Risk Integrated Methodology TRIM.FaTE Technical Support Document Volume I: Description of Module U.S. Environmental Protection Agency, Research Triangle Park, Nort Carolina, US.

Fadakar, Y. (2014). *Stochastic Modelling of Fractures in Rock Masses*. PhD thesis, The University of Adelaide.

Faille, I., Fumagalli, A., Jaffré, J., and Roberts, J. E. (2016). Model reduction and discretization using hybrid finite volumes for flow in porous media containing faults. *Computational Geosciences*, 20(2):317–339.

Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D. S., and Sentz, K. (2002). *Constructing probability boxes and Dempster-Shafer structures*, volume 835. Sandia National Laboratories.

Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., and Ginzburg, L. (2007). Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty.

Fisher, R. (1953). Dispersion on a Sphere. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 217(1130).

Francioni, M., Salvini, R., Stead, D., and Litrico, S. (2014). A case study integrating remote sensing and distinct element analysis to quarry slope stability assessment in the Monte Altissimo area, Italy. *Eng. Geology*, 183:290–302.

Franke, J., Redenbach, C., and Zhang, N. (2016). On a Mixture Model for Directional Data on the Sphere. *Scandinavian Journal of Statistics*, 43(1):139–155.

Freudenthal, A. M. (1956). Safety and the probability of structural failure. Technical report.

Gaich, A., Fasching, A., and Schubert, W. (2003). Improved site investigation Acquisition of geotechnical rock mass parameters based on 3D computer vision. In *Numerical Simulation in Tunnelling*, pages 13–46. Springer Vienna, Vienna.

Gaich, A., Pötsch, M., Fasching, A., and Schubert, W. (2004). Contact-free measurement of rock mass structures using the JointMetriX3D system. *International Journal of Rock Mechanics and Mining Sciences*, 41:304–309.

Ganeiber, A. M. (2012). *Estimation and simulation in directional and statistical shape models.* University of Leeds.

Geologia y Geotecnia (2011). Informe GYG-INF-095. Informe de Actualización de Modelo Geológico-Geotécnico y Ajuste del Diseño Minero de Holcim S.A. Technical report, Holcim Colombia S.A., Bogotá, Colombia.

Gheibie, S., Duzgun, S., and Akgun, A. (2013). Probabilistic-Numerical Modeling of Stability of a Rock Slope in Amasya-Turkey. In *47th U.S. Rock*

Mechanics/Geomechanics Symposium, volume 1, pages 341–346. American Rock Mechanics Association.

Glynn, E. F. (1979). A probabilistic approach to the stability of rock slopes.

GmbH, G. (2010). ShapeMetrix3D Manual. Technical report, Graz, Austria.

Goodman, R. E. and Shi, G. (1985). *Block Theory and its Application to Rock Engineering*. Prentice Hall, New Jersey, 1 edition.

Goodman, R. E. and Taylor, R. L. (1966). Methods Of Analysis For Rock Slopes And Abutments: A Review Of Recent Developments.

Hagan, T. (1980). A case for terrestrial photogrammetry in deep-mine rock structure studies. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 17(4):191–198.

Hamelryck, T. (2009). Probabilistic models and machine learning in structural bioinformatics. *Statistical Methods in Medical Research*, 18(5):505–526.

Hamelryck, T., Kent, J. T., and Krogh, A. (2006). Sampling Realistic Protein Conformations Using Local Structural Bias. *PLoS Computational Biology*, 2(9):e131.

Hammah, R. (2009). Numerical modelling of slope uncertainty due to rock mass jointing. In *Proceedings of the international conference on rock joints and jointed rock masses*.

Haztor, Y. (1992). *Validation of block theory using field case histories*. PhD thesis, University of Berkeley.

Hoek, E., Bray, J. W., and Boyd, J. M. (1973). The stability of a rock slope containing a wedge resting on two intersecting discontinuities. *Quarterly Journal of Engineering Geology and Hydrogeology*, 6(1):1–55.

Huadong Wu, Siegel, M., and Ablay, S. (2007). Sensor fusion using Dempster-Shafer theory II: static weighting and Kalman filter-like dynamic weighting. In *Proceedings of the 20th IEEE Instrumentation Technology Conference (Cat. No.03CH37412)*, volume 2, pages 907–912. IEEE.

Huadong Wu, Siegel, M., Stiefelhagen, R., and Jie Yang (2003). Sensor fusion using Dempster-Shafer theory [for context-aware HCI]. In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*, pages 7–12.

Jimenez-Rodriguez, R. and Sitar, N. (2006). A spectral method for clustering of rock discontinuity sets. *International Journal of Rock Mechanics and Mining Sciences*, 43(7):1052–1061.

Jimenez-Rodriguez, R., Sitar, N., and Chacón, J. (2006). System reliability approach to rock slope stability. *International Journal of Rock Mechanics and Mining Sciences*, 43(6):847–859.

Jing, L. (2003). A review of techniques, advances and outstanding issues in numerical modelling for rock mechanics and rock engineering. *International Journal of Rock Mechanics and Mining Sciences*, 40(3):283–353.

Jing, L. and Stephansson, O. (1994). Topological identification of block assemblages for jointed rock masses. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 31(2):163–172.

Jing, L. and Stephansson, O. (2007). Case Studies of Discrete Element Method Applications in Geology, Geophysics and Rock Engineering. In Engineering, L. J. and in Geotechnical, O. S. B. T. D., editors, *Fundamentals of Discrete Element Methods for Rock Engineering Theory and Applications*, volume Volume 85, pages 447–538. Elsevier.

Johnson, J. D., Helton, J. C., Oberkampf, W. L., Sallaberry, C. J., Johnson, J. D., Oberkampf, W. L., Sallaberry, C. J., Helton, J. C., Johnson, J. D., Oberkampf, W. L., and Sallaberry, C. J. (2008). Representation of analysis results involving aleatory and epistemic uncertainty. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA.

Kari, T., Gao, W., Zhao, D., Zhang, Z., Mo, W., Wang, Y., and Luan, L. (2018). An integrated method of ANFIS and Dempster-Shafer theory for

fault diagnosis of power transformer. *IEEE Transactions on Dielectrics and Electrical Insulation*, 25(1):360–371.

Karimi-Fard, M., Durlofsky, L., and Aziz, K. (2004). An Efficient Discrete-Fracture Model Applicable for General-Purpose Reservoir Simulators. *SPE Journal*, 9(02):227–236.

Kasarapu, P. (2015). Modelling of directional data using Kent distributions. *arXiv preprint arXiv:1506.08105*.

Kasarapu, P. and Allison, L. (2015). Minimum message length estimation of mixtures of multivariate Gaussian and von Mises-Fisher distributions. *Machine Learning*, 100(2-3):333–378.

Kent, J. T. (1980). The Fisher-Bingham Distribution on the Sphere. Technical report, Department of Statistics, Princeton University, New Jersey.

Kent, J. T. (1982). The Fisher-Bingham Distribution on the Sphere.

Kent, J. T. (2012). Statistical Modelling and Simulation Using the Fisher-Bingham Distribution. pages 179–188. Springer, Berlin, Heidelberg.

Kent, J. T., Constable, P. D., and Er, F. (2004). Simulation for the complex Bingham distribution. *Statistics and Computing*, 14(1):53–57.

Kent, J. T., Ganeiber, A. M., and Mardia, K. V. (2017). A new unified approach for the simulation of a wide class of directional distributions. *Journal of Computational and Graphical Statistics*, pages 0–0.

Kent, J. T. and Hamelryck, T. (2005). Using the Fisher-Bingham distribution in stochastic models for protein structure. *Quantitative Biology, Shape Analysis, and Wavelets*, 24(1):57–60.

Khairina, D. M., Hatta, H. R., Rustam, R., and Maharani, S. (2018). Automation Diagnosis of Skin Disease in Humans using Dempster-Shafer Method. *E3S Web of Conferences*, 31:11006.

Khan, M. S. (2010). *Investigation of Discontinuous Deformation Analysis for Application in Jointed Rock Masses*. PhD thesis, University of Toronto.

Kim, D. H., Gratchev, I., and Balasubramaniam, A. (2015). Back analysis of a natural jointed rock slope based on the photogrammetry method. *Landslides*, 12(1):147–154.

Kim, J. K., Choi, M. J., Lee, J. S., Hong, J. H., Kim, C.-S., Seo, S. I., Jeong, C. W., Byun, S.-S., Koo, K. C., Chung, B. H., Park, Y. H., Lee, J. Y., and Choi, I. Y. (2018). A Deep Belief Network and Dempster-Shafer-Based Multiclassifier for the Pathology Stage of Prostate Cancer. *Journal of Healthcare Engineering*, 2018:1–8.

Klapperich, H., Rafig, A., and Wu, W. (2012). Non-Deterministic Analysis of Slope Stability based on Numerical Simulation.

Lai, X.-P., Shan, P.-F., Cai, M.-F., Ren, F.-H., and Tan, W.-H. (2015). Comprehensive evaluation of high-steep slope stability and optimal high-steep slope design by 3D physical modeling. *International Journal of Minerals, Metallurgy, and Materials*, 22(1):1–11.

Lei, Q., Latham, J.-P., and Tsang, C.-F. (2017). The use of discrete fracture networks for modelling coupled geomechanical and hydrological behaviour of fractured rocks. *Computers and Geotechnics*, 85:151–176.

Lei, Q., Latham, J.-P., Xiang, J., Tsang, C.-F., Lang, P., and Guo, L. (2014). Effects of geomechanical changes on the validity of a discrete fracture network representation of a realistic two-dimensional fractured rock. *International Journal of Rock Mechanics and Mining Sciences*, 70:507–523.

Lin, D., Fairhurst, C., and Starfield, A. (1987). Geometrical identification of three-dimensional rock block systems using topological techniques. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 24(3):331–338.

Lin, J., Tao, H., Wang, Y., and Huang, Z. (2010). Practical application of unmanned aerial vehicles for mountain hazards survey. In *IEEE 18 th Geoinformatics Int. Conf.*

Liu, F., Zhao, Q., and Yang, Y. (2018). An approach to assess the value of industrial heritage based on Dempster–Shafer theory. *Journal of Cultural Heritage.*

Long, J. C. S. and Billaux, D. M. (1987). From field data to fracture network modeling: An example incorporating spatial structure. *Water Resources Research*, 23(7):1201–1216.

Long, J. C. S., Remer, J. S., Wilson, C. R., and Witherspoon, P. A. (1982). Porous media equivalents for networks of discontinuous fractures. *Water Resources Research*, 18(3):645–658.

Low, B. (2007). Reliability analysis of rock slopes involving correlated non-normals. *International Journal of Rock Mechanics and Mining Sciences*, 44(6):922–935.

Low, B. and Einstein, H. (1991). Simplified Reliability Analysis for Wedge Mechanisms in Rodk Slopes. In *Sixth International Symposium on Landslides*, pages 199–507, Christchurch, New Zealnd. A. A. Balkema.

Low, B. K. (1979). *Reliability of rock slopes with wedge mechanisms.* PhD thesis, Massachusetts Institute of Technology.

Low, B. K. (1997). Reliability Analysis of Rock Wedges. *Journal of Geotechnical and Geoenvironmental Engineering*, 123(6):498–505.

Low, B. K. and Einstein, H. H. (1992). Simplified reliability analysis for wedge mechanisms in rock slopes. In *Proc., 6th Int. Symp. on Landslides*, pages 499–507. Balkema, Rotterdam The Netherlands.

Lucieer, A., de Jong, S. M., and Turner, D. (2014). Mapping landslide displacements using Structure from Motion (SfM) and image correlation of multitemporal UAV photography. *Progress in Physical Geography*, 38(1):97–116.

Lunga, D. and Ersoy, O. (2011). Kent Mixture Model for Hyperspectral Clustering via Cosine Pixel Coordinates on Spherical Manifolds. Technical report, Purdue University, West Lafayette, IN.

Ma, G. and Fu, G. (2014). A rational and realistic rock mass modelling strategy for the stability analysis of blocky rock mass. *Geomechanics and Geoengineering*, 9(2):113–123.

Ma, G. W. and Fu, G. Y. (2013). Stochastic key block analysis of underground excavations. In *Frontiers of Discontinuous Numerical Methods and Practical Simulations in Engineering and Disaster Prevention - Proceedings of the 11th Int. Conf. on Analysis of Discontinuous Deformation, ICADD 2013*, pages 51–60.

Mardia, K. V. (1975). *Statistics of Directional Data*, volume 37. Academic Press.

Marek, L., Miřijovský, J., and Tuček, P. (2015). Monitoring of the Shallow Landslide Using UAV Photogrammetry and Geodetic Measurements. In *Engineering Geology for Society and Territory - Volume 2*, pages 113–116. Springer International Publishing, Cham.

Martinez, J., Buill, F., and Bartoll, J. (2005). Utilización de técnicas láser escáner y de fotogrametría terrestre para el estudio de desprendimientos de rocas: el caso de la zona de más riesgo del tren cremallera. *Mapping*, 103:26–33.

McLachlan, G. J. and Peel, D. (1999). The EMMIX Algorithm for the Fitting of Normal and t -Components. *Journal of Statistical Software*, 4(2):1–14.

McMahon, B. (1971). Statistical methods for the design of rock slopes. In *First, Australian-New Zealand Conference on Geomechanics*, pages 314–321.

Min, K.-B. and Jing, L. (2004). Stress dependent mechanical properties and bounds of poisson's ratio for fractured rock masses investigated by a DFN-DEM technique. *International Journal of Rock Mechanics and Mining Sciences*, 41:390–395.

Mintransporte (2011). Transporte en cifras. Version 2011. Technical report, Ministerio del Transporte, Bogotá.

Moradi, M., Chaibakhsh, A., and Ramezani, A. (2018). An intelligent hybrid technique for fault detection and condition monitoring of a thermal power plant. *Applied Mathematical Modelling*, 60:34–47.

Mosaad Allam, M. (1978). The estimation of fractures and slope stability of rock faces using analytical photogrammetry. *Photogrammetria*, 34(3):89–99.

Nasekhian, A. and Schweiger, H. F. (2010). Random Set Finite Element Method Application to Tunnelling.

Nasekhian, A. and Schweiger, H. F. (2011). Random set finite element method application to tunnelling. *International Journal of Reliability and Safety*, 5(3/4):299.

Nguyen, H. T. (2006). *An Introduction to Random Sets*. Taylor & Francis Group, Las Crices, NM.

Norrish, N. and Willey, D. (1996). Rock Slope Stability Analysis. In Transportation Research Board, editor, *Lanslides: Investigation and Mitigation - Special Report 247*, chapter 15, pages 391–424. National Academy Press.

Oberguggenberger, M. (2012). Combined methods in nondeterministic mechanics. In Isaac Elishakoff, C. S., editor, *Nondeterministic Mechanics*, pages 263–356. Springer, Vienna.

Oberkampf, W. L., Helton, J. C., Joslyn, C. A., Wojtkiewicz, S. F., and Ferson, S. (2004a). Challenge problems: uncertainty in system response given uncertain parameters. *Reliability Engineering & System Safety*, 85(1):11–19.

Oberkampf, W. L., Tucker, W. T., Zhang, J., Ginzburg, L., Berleant, D. J., Ferson, S., Hajagos, J., and Nelsen, R. B. (2004b). Dependence in probabilistic modeling, Dempster-Shafer theory, and probability bounds analysis. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA.

Ortega, I., Booth, P., and Darras, J. (2013). Stability analysis and remedial design of two road cuttings in North Queensland based on remote geotechnical mapping using digital photogrammetry. In *Proc. Slope Stability*, Perth, Australia.

Pahl, P. (1981). Estimating the mean length of discontinuity traces. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 18(3):221–228.

Paine, P. J., Preston, S. P., Tsagris, M., and Wood, A. T. A. (2017). An elliptically symmetric angular Gaussian distribution. *Statistics and Computing*, pages 1–9.

Park, H. (1999). Risk analysis of rock slope stability and stochastic properties of discontinuity parameters in western North Carolina. *Theses and Dissertations Available from ProQuest*.

Park, H. (2000). Probabilistic Approach of Stability Analysis for Rock Wedge Failure. *Economic and Environmental Geology*, 33(4):295–307.

Park, H. and West, T. R. (2001). Development of a probabilistic approach for rock wedge failure. *Engineering Geology*, 59(3-4):233–251.

Park, H.-J., West, T. R., and Woo, I. (2005). Probabilistic analysis of rock slope stability and random properties of discontinuity parameters, Interstate Highway 40, Western North Carolina, USA. *Engineering Geology*, 79(3-4):230–250.

Park, J., Bates, M., Jeong, Y. S., Kim, K. M., and Kemeny, J. (2016). Creating a Digital Outcrop Model by Using Hyper-Spectrometry and Terrestrial LiDAR. In *50th U.S. Rock Mechanics/Geomechanics Symposium*, pages 26–29. American Rock Mechanics Association.

Peck, R. B. (1969). Advantages and Limitations of the Observational Method in Applied Soil Mechanics. *Géotechnique*, 19(2):171–187.

Peel, D. and Mclachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, 10:339–348.

Peel, D., Whiten, W. J., and McLachlan, G. J. (2001). Fitting Mixtures of Kent Distributions to Aid in Joint Set Identification. *Journal of the American Statistical Association*, 96(453):56–63.

Peschl, G. M. (2004). *Reliability Analyses in Geotechnics with Random Set Finite Element Method*. PhD thesis, Technische Universitat Graz.

Priest, S. D. S. D. (1993). *Discontinuity analysis for rock engineering*. Chapman & Hall.

Qiu, Z., Yang, D., and Elishakoff, I. (2008). Probabilistic interval reliability of structural systems. *International Journal of Solids and Structures*, 45(10):2850–2860.

Rathman, J. F., Yang, C., and Zhou, H. (2018). Dempster-Shafer theory for combining in silico evidence and estimating uncertainty in chemical risk assessment. *Computational Toxicology*.

Riquelme, A., Cano, M., Tomás, R., and Abellán, A. (2017). Identification of rock slope discontinuity sets from laser scanner and photogrammetric point clouds: A comparative analysis. In *Symposium of the International Society for Rock Mechanics, EUROCK 2017.*, pages 838–845, Ostrava-Poruba, Czech Republic.

Robinson, P. C. (1983). Connectivity of fracture systems-a percolation theory approach. *Journal of Physics A: Mathematical and General*, 16(3):605–614.

Ruan, Z., Li, C., Wu, A., and Wang, Y. (2019). A New Risk Assessment Model for Underground Mine Water Inrush Based on AHP and D–S Evidence Theory. *Mine Water and the Environment*.

Ruzic, I., Marovic, I., Benac, C., and Ilic, S. (2014). Coastal cliff geometry derived from structure-from-motion photogrammetry at Stara Baška, Krk Island, Croatia. *Geo-Marine Letters*, 34(6):555–565.

Salvini, R., Francioni, M., Riccucci, S., Bonciani, F., and Callegari, I. (2013). Photogrammetry and laser scanning for analyzing slope stability and rock

fall runout along the Domodossola–Iselle railway, the Italian Alps. *Geomorphology*, 185:110–122.

Sandve, T., Berre, I., Physics, J. N. J. o. C., and 2012, U. (2012). An efficient multi-point flux approximation method for discrete fracture–matrix simulations. *Journal of Computational Physics*, 231(9).

Schweiger, H. F. and Peschl, G. M. (2004). Numerical analysis of deep excavations utilizing random set theory. *Proc. Geotechnical Innovations, Essen: VGE*, pages 277–294.

Schweiger, H. F. and Peschl, G. M. (2005a). Application of the random set finite element method (RS-FEM) in geotechnics. *Plaxis Bulletin (17)*.

Schweiger, H. F. and Peschl, G. M. (2005b). Reliability analysis in geotechnics with the random set finite element method. *Computers and Geotechnics*, 32(6):422–435.

SCIRO (2015). Siromodel OPS. Technical report, Commonwealth Scientific and Research Organisation, Australia.

Seidel, W., Mosler, K., and Alker, M. (2000). A Cautionary Note on Likelihood Ratio Tests in Mixture Models. *Annals of the Institute of Statistical Mathematics*, 52(3):481–487.

Sentz, K. and Ferson, S. (2002). Combination of Evidence in Dempster-Shafer Theory. Technical Report April, Sandia National Laboratories, Albuquerque, NM.

Seraj, S., Delavar, M., and R, R. (2019). A hybrid GIS-assisted framework to integrate Dempster-Shafer theory of evidence and fuzzy sets in risk analysis: An application in hydrocarbon exploration. *Geocarto International*, pages 1–19.

SGC, S. G. C. (2015). Guía metodológica para estudios de amenaza, vulnetabilidad y riesgo por movimientos en masa. Technical report, SGC y Minminas, Bogotá, Colombia.

Shafer, G. (1976). *A Mathematical Thoery of Evidence.* Princeton Uiversity Press, New Jersey, NJ, US.

Shen, H. and Abbas, S. M. (2013). Rock slope reliability analysis based on distinct element method and random set theory. *International Journal of Rock Mechanics and Mining Sciences*, 61:15–22.

Shi, G. (1988). *Discontinuous deformation analysis: A new numerical model for the statics and dynamics of block systems.* PhD thesis, University of California, Berkeley.

Shi, G.-H., Goodman, R. E., and Tinucci, J. P. (1985). The Kinematics Of Block Inter-Penetrations. In *The 26th U.S. Symposium on Rock Mechanics (USRMS)*, volume 1, pages 121–130. American Rock Mechanics Association.

Sra, S. and Karp, D. (2013). The multivariate Watson distribution: Maximum-likelihood estimation and other aspects. *Journal of Multivariate Analysis*, 114:256–269.

Stumpf, A., Malet, J., Allemand, P., Pierrot, M., Deseilligny, and Skupinnski, G. (2015). Ground-based multi-view photogrammetry for the monitoring of landslide deformation and erosion. *Geomorphology*, 231:130–145.

Sturzenegger, M. and Stead, D. (2009). Close-range terrestrial digital photogrammetry and terrestrial laser scanning for discontinuity characterization on rock cuts. *Engineering Geology*, 106(3-4):163–182.

Talavera, A., Aguasca, R., Galván, B., and Cacereño, A. (2013). Application of Dempster–Shafer theory for the quantification and propagation of the uncertainty caused by the use of AIS data. *Reliability Engineering & System Safety*, 111:95–105.

Tannant, D. (2015). Review of Photogrammetry-Based Techniques for Characterization and Hazard Assessment of Rock Faces. *International journal of geohazards and environment*, 1(2):76–87.

Tonon, F. and Bernardini, A. (1999). Multiobjective Optimization of Uncertain Structures Through Fuzzy Set and Random Set Theory. *Computer-Aided Civil and Infrastructure Engineering*, 14(2):119–140.

Tonon, F., Bernardini, A., and Mammino, A. (2000a). Determination of parameters range in rock engineering by means of Random Set Theory. *Reliability Engineering & System Safety*, 70(3):241–261.

Tonon, F., Bernardini, A., and Mammino, A. (2000b). Reliability analysis of rock mass response by means of Random Set Theory. *Reliability Engineering & System Safety*, 70(3):263–282.

Tonon, F., Mammino, A., and Bernardini, A. (1996). A Random Set Approach to the Uncertainties In Rock Engineering And Tunnel Lining Design. In *ISRM International Symposium - EUROCK 96*, Turin, Italy. International Society for Rock Mechanics.

Torkzadeh-Mahani, N., Dehghani, M., Mirian, M. S., Shakery, A., and Taheri, K. (2018). Expert finding by the Dempster-Shafer theory for evidence combination. *Expert Systems*, 35(1):13.

Torrero, L., Seoli, L., Molino, A., Giordan, D., Manconi, A., Allasia, P., and Baldo, M. (2015). The Use of Micro-UAV to Monitor Active Landslide Scenarios. In *Engineering Geology for Society and Territory - Volume 5*, pages 701–704. Springer International Publishing, Cham.

Tucker, W. T. and Ferson, S. (2006). Sensitivity in risk analyses with uncertain numbers. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA.

UPME (2015). Sistema de informacion minero Colombiano.

Vasuki, Y., Holden, E., Kovesi, P., and Micklethwaite, S. (2014). Semi-automatic mapping of geological Structures using UAV-based photogrammetric data: An image analysis approach. *Computers & Geosciences*, 69:22–32.

Vollmer, F. W. (2015). Orient 3: a new integrated software program for orientation data analysis, kinematic analysis, spherical projections, and Schmidt plots. In *Geological Society of America, Abstracts with Programs*, volume 47, pages 185–187.

Wang, C., Zhang, H., and Beer, M. (2019). Tightening the bound estimate of structural reliability under imprecise probability information. In *13th International Conference on Applications of Statistics and Probability in Civil Engineering(ICASP13), Seoul, South Korea*, pages 1–8.

Wang, P., Puterman, M., Cockburn, I., and Le, N. (1996). Mixed Poisson regression models with covariate dependent rates. *Biometrics*, pages 381–400.

Wang, Y. (2010). Imprecise probabilities based on generalised intervals for system reliability assessment. *International Journal of Reliability and Safety*, 4(4):319.

Wang, Y.-m. and Jing, L. T. (2012). Application of Dempster-Shafer theory for network selection in heterogeneous wireless networks. *The Journal of China Universities of Posts and Telecommunications*, 19:86–91.

Warburton, P. (1980). Stereological interpretation of joint trace data: Influence of joint shape and implications for geological surveys. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 17(6):305–316.

Warburton, P. (1983). Application of a new computer model for reconstructing blocky block geometry analysis single block stability and identifying keystones. In *Proceedings of the 5th International Congress on Rock Mechanics*, pages 225–230, Melbopurne, Australia. Balkema.

Whitman, R. V. (1984). Evaluating Calculated Risk in Geotechnical Engineering. *Journal of Geotechnical Engineering*, 110(2):143–188.

Wickens, E. H. and Barton, N. (1971). The application of photogrammetry to the stability of excavated rock slopes. *The Photogrammetric Record*, 7(37):16–54.

Wood, A. T. A. (1987). The simulation of spherical distributions in the Fisher-Bingham family. *Communications in Statistics - Simulation and Computation*, 16(3):885–898.

Xu, C. and Dowd, P. (2010). A new computer code for discrete fracture network modelling. *Computers & Geosciences*, 36(3):292–301.

Yager, R. R. (1987). On the dempster-shafer framework and new combination rules. *Information Sciences*, 41(2):93–137.

Yamaji, A. (2016). Genetic algorithm for fitting a mixed Bingham distribution to 3D orientations: a tool for the statistical and paleostress analyses of fracture orientations. *Island Arc*, 25:72–83.

Yamaji, A. and Sato, K. (2011). Clustering of fracture orientations using a mixed Bingham distribution and its application to paleostress analysis from dike or vein orientations. *Journal of Structural Geology*, 33(7):1148–1157.

Yang, Q., Lu, P., Cui, T., Ma, M., Liu, Y., Zhou, C., and Zhao, L. (2012). Application of low-altitude remote sensing image by unmanned airship in geological hazards investigation. In *Proc. Image and Signal Processing 5 th Int. Cong*, Chongqing, China.

Zadeh, L. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2):85–90.

Zargar, A., Sadiq, R., Naser, G., Khan, F. I., and Neumann, N. N. (2012). Dempster-Shafer Theory for Handling Conflict in Hydrological Data: Case of Snow Water Equivalent. *Journal of Computing in Civil Engineering*, 26(3):434–447.

Zhang, Y., Xiao, M., and Chen, J. (2010). A new methodology for block identification and its application in a large scale underground cavern complex. *Tunnelling and Underground Space Technology*, 25(2):168–180.

Zhang, Y., Xiao, M., Ding, X., and Wu, A. (2012). Improvement of methodology for block identification using mesh gridding technique. *Tunnelling and Underground Space Technology*, 30:217–229.

Zhang, Z. X. and Lei, Q. H. (2013). Object-oriented modeling for three-dimensional multi-block systems. *Computers and Geotechnics*, 48:208–227.

Zhang, Z. X. and Lei, Q. H. (2014). A Morphological Visualization Method for Removability Analysis of Blocks in Discontinuous Rock Masses. *Rock Mechanics and Rock Engineering*, 47(4):1237–1254.

Zhao, Z., Rutqvist, J., Leung, C., Hokr, M., Liu, Q., Neretnieks, I., Hoch, A., Havlíček, J., Wang, Y., Wang, Z., Wu, Y., and Zimmerman, R. (2013). Impact of stress on solute transport in a fracture network: A comparison study. *Journal of Rock Mechanics and Geotechnical Engineering*, 5(2):110–123.

Zheng, Y., Xia, L., Yu, Q., Yinhe, Z., Lu, X., Qingchun, Y., Zheng, Y., Xia, L., Yu, Q., Yinhe, Z., Lu, X., and Qingchun, Y. (2015). A method for identifying three-dimensional rock blocks formed by curved fractures. *Computers and Geotechnics*, 65:1–11.

A. Short Range Photogrammetry for Geotechnical Characterization of Rock Masses

A.1. Introduction

The three primary sources of uncertainty in rock mechanics problems are the inherent spatial variability of properties, the error induced in measuring and estimating engineering properties and inaccuracies in modeling [Einstein and Baecher, 1982].

The first two sources can be handled by improving the data collection process. The collection of a higher number of observations will not reduce the number of computations but will improve the characterization of such variability. On the other hand, involving alternative and reproducible methods for collecting observations on the rock mass will contribute to reduce the uncertainty linked to the second source of uncertainty, the measurement error.

With this framework, in this project two alternatives were used to collect the information on joints geometry. First, the orientation of joint planes was measured by using a compass. Subsequently, a short-range photogrammetry system was used to gather information on rock joints structure.

The applied method integrates an extensive experimental plan in both, field and computer laboratory, to collect structural data such as orientation, persistence, and spacing of discontinuities in a rock mass taken as a case study. Simultaneously, an intensive work of data collection in the field was carried out by conventional methodologies using compass and tape, for comparative purposes and statistical validation. The results allow establishing the advantages of using unconventional techniques, in terms of quality and quantity

of information, as well as efficiency in terms of time spent on field data collection. Through the semi-automatic analysis of three-dimensional images, geological mapping and objective characterization of discontinuities for geotechnical purposes were achieved, constituting a powerful tool with great potential for use in road and mining projects in Colombia.

A.2. Conventional measurements

The two main conventional techniques for collecting rock mass structure are boreholes and compass measurements. Boreholes provide information on joints geometry in depth, beyond the exposed area. However, it is relatively expensive and time-consuming. Besides, given the small size of the sample, the information provided is very limited, and a proper interpretation of the information is required.

Conversely, the compass and tape for measuring information in outcrops and engineering rock slopes, provide an important amount of information at a much lower cost. However, by itself, it does not provide information on the evolution of geometry with depth, as boreholes. Hence a combination of both approaches is desirable.

In both cases, direct access to the observation area is required, which intermediately generates the following limitations:

1. Limited access

2. Information limited by operators height and ability to access some points

3. Time-consuming activity, which can be a drawback, especially in mining operations, where operation constrains access to slopes

4. Measurements depend on the operator ability to position and read the compass. e.g., Most measurements are taken up to the operator's height.

The abovementioned drawbacks have two main consequences: results are biased, and reproducibility is not guaranteed [Gaich et al., 2004]. Image proces-

sing systems are a relatively new approach to characterize rock mass joint geometry, overcoming referred limitations.

In practice, the selection of sampling sector is conditioned by the availability of accessible outcrops and engineered slopes and depends upon expert opinion based on its interpretation of the geological conditions of the rock mass and the subjectivity for the selection of sampling points.

In order to overcome and reduce the uncertainty linked to the joints geometry sampling process, new technologies based on remote measurements have been developed [Campbell and Wynne, 2011]. Bhreasail et al. [2018] developed a revision of the remote techniques available for geotechnical purposes, to be implemented in roads management in England, such as multispectral and hyperspectral images, Synthetic Apertura Radar Interferometry (InSAR) [Cigna et al., 2013], passive microwave radiometry and cameras with motion sensors. Besides, there are several examples of applications of the laser with LIDAR [Martinez et al., 2005], a combination of techniques, [Chen et al., 2015, Park et al., 2016, Salvini et al., 2013] , and comparative analysis [Riquelme et al., 2017].

As for photogrammetry applied to rock structure, measurements Wickens and Barton [1971] published a pioneering work to measure the roughness profile on rocks. Then, the Topographical Survey Directorate of the Surveys and Mapping Branch, Department of Energy, Mines and Resources in Canada, carried out a research to measure joints geometry and slope face movements by using photogrammetry [Mosaad Allam, 1978]. Subsequently, Hagan [1980] utilized pictures taken from a setup of two cameras to compute the position of five points along a single discontinuity. With this information, the orientation of the plane is computed.

Several works have focused on the measurement of geometrical features of rock slopes by photogrammetry [Birch, 2006, Ruzic et al., 2014, Sturzenegger and Stead, 2009, Vasuki et al., 2014], for hazard assessment [Lin et al., 2010, Yang et al., 2012] for landslide mapping Lucieer et al. [2014], and monitoring [Marek et al., 2015, Stumpf et al., 2015, Torrero et al., 2015].

Information collected using photogrammetry has been used as input information for further geotechnical stability assessment [Bonilla-Sierra et al., 2015, Booth and Meyer, 2013, Francioni et al., 2014, Kim et al., 2015, Lai et al.,

2015, Ortega et al., 2013]

This list does not pretend to be exhaustive, but it shows a broad variety of applications of photogrammetry in rock mechanics and geology-related activities. The most relevant advantages of implementing this sort of techniques in engineering applications, compared to mapping with compass or lidar instruments are [Tannant, 2015]:

1. Quick and cost-effective (camera less expensive than lidar instruments)

2. Hazardous areas can be safely measured

3. Fieldwork has no impact on construction or mining activities

4. The technique provides a detailed images form the rock surface

For researchers and practitioners in the rock mechanics field, the last advantage is relevant, since images can have a permanent and close view of the rock mass they are dealing with. This information does help to get acquainted with the rock mass features and conditions. This characteristic can not be quantified, but it can improve the quality of the design and will assist the designer during the decision making process.

As for Colombia, Arango Velez [2014] used image form radar, interferometry, and satellite in rock mechanics applications. However, there are no references to the use of photogrammetry in rock mechanics applications.

Considering the need for developing more reliable models and introducing up-to-date technologies to daily engineering activity. This project resorts to ShapeMetrix3D that is a short-range photogrammetry system that builds a 3D image of the rock face based on two overlapping images, using the principle of stereoscopy. The system is based on the original work of Gaich et al. [2003] and then commercialized by the Austrian company 3GSM..

A.3.   Case study

The Pedregal mine is a sandstone quarry situated in Une, Cundinamarca, Colombia, as shown in Figure **A-1** . This mine is in a sedimentary rock mass,

formed mainly by alternating layers of sandstone and shales. The mine has been operated since mid 1990ies until today. Holcim Colombia S.A. operated the mine until 2013, then Gravillera Albania S.A. took over control of the operation. Both companies have had a responsible follow-up program on geological and geotechnical aspects, according to Colombian legislation.
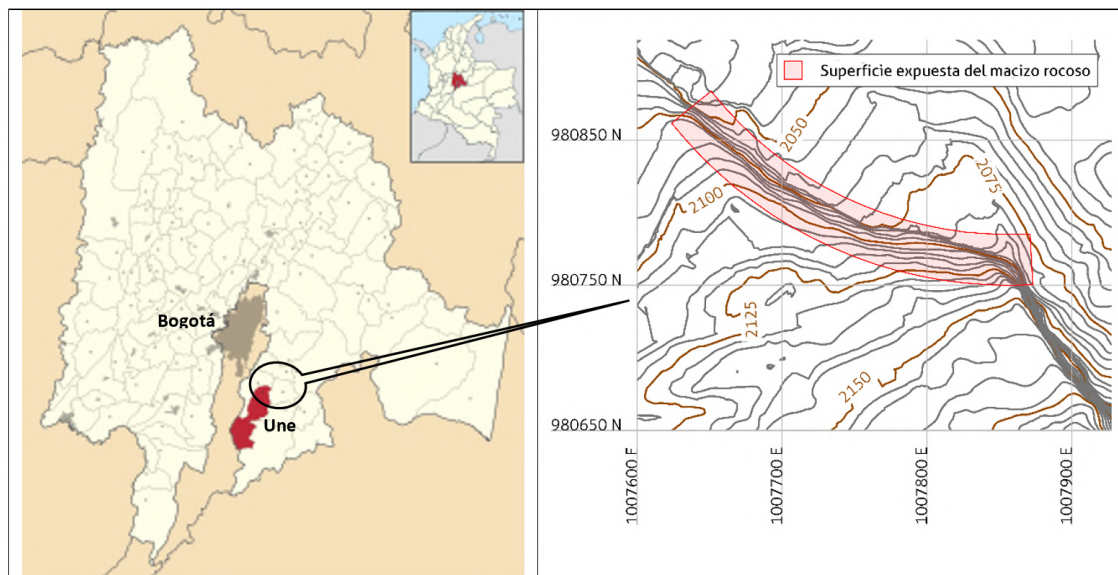
Figure A-1.: Location of The Pedregal Mine

As for the geology, the mine is located on a sedimentary complex from Arenisca de Une formation. The rock mass is on monoclinal dipping northwest with variable dip ranging between 18°and 30 °. A 1500m length scarp bounds the east side of the monoclinal. On the north edge of the scarp is located the mine exploration. The main exploitation slope has approximately 270m length and 25m high.

Regarding the stratigraphy, sandstone layers are identified with inter-bedded layers of shale. These are followed by inter-bedded strata of quartzitic sandstone of variable grain size (layers F, G1, and G2), carbonaceous shales (F1m1, G1m1, and G2m2). On the lower part of the sequence, there are strata of medium to fine-grained gray sandstone (H1 and H2), with thin layers of carbonaceous shales (H1m1 and H2m2). Figure **A-2** shows a detail view of the lithologies identified in the main mine slope.

The sequence is summarized in Table **A-1**. Besides, in Figure **A-3** a general plan view of the surface geological model is depicted, along with a typical
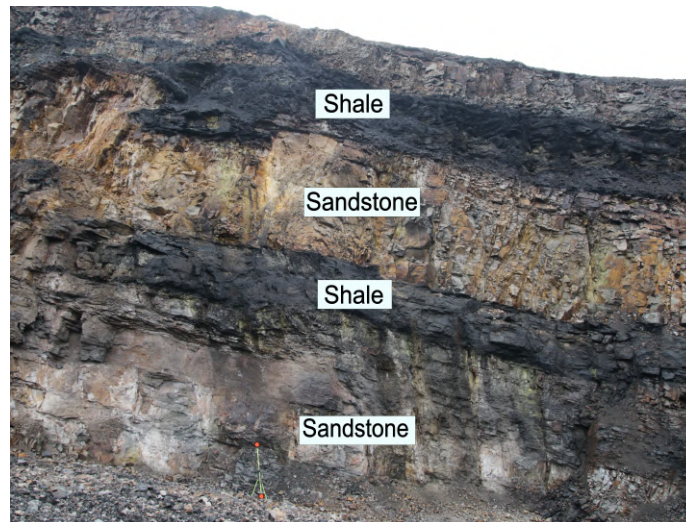
Figure A-2.: Detailed view of the sandstone shale sequence in the main slope

cross-section of the main rock slope is depicted in Figure **A-4**. Gravillera Albania S.A. provided the information on the mine geology included in Geologia y Geotecnia [2011].

A.3.1. Structural geology

Regarding the structural geology, the mine is located in a tectonic compressive stresses area, resulting from the interaction between the continental South American and oceanic Nazca plates. This interaction has generated several events of rock crust rupture that are evident by the presence of faults, which control the fracturing pattern of the mine.

There are two NE-SW strike faults, as can be seen in Figure **A-3**. The north fault has a predominant orientation N-60-E and is dipping between 60° and 65°. The south fault presents and orientation N-58-E and a dip that varies from 80° and 85° [Geologia y Geotecnia, 2011]. Aside from these faults, the analysis of the structural information collected in the mine has allowed identifying three main joint sets, on the main slope. A summary of these structural features is presented in Table **A-2**.
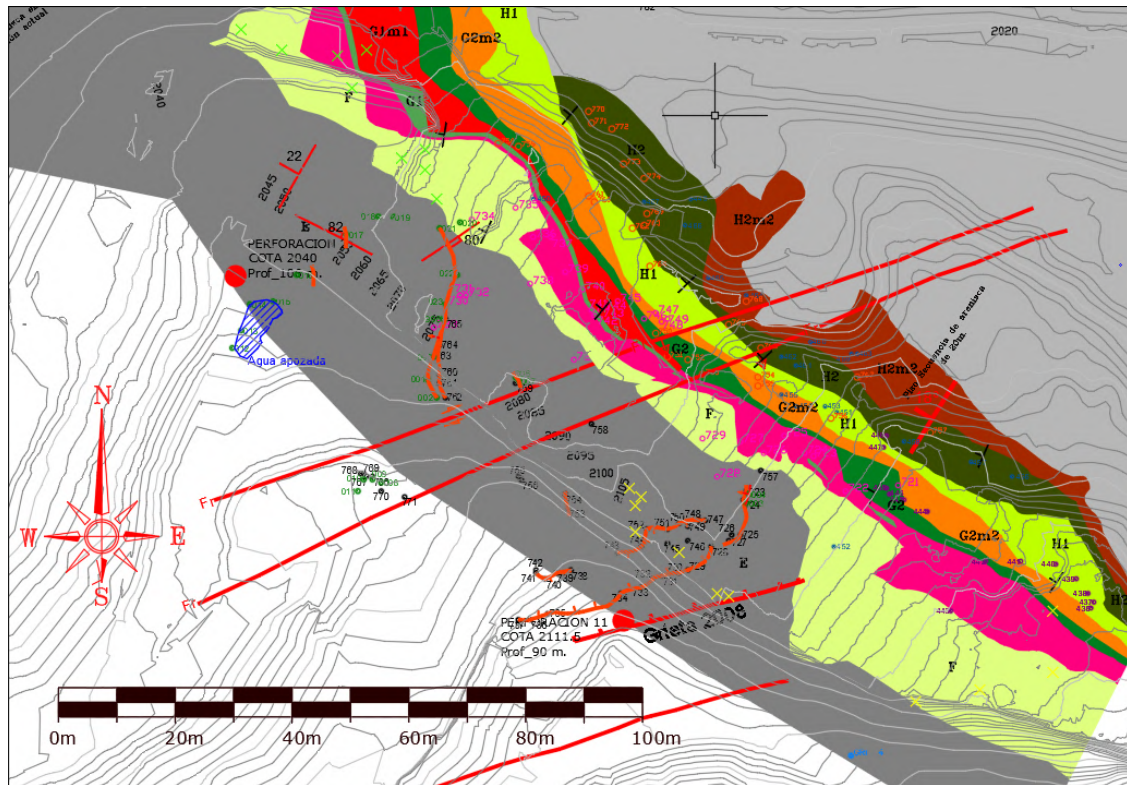
Figure A-3.: El Pedregal Mine. General plan view of the geological model.
After Geologia y Geotecnia [2011]

Table A-2.: Summary of the main structural features identified in the mine

Structure	Dip direction/Dip	Comments
D1	158/76	Joint set 1
D2	056/75	Joint set 2
E	309/19	Bedding
F1	148/62	North Fault
F2	148/84	South Fault

A.3.2.   Data adcquisition by conventional technique

An extensive experimental campaign was carried out both in the field and
computational laboratory to gather information on rock mass geometry, i.e.
joints orientation, size, and persistence. Compass and tape were used to va-
lidate the planes orientation computed from the 3D images, as well.

Table A-1.: Description of lithological sequence reported in El Pedregal Mine

	Lithology
E	Sandstone with inter-bedded thin layers of shales
F	Quartzitic medium to fine grained sandstone with siliceous cement. There are some thin layers of carbonaceous material
F1m1	Carbonaceous shale
G1	Quartzitic medium to fine grained light gray sandstone with siliceous cement
G1m1	Carbonaceous shale, with transitions to sandstone
G2	Quartzitic medium to fine grained light gray sandstone with siliceous cement
G2m2	Carbonaceous shale, with transitions to sandstone
H1	Quartzitic fine grained gray sandstone. Highly cemented
H1m1	Strata of shale with transitions to quartzitic sandstone
H2	Quartzitic fine grained gray sandstone. Highly cemented
H2m2	Strata of shale with transitions to quartzitic sandstone
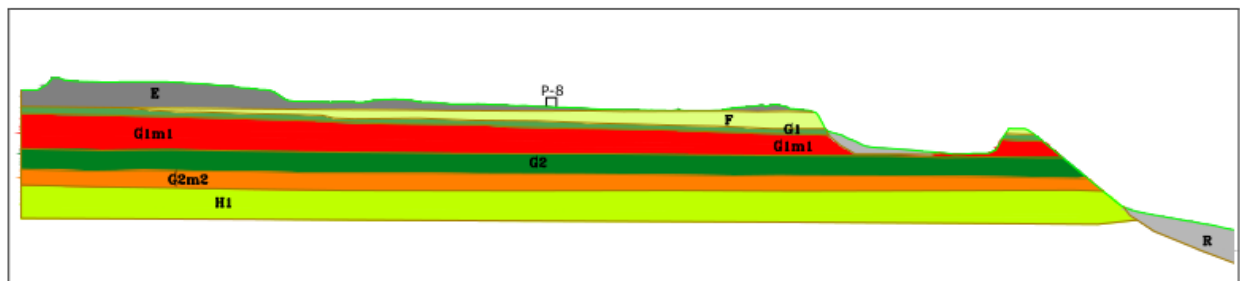
Figure A-4.: El Pedregal Mine. Typical cross section of the rock mass. After Geologia y Geotecnia [2011]

As mentioned above, the mine has had a follow-up on geology and geotechnics. As a result, a data set on joints dip and dip direction are available. The first pieces of information were collected in 1997 when the operation started. Then in 2011 a detailed geotechnical assessment to update the mine design was carried out, which included a back-calculation of a rock slope failure reported in 2000. The information was complemented in 2016 by this study. The fieldwork has focused on two main aspects, the measurement of joint planes and geomechanical characterization. The latter has been carried out mainly by block sampling and a few results from boreholes.

The collection of information on joints geometry has been addressed using

Table A-3.: Summary of the historical information collected by traditional techniques

Year	Suorce	N	Mean Dip direction	Dip
1997	1	116	138.1	83.7
		163	68.9	84.5
		23	292.8	25.0
	2	43	140.5	71.8
		56	107.5	88.7
		17	323.7	39.7
	3	87	132.3	76.6
		42	207.2	86.9
		13	304.7	23.3
2011	1-Central	23	154.3	76.6
		10	53.3	79.2
		6	305.3	29.0
	2-South	30	150.9	81.2
		26	62.5	78.0
		17	307.1	20.7
	2-North	71	157.2	76.8
		75	59.0	75.0
		34	308.7	23.7
	3-South	14	161.3	82.1
		19	62.5	74.5
		8	325.4	17.7
2016	1	25	132.7	82.4
		23	70.8	75.7
		20	292.8	24.3
	2	15	124.5	77.7
		38	53.4	84.6

conventional compass measurements. In 1997 just a compass and approximate location were used. Then in 2011, the GPS location was added. Since then, information was collected by the author of this thesis. Then in 2016 and 2017, new information was collected following the same procedure.

Initially, orientations were clustered using a k-means algorithm. Then, mean orientation was defined. This procedure was carried out in Orient software [Vollmer, 2015]. A summary of the information collected by traditional techniques is included in Table **A-3**, where N corresponds to the number of measured orientations.

A.3.3.   Data acquisition by the remote system

The author, a civil engineer with experience in rock mechanics, carried out the data acquisition by the remote system, assisted by a research assistant. As preparing task, the author took a training course in Graz, Austria for operating ShapeMetrix3D.

The characterization of the rock slope with ShapeMetrix3D consists of four steps [Gaich et al., 2004]:

1. Image data acquisition

2. Generation of 3D images

3. Assessment of 3D slopes

4. Derivation of descriptive rock mass parameters

Figure **A-5** shows the parts of the imaging system. It consists of a calibrated camera to capture the images, a range pole to scale the images, and processing software divided into components to construct the 3D image, normalize, trim and reference the image and perform the mapping. With this system, the current operation of the slope was characterized following the steps depicted in Figure **A-6**.

Figure A-5.: Componets of ShapeMetrix3D system. After GmbH [2010]

The slope was divided into 20 regions. At each region, the range pole is located on one side of the measurement area as shown in Figure **A-7** to
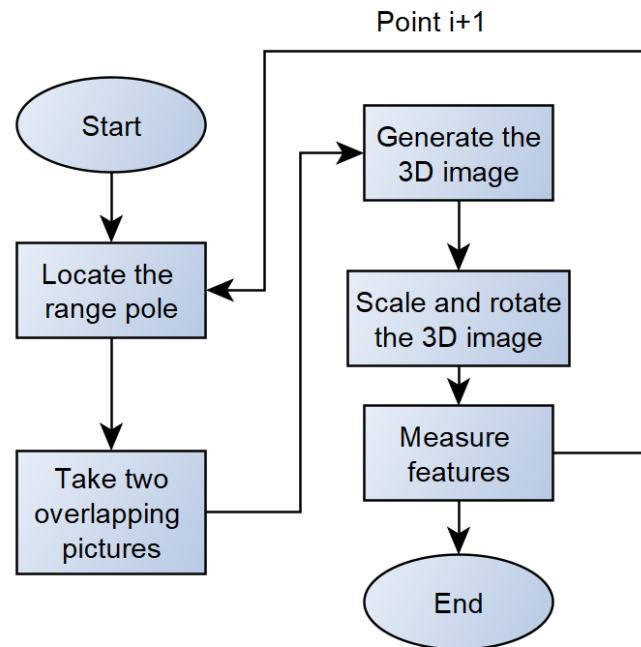
Figure A-6.: Slope characterization with ShapeMetrix3D

prevent interference with the measurements. The range pole is kept vertically aided by an incorporated bubble level. It is crucial that this reference element appears in both pictures.

In Figure **A-7** two disks are distinguished. Those elements are separated a fixed distance (2.35m). This distance allows scaling the image to the actual size during processing.

Aside from scale, the range pole is also used for orientating the information to the magnetic north (although the kinematic analysis might be carried out using local coordinates). The ShapeMetrix3D manufacturer recommends measuring the orientation of a line within the image area by using a compass. Then data can be rotated based on this measurement.

In order to reduce the bias induced by this procedure, two target points on the slope were surveyed (see Figure **A-7**). Based on the orientation of the line defined by these points the image was rotated.

Then, the operator takes two consecutive overlapping images with the calibrated camera, as shown in Figure **A-8**. The separation between pictures (baseline) must range between $0,125 * d$ and $0,20 * d$, where d is the distance
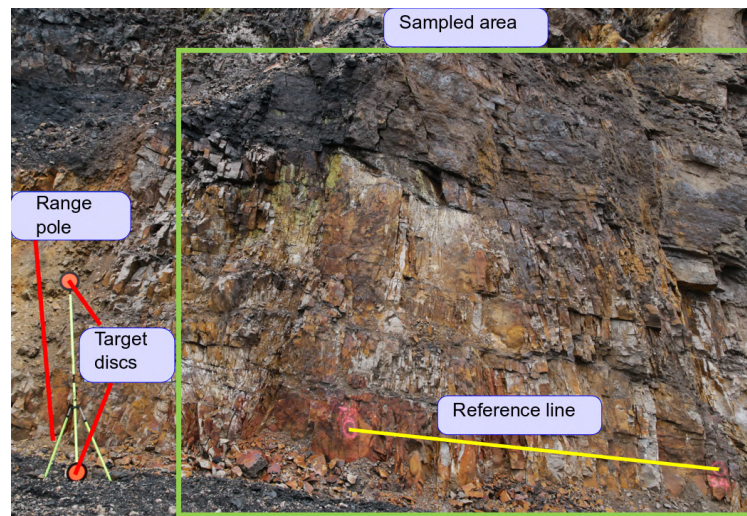
Figure A-7.: Surveyed points and orientation line used for image rotation.

from the camera to the slope.
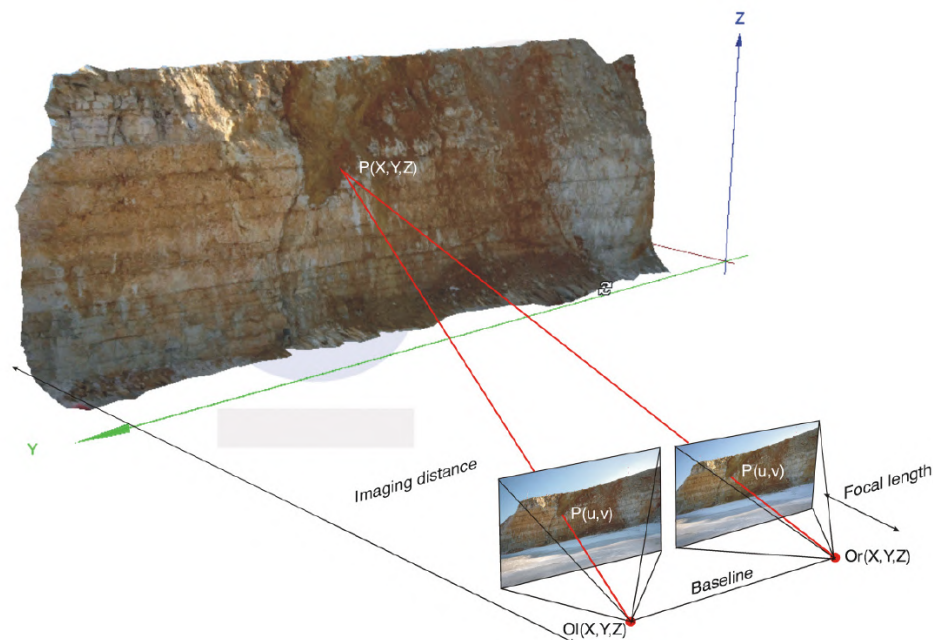
Figure A-8.: 3D image model reconstructed form two overlapping images. After GmbH [2010]

Subsequently, the SMX ReconstructionAssistant software component interpolates the pair of photographs to construct a 3D image of the slope. Then, the SMX Normalizer scales and rotates the image. First, the discs of the range

pole are located on both pictures and scaled based on the 2.35m separation. Likewise, surveyed points are located on both pictures, and the reconstructed image is rotated accordingly. Besides, editing options are available including trimming. A detailed description of the processing software is included in GmbH [2010].

Once the 3D reconstructed image is appropriately scaled and rotated, the next step is to identify and measure the geometrical features of the rock mass. Figure **A-9** shows a reconstructed 3D image using ShapeMetrix3D.
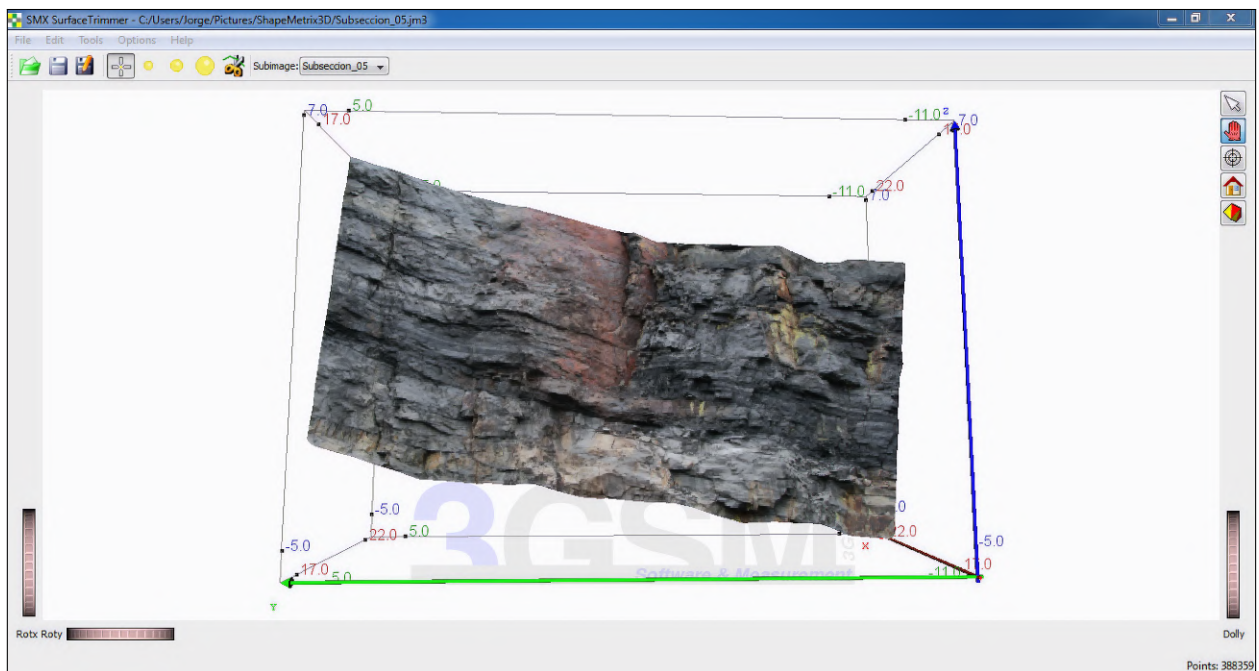
Figure A-9.: 3D image form El Pedregal Mine generated by ShapeMetrix3D

The described procedure was applied in slope highlighted in Figure **A-10**, in which the operation focused back then. A summary of the information collected on joints orientations is included in Table **A-4**.
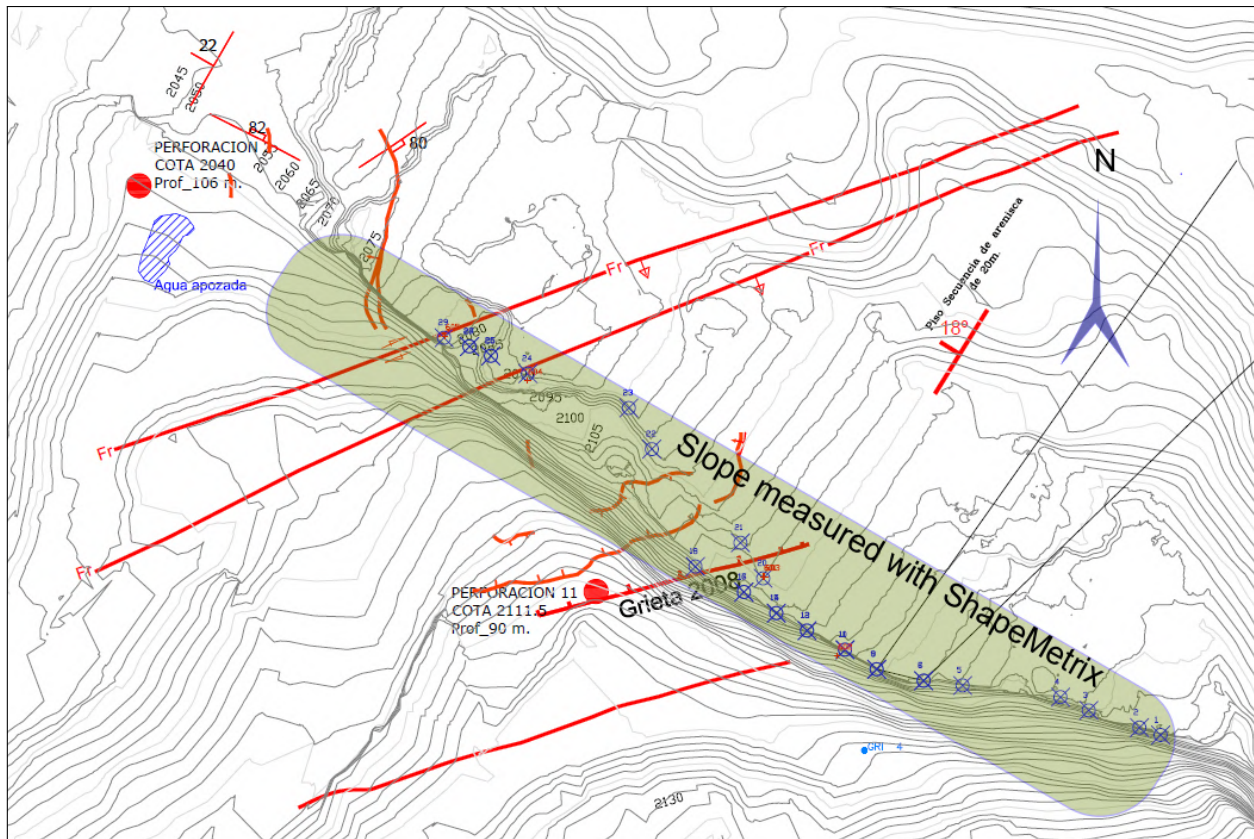
Figure A-10.: El Pedregal Mine. Slope mapped with ShapeMetrix3D

A.3.4. Measuring geometrical features

The measurements of rock mass structure are carried out by JMX Analyst module. This is a semi-automatic process, in which a trained operator identifies every joint trace and/or plane.

For traces, the operator defines polylines for each trace. Then the software defines the best-fit plane to that 3D polyline. In this case, at least three off plane points are required. As for planes, a polygon surrounding each plane is defined. This is done automatically or by the user. For each identified feature, the software defines the mean plane orientation, as well as its extension.

With the procedure just described, detailed information on the joint planes of the main slope plane was obtained. In fact, it was possible to measure the orientation of 2152 planes, during one day of fieldwork. This represents an increment of 91 % on the number of planes, compared with the informa-

Table A-4.: Summary of information collected with ShapeMetrix3D

Station	N	Joint set	Mean Alpha	Mean Beta	Station	N	Joint set	Mean Alpha	Mean Beta
Station 1	15	1 Fault	137.2	82.2	Station 9	49	1 Fault	169.8	81.9
	46	2	19.0	83.8		26	2	67.1	80.9
	22	3 Bedding	301.9	14.8		18	3 Bedding	280.0	13.8
Station 2	41	1 Fault	97.5	78.6	Station 10	36	1 Fault	151.9	74.2
	120	2	14.1	80.8		70	2	41.8	79.1
	86	3 Bedding	279.2	15.3		42	3 Bedding	287.5	17.8
Station 3	20	1 Fault	144.2	85.1	Station 12_lower	28	1 Fault	151.9	74.2
	44	2	11.6	88.1		34	2	41.8	79.1
	40	3 Bedding	228.9	16.0		9	3 Bedding	287.5	17.8
Station 4	60	1 Fault	184.0	87.7	Station 12_upper	20	1 Fault	157.4	76.4
	13	2	66.9	70.6		23	2	37.6	78.0
	21	3 Bedding	342.5	16.2		30	3 Bedding	281.9	17.8
Station 5	82	1 Fault	353.8	88.6	Station 13_lower	40	1 Fault	145.4	65.3
	92	2	58.0	78.9		22	2	55.4	84.4
	71	3 Bedding	302.2	17.6		22	3 Bedding	328.7	17.1
Station 6	54	1 Fault	0.2	86.1	Station 14	48	1 Fault	165.1	81.7
	64	2	58.7	88.3		39	2	43.2	82.6
	49	3 Bedding	321.5	14.1		21	3 Bedding	309.2	19.2
Station 7	42	1 Fault	174.9	88.1	Station 15	84	1 Fault	150.3	84.8
	42	2	54.8	85.2		75	2	44.3	80.9
	35	3 Bedding	296.6	15.3		30	3 Bedding	289.9	16.4
Station 8	33	1 Fault	152.2	77.4	Station 16	48	1 Fault	164.9	78.8
	62	2	45.5	82.1		43	2	43.4	80.0
	24	3 Bedding	290.7	11.3		50	3 Bedding	294.2	17.5

tion collected manually, during the field campaigns in 1997, 2011 and 2016 altogether, covering a larger area.

Additional advantages that were evident during the fieldwork were:

1. Planes dip and dip direction can be weighted by size. This is important, because there is a high variability on the size of planes that can be measured

2. An average orientation of planes is obtained, which is an essential improvement compared to the compass that measures small areas, and so, the average orientation becomes subjective

3. An actual image of the slope is obtained. Then, both engineers and geologist grasp a better idea of the features of the rock mass

4. The SMX analyst generates an objective and reproducible measurement of separation of joints

5. Results are conveniently described and stored to subsequent reliability computations.

A sample of the information obtained from the image processing with Shape-Metrix is included in Figure **A-11**. ShapeMetrix3D also generates a stereonet with the information resulting from each 3D image and performs a k-means clustering. An example of this is presented in Figure **A-12**
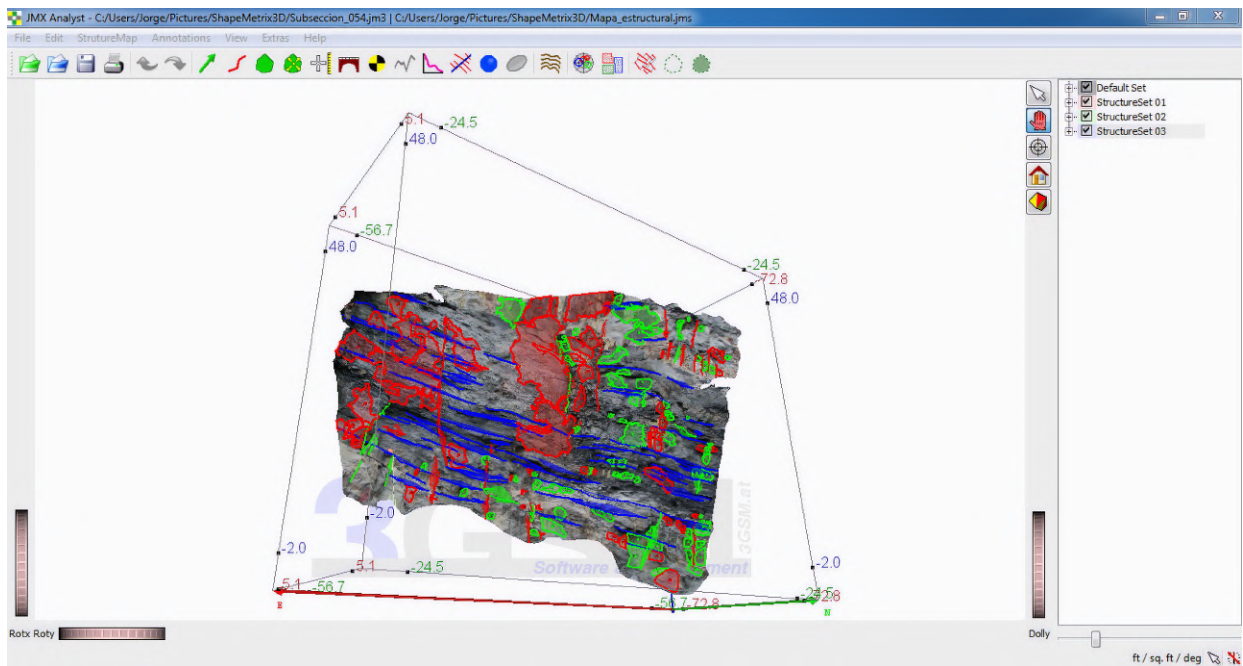
Figure A-11.: Example of 3D image processing. Planes are highlighted according to clustering

A.4. Results

In the following, a description of the information collected by using Shape-Metrix3D is presented, as well as, a comparison with the results obtained by traditional compass technique.

One of the first noticeable differences has to do with the area measured by the remote system. Here, the entire area can be mapped, which ranges between $150m^2$ and $450m^2$, while with the compass up to 2m high was reachable, which means that only 13 % or the area is sampled.
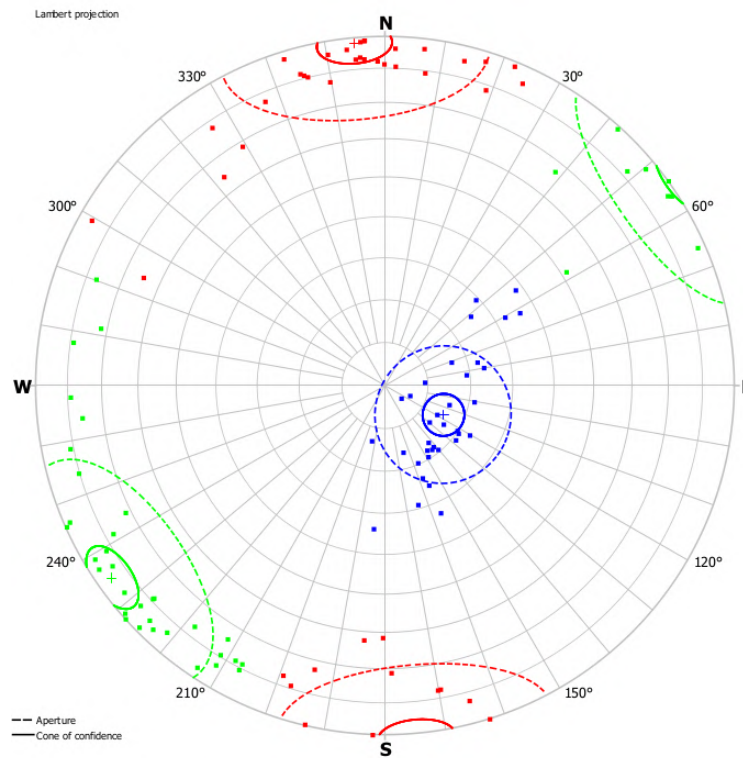
Figure A-12.: Poles plotted on stereonet for Station 7

A.4.1. Orientation

Orientations measured by using the compass and the remote system are shown in Figure **A-13**. At first glance, the poles diagrams show similar trends, i.e. There are three joint sets: one for the bedding (blue) and two steep joint sets (green and red).

Statistics about the measured data are included in tables **A-5** and **A-6**. Other than the amount of information collected, it is important to highlight how the information is distributed among the different joint sets. For the compass, most of the data (90 %) belong to joint sets 2 and 3, only 56 poles were measured on the bedding. It was complicated to locate the compass along such a discontinuity.

This bias is less evident for ShapeMetrix, where 35 % of the information belong to Joint set 1, 38 % to joint set 2 and 27 % to the bedding. The difference on the information distribution has to do with the drawbacks to measure bedding planes, since most of the are not suitable to be measured
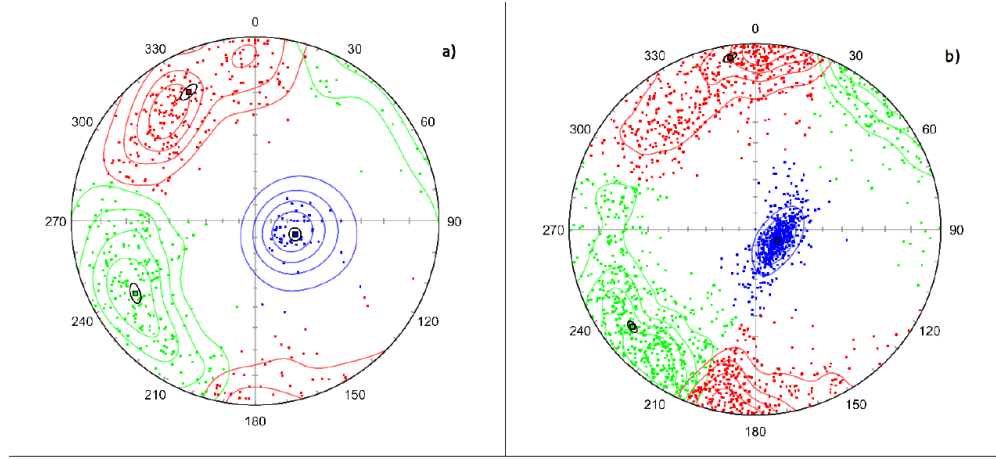
Figure A-13.: Poles measured in El Pedregal mine. Left: measured by compass. Right: measured by ShapeMetrix3D

by compass, since only the trace can be mapped.

In regard the measured angles, the highest difference is obtained for joint set 1, where the dip direction difference is 18°, and the dip difference is 10°. This is the result of the difficulties for measuring vertical angles in this rock mass. This is the case for the bedding dip direction. It also reflects the difficulties to measure these planes.

Results demonstrate that there are similar trends in the joint sets orientation.

Table A-5.: Mean orientation of planes using compass

	Joint set 1	Joint set 2	Joint set 3 (bedding)
Number of poles	272	235	56
Dip direction(°)	152.7	58.9	288.2
Dip(°)	76.5	74.5	25.7
Confidence cone	2.8	3.0	3.7

A.4.2. Spacing and joint trace length

Tables **A-7** and **A-8** summarize trace length and spacing results respectively. The first evident result is the amount of information collected. For the

Table A-6.: Mean orientation of planes using ShapeMetriX3D

	Joint set 1	Joint set 2	Joint set 3 (bedding)
Number of poles	657	732	521
Dip direction(°)	171.5	51.6	297.39
Dip(°)	86.2	80.4	15.1
Confidence cone	1.1	1.2	1.1

nonconventional technique 1810 measurements of length were collected, while 465 for the tape. As for spacing, this difference is even more dramatic; 5338 orientations were measured by ShapeMetriX3D and just 408 by tape. Moreover, as orientations, spacing, and trace length measurement are evenly distributed when collected by ShapeMetriX3D.

The image system provides more reliable spacing measurements than traditional tape because there is no chance for defining actual distances directly during the field work. Only apparent spacing can be measured. Instead, ShapeMetriX3D defines objectively the distance between two planes based on the normal vector. This task requires computing beforehand the joint plane orientation.

Table A-7.: Measured trace length

Technique	Parameter	Set 1	Set 2	Set 3 (Bedding)
Trace length	Number of measurements	233	225	7
	Mean (m)	1.91	1.66	0.93
	Standard deviation (m)	2.61	1.68	0.73
ShapeMetriX3D	Number of measurements	590	721	499
	Mean (m)	1.16	1.03	2.71
	Standard deviation (m)	0.93	0.93	1.97

Beyond the amount of information, results on spacing are more reliable than those obtained by conventional tape measurements, since there is no chance for defining objective actual separation during the field work. Only apparent

Table A-8.: Measured joint spacing

Technique	Parameter	Set 1	Set 2	Set 3 (Bedding)
Spacing	Number of measurements	205	200	3
	Mean (m)	0.26	0.32	0.58
	Standard deviation (m)	0.33	0.35	0.65
ShapeMetriX3D	Number of measurements	1690	2098	1550
	Mean (m)	1.77	1.40	1.28
	Standard deviation (m)	2.63	2.13	1.75

spacing can be measured. This is overcome by the image technique, since it defines objectively the distance between two planes based on the normal vector. This task requires defining beforehand the joint plane orientation.

A.4.3. Time and efficiency

Concerning the time spent collecting the information, for the conventional technique it took 10 days performing the field work, plus 2 days of work office to process the information.
Regarding ShapeMetrix3D, it took 6 hours to take the required pictures in the slope. Depending on the expertise, the time spent in processing the images and measuring ranges between two and five days.

A.5. Conclusions

In this work, a short-range stereoscopy technique (ShapeMetriX3D) for collecting remotely structural information from a rock mass has been utilized. Results were compared with the traditional compass and tape technique.
One of the main advantages is related to contactless measurements. It makes accessible the whole exposed rock mass face (slope or outcrop). In conventional compass measurements, the access is constrained by the operators' ability to access and measure exposed planes. Hence, a larger area is mapped using

the short-range stereoscopy technique.

Consequently, the amount of collected information by the 3D image system is larger compared to the compass. Specifically for case studied in this project 9058 data were gathered with ShapeMetriX, including 1910 orientations, 5338 spacing, and 1810 trace length records, while with compass and tape 1436 data were measured distributed in 563 orientation, 408 spacing, and 465 length records.

Aside from the bias induced by access restriction, the remote system reduces bias generated by preference to measure only clearly exposed planes, the ability to measure traces orientation directly from the slope face, and changes in mean slope orientation in the same exposed plane.

Besides, it reduces the fieldwork time required to collect the information, which means less time for the operator to be exposed to an eventual slope instability event, rock fall or an accident, the latter is crucial in mining activities.

Regarding orientations, mean joint set orientation measured with ShapeMetriX3D and compass are consistent, despite the difference in the amount of evidence. This validates the applicability of this 3D image technique for measuring rock mass structure features.

The 3D image technique allows having reproducible and objective measurements of joint planes spacing, based on the normal distance between planes, using the same algorithm over and over. This is a significant advantage compared to tape measurement, where only apparent spacing is measured and depends on the operator expertise.

The comparative advantages of the short-range stereoscopy technique over compass and tape make it a suitable alternative to be implemented in routine geomechanical applications for rock slope design in Colombia.

Once the image is generated, it is available at any time. Hence, additional measurements or assessment may be carried out without going back to the site. Besides, it allows the geotechnical engineer to have a close and detailed view of the rock slope, which improves his/her understanding of the rock slope potential behavior.

Finally, the application of this tool in the investigation resulted in improvements in the reliability of the data used for rock mass structural characteriza-

tion, as a fairly robust, representative and repeatable database was achieved.

B. Kent distribution vs Fisher distribution

This appendix shows the calculation of the factor of safety of rock wedges, in which the joints are modeled as Kent distributed variables. The concentration and ovalness parameters are varied to verify their influence on the probability of failure and the probability function.

The deterministic wedge's geometry corresponds to a wedge failure in jointed andesite near Ankara Castle in the Bent Deresi region of Ankara City, Turkey, reported by **?**. Orientations of planes 1 and 2 are 194/44 and 103/71, respectively, and slope orientation 162/69. This geometry leads to a failure along the intersection line, as can be seen in Figure **B-1**, where joints are plotted in red and the slope in red. This orientation generates a failure of the mean wedge along plane 1. A friction angle of 30°was reported. Values of cohesion of 25kPa, 35kPa, 40kPa, and 45kPa were assumed. Figure **B-1** shows the major planes for the average joint planes and the slope face.

Both joint planes were simulated as bi-variate random variables with a Kent distribution, considering the scenarios included in Table **B-1**. These values cover a broad spectrum of possibilities, including the Fisher distribution, i.e., $\beta = 0$. For each scenario, 10000 simulations of planes were performed, by the A/R algorithm implemented according to the flowchart included in Figure **3-6**.

Given that the aim is to evaluate the influence of the shape of the distribution of joint planes on the probability function and probability of failure, only these variables were modeled as probabilistic. Other properties were assumed as deterministic.
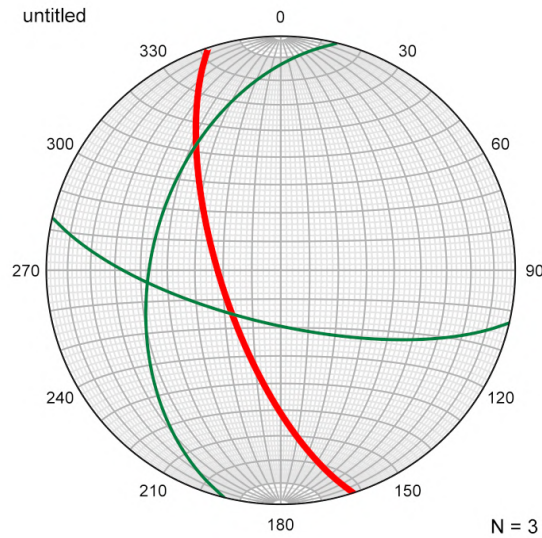
Figure B-1.: Joint planes and slope orientation assessed

Information reported in Ankara Castle in the Bent Deresi region of Ankara City, Turkey by **?**. Joints in green, slope in red

Table B-1.: Input shape parameters for rock joint planes simulation

kappa	beta	Lambda	kappa	beta	Lambda	kappa	beta	Lambda
100	0	0	500	0	0	10000	0	0
100	1	0.01	500	5	0.01	10000	100	0.01
100	10	0.1	500	50	0.1	10000	1000	0.1
100	25	0.25	500	125	0.25	10000	2500	0.25
100	40	0.4	500	200	0.4	10000	4000	0.4
100	49	0.49	500	245	0.49	10000	4900	0.49

Joints simulation

In order to illustrate the importance of considering the more general Kent distribution, over the Fisher distribution, Figure **B-2** compares the simulated data with the planes simulated using both distributions. This planes were simulated using the algorithm depicted in Figure **3-6**, firstly assuming $\beta = 0$ and then $\beta = 25$ and $\beta = 49$. This figure also shows the effect of increasing the concentration parameter κ. The higher κ, the closer the simulated planes to the mean orientation.

The procedure to compute the probability of failure is as follows. Each combination i of the simulated joint planes defines a realization. Then, the factor of safety is computed according to Eq. 3-26. The procedure is repeated for each combination of planes. As a result, a factor of safety linked to each removable wedge is obtained. From those removable blocks, the number of wedges with $FS < 1{,}0$ leads to the probability of failure. Moreover, the probability distribution function of the FOS can be computed as well.

As for the probability of failure, Figure **B-3** shows the results of different scenarios. Each line defines the probability of failure for the same concentration parameters with variable ovalness factors, which is reflected in the parameter λ.

(a) $\kappa = 100$ and $\beta = 0$ (b) $\kappa = 100$ and $\beta = 25$ (c) $\kappa = 100$ and $\beta = 49$

(d) $\kappa = 500$ and $\beta = 0$ (e) $\kappa = 500$ and $\beta = 125$ (f) $\kappa = 500$ and $\beta = 245$
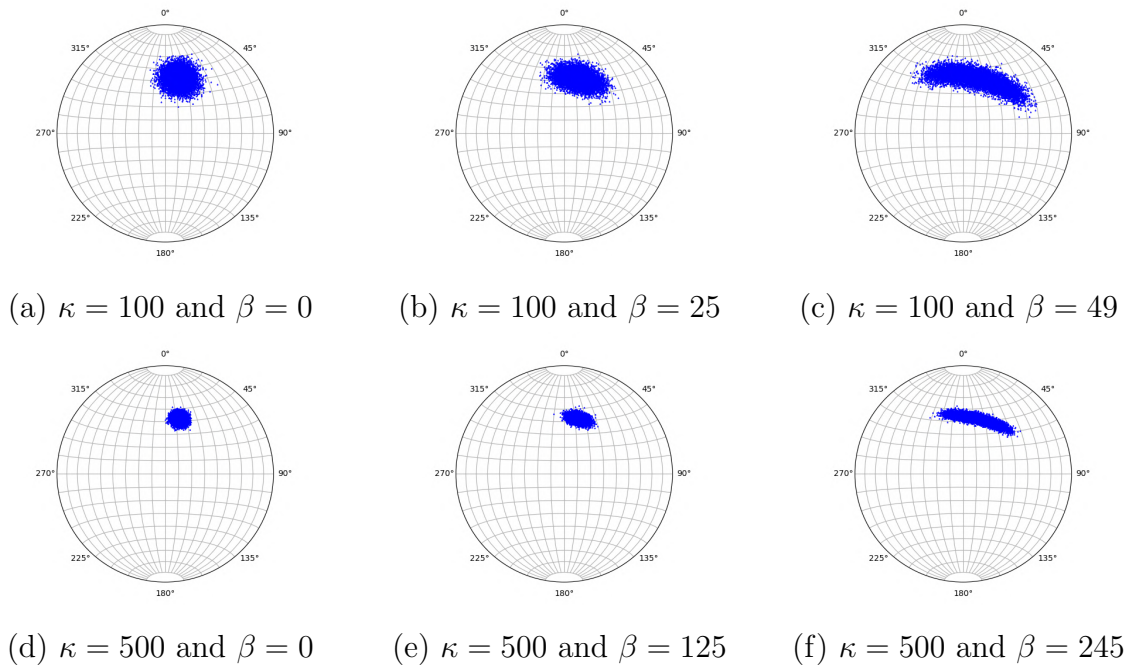
Figure B-2.: Simulation of orientation planes for joint 1. 194/44
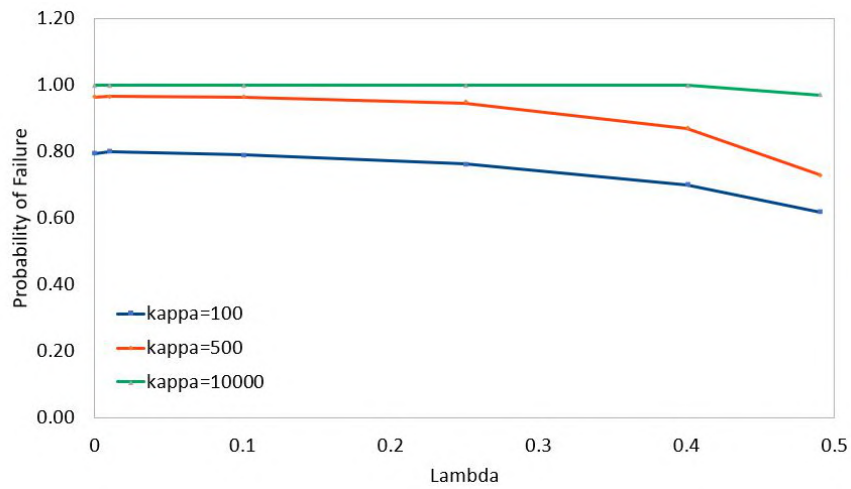
(a) c = 25 kPa
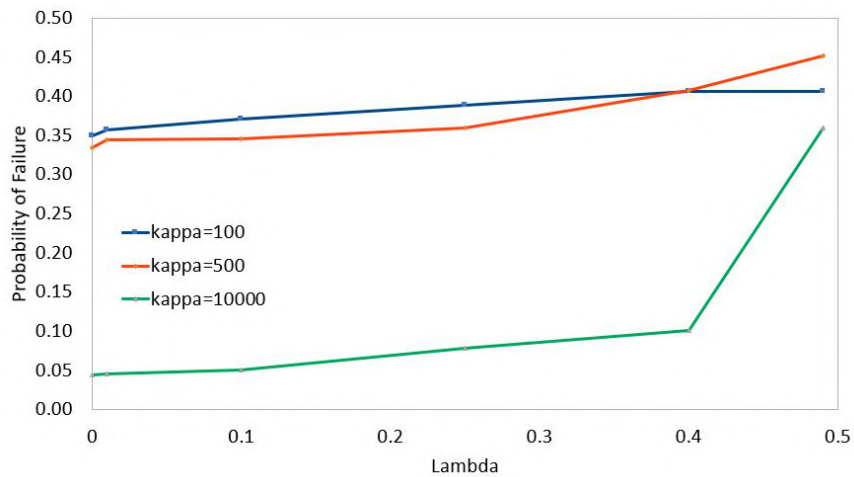
(b) c = 40 kPa
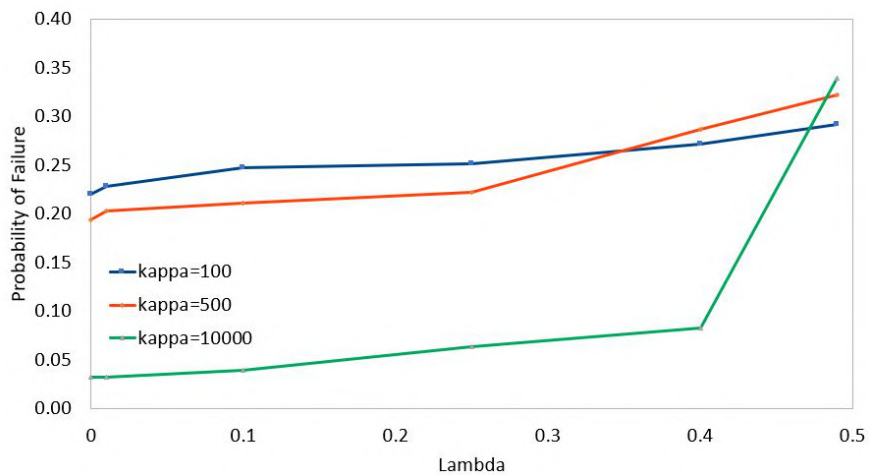
(c) c = 45 kPa

Figure B-3.: Probability of failure for different scenarios

Two facts can be distinguished from these results. First, the probability of failure changes as the concentration parameter changes, which is expected, since the wedge geometry changes at each realization. However, the variation is not straightforwardly defined. First, for the cohesion of 25kPa, as the concentration increases, the probability of failure increases. On the other hand, for values of cohesion above 35kPa, as the concentration increases, the probability of failure decreases.

Second, the ovalness has a noticeable effect on the probability of failure. For a given concentration parameter κ concentration, when the cohesion is less than 25kPa, the probability of failure reduces as the ovalness factor increases. The opposite occurs for cohesion is higher or equal to 35 kPa.

Figures **B-4** and **B-5** show the change of the probability of failure in percent, with respect to the Fisher distribution, against the eccentricity lambda. Here, when the cohesion equals 25kPa, there is a reduction in the probability of failure of 25 %, falling from 0.97 for Fisher to 0.73 for Kent with an eccentricity of 0.4. Conversely, when the cohesion equals 40kPa, and the concentration is 10000, the variation of the probability is as high as 709 %, going from 0.04 to 0.36.
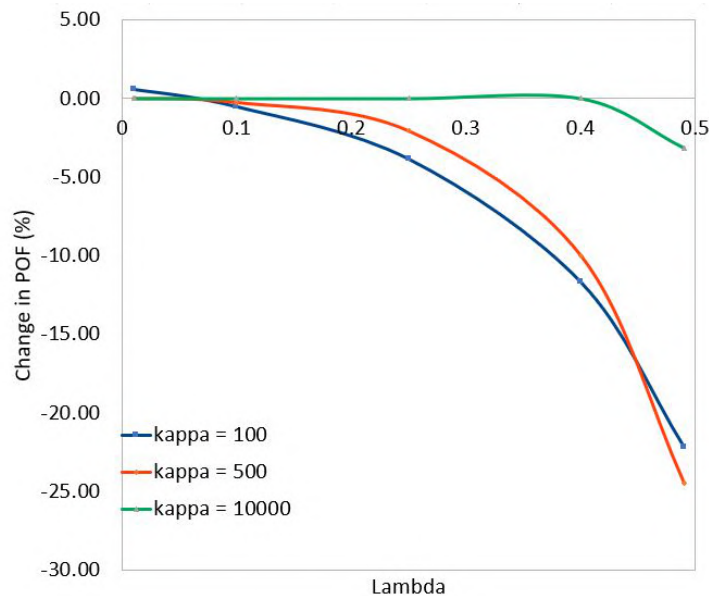

Figure B-4.: Variation of the probability of failure with respect to the Fisher distribution value. c = 25 kPa
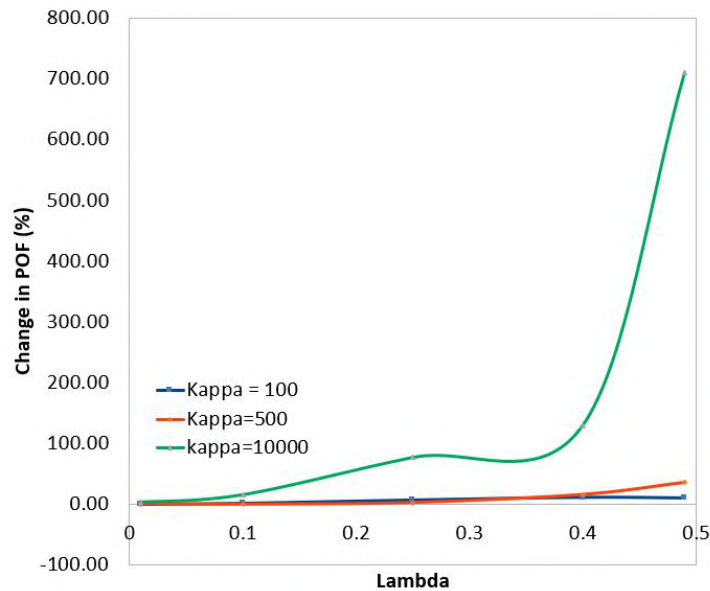
Figure B-5.: Variation of the probability of failure with respect to the Fisher distribution values. c = 40 kPa

So far, the variation of the probability of failure has been explicitly presented. Nevertheless, these variations are evident throughout the probability function for different values of the shape parameters of the Kent distribution. Figures **B-6**a, **B-6**b, **B-6**c presents the probability function for κ equals to 100, 500 and 10000, respectively, and **B-7** gathers all probability functions. These graphs reinforce the fact that there is an effect on the probability of failure of wedge stability of the input parameters selected to model the variability of input orientation. Even more, the assumption of a Gaussian symmetry distribution around the mean, has a noticeable effect on the probability function and consequently on the probability of failure.

(a) kappa = 100   (b) kappa = 500   (c) kappa = 10000
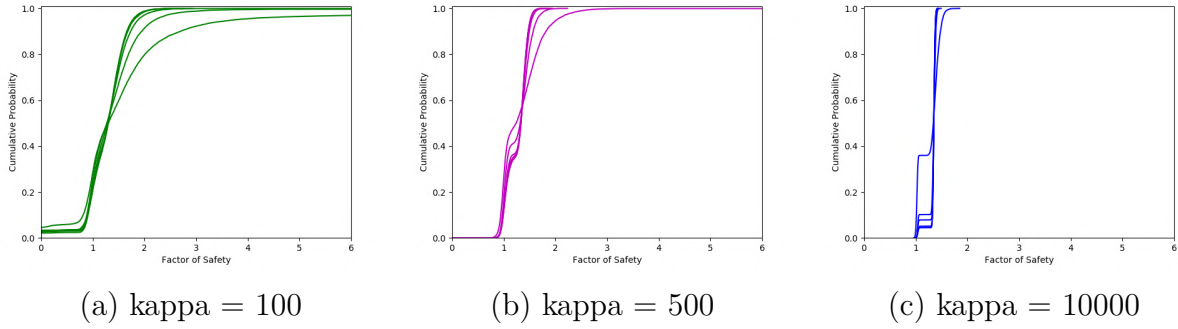
Figure B-6.: Probability function for different values of concentration and ovalness parameters. c = 40kPa, phi = 30
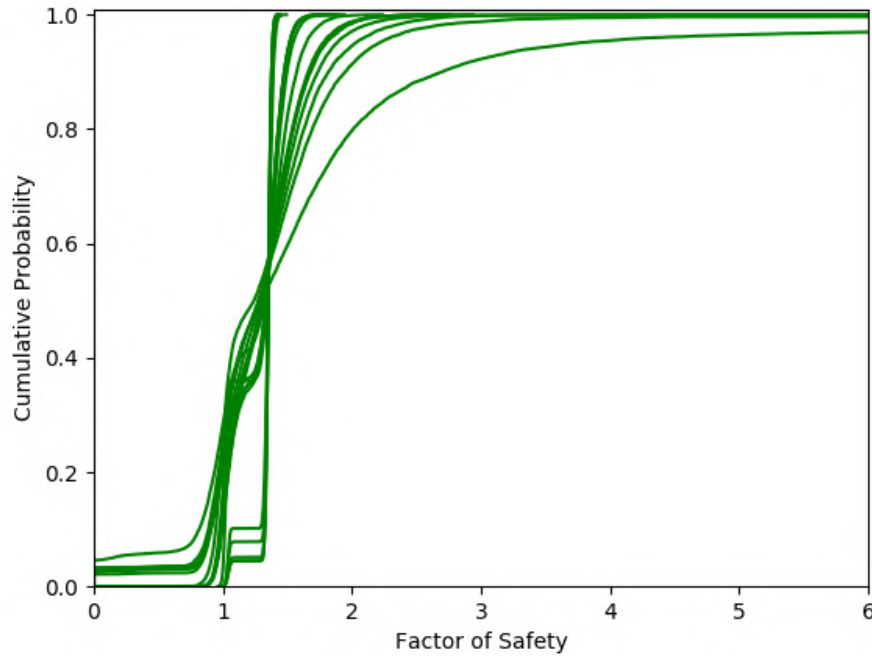
Figure B-7.: Probability function all combinations of concentration and ovalness parameters. c = 40kPa, phi = 30

C. Results supplement

In this appendix the resulting DSS computed for the factor of safety at El Pedregal Mine are included

(a) 2017. Piece of information 0

(b) 2017. Piece of information 1

(c) 2017. Piece of information 2

(d) 2017. Piece of information 3

(e) 2017. Piece of information 4

(f) 2017. Piece of information 5

Figure C-1.: DSS for the factor of safety computed for the combinations of measured planes
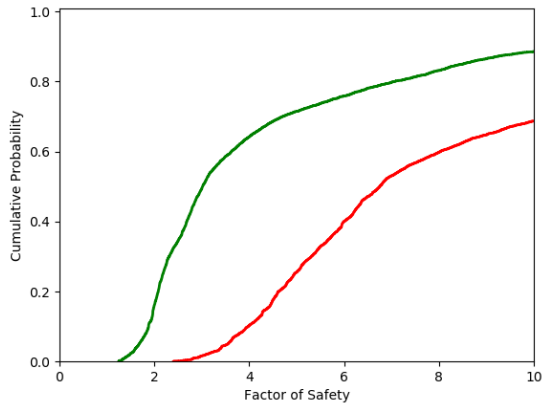
Wedge formed by bedding and joint set 2 planes (Scenario 1)

(g) 2017. Piece of information 6
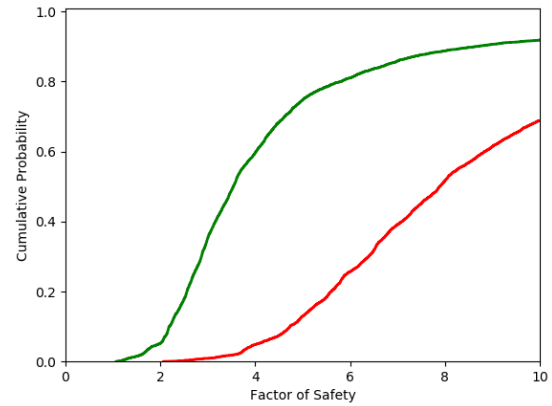
(h) 2017. Piece of information 7
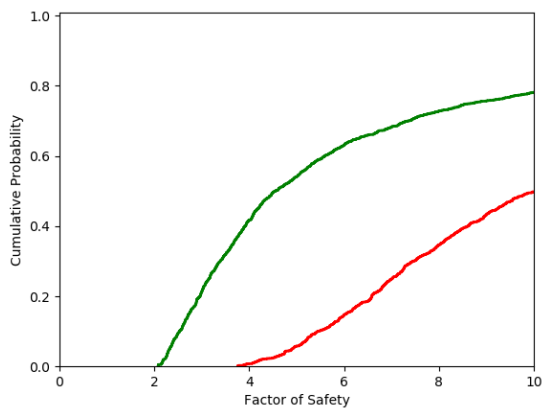
(i) 2017. Piece of information 8
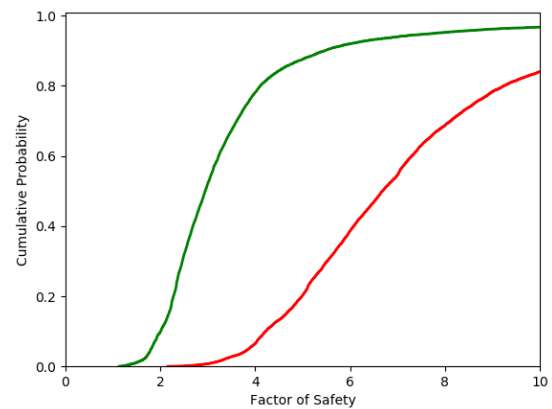
(j) 2017. Piece of information 9
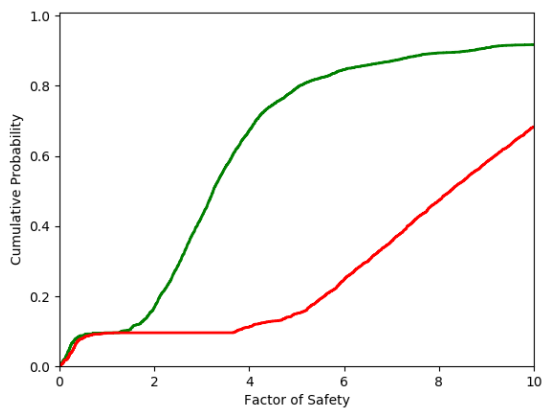
(k) 2017. Piece of information 10
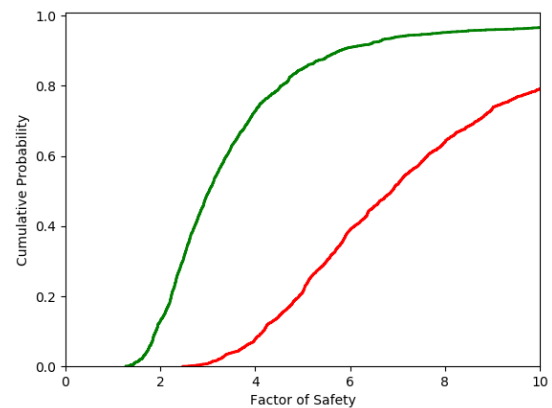
(l) 2017. Piece of information 11

Figure C-1.: DSS for the factor of safety computed for the combinations of measured planes
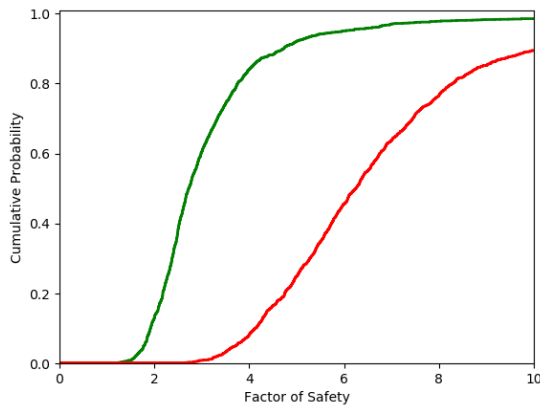
Wedge formed by bedding and joint set 2 planes(Scenario 1)

(m) 2017. Piece of information 12
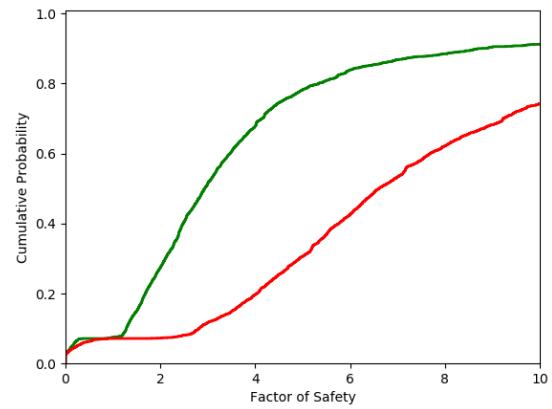
(n) 2017. Piece of information 13

(ñ) 2017. Piece of information 14

(o) 2017. Piece of information 15

Figure C-1.: DSS for the factor of safety computed for the combinations of measured planes

Wedge formed by bedding and joint set 2 planes (Scenario 1)
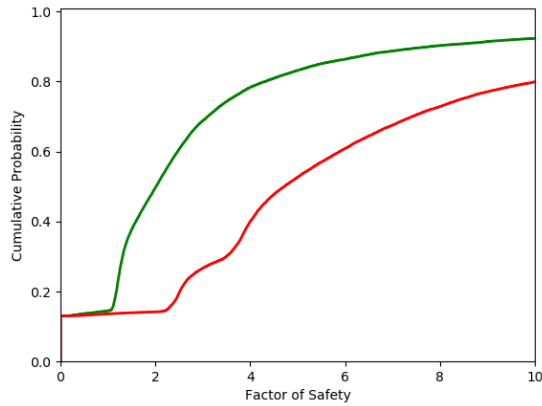
(a) 2017. Piece of information 0
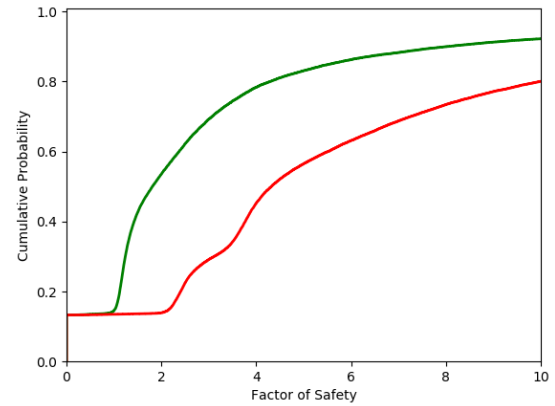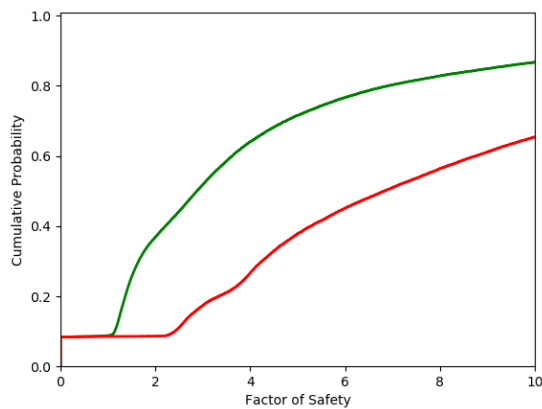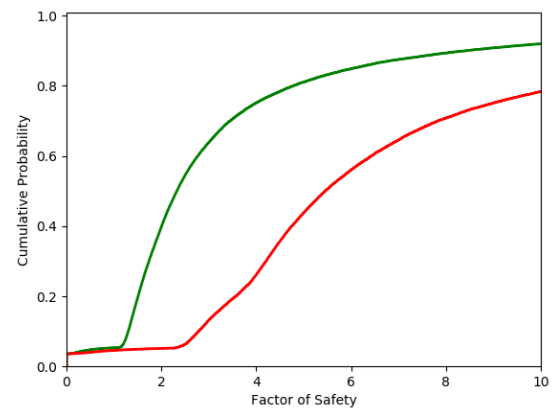
(b) 2017. Piece of information 1

(c) 2017. Piece of information 2

(d) 2017. Piece of information 3
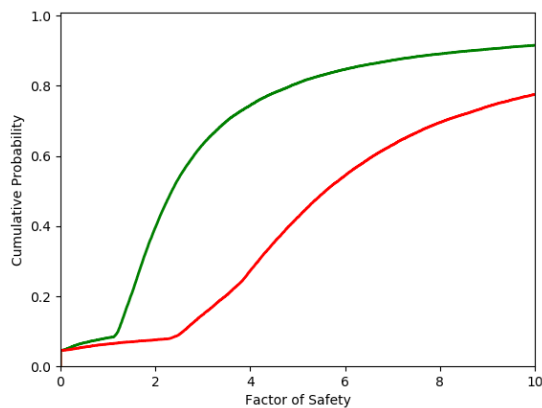
(e) 2017. Piece of information 4
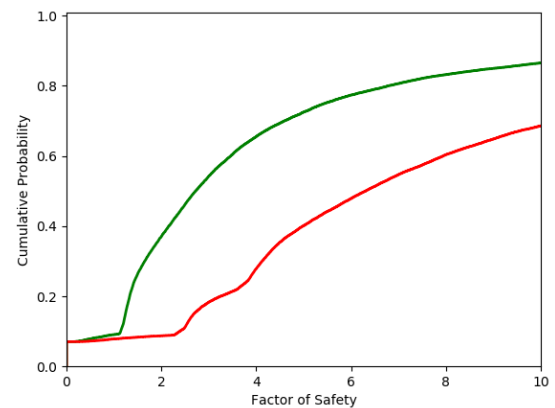
(f) 2017. Piece of information 5

Figure C-2.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable
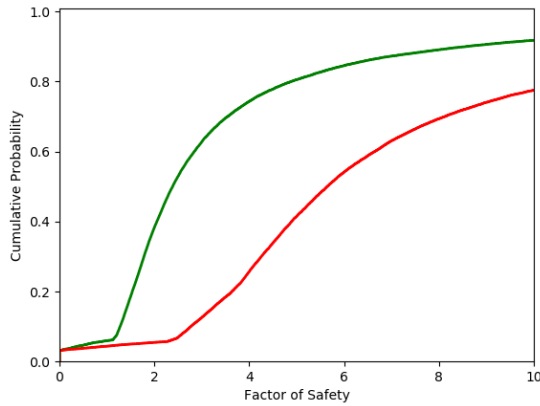
(g) 2017. Piece of information 6
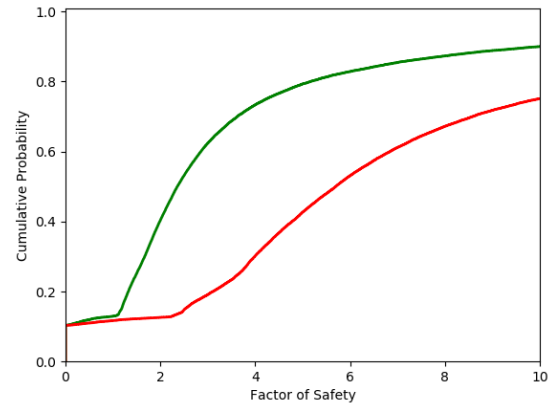
(h) 2017. Piece of information 7
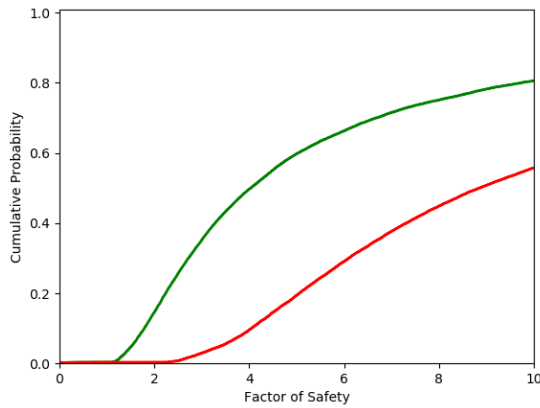
(i) 2017. Piece of information 8
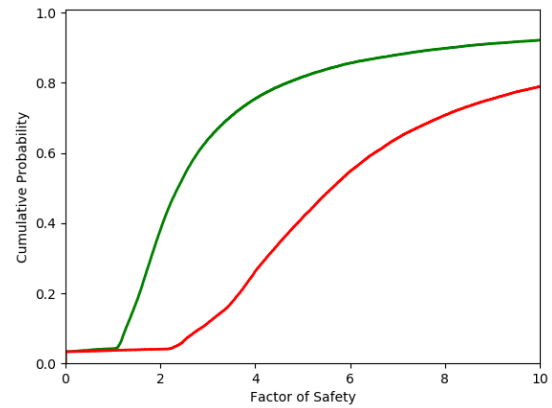
(j) 2017. Piece of information 9
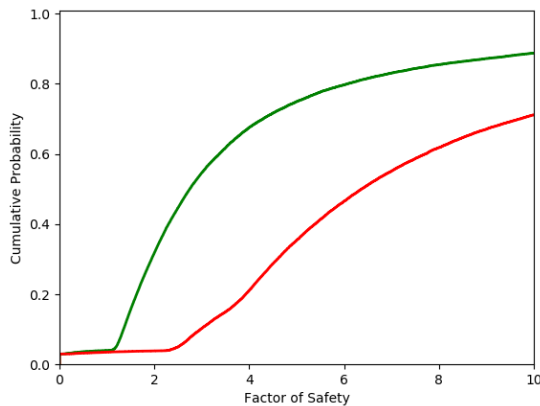
(k) 2017. Piece of information 10
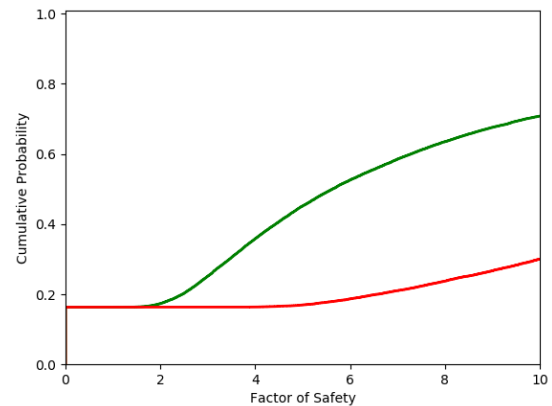
(l) 2017. Piece of information 11

Figure C-2.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable
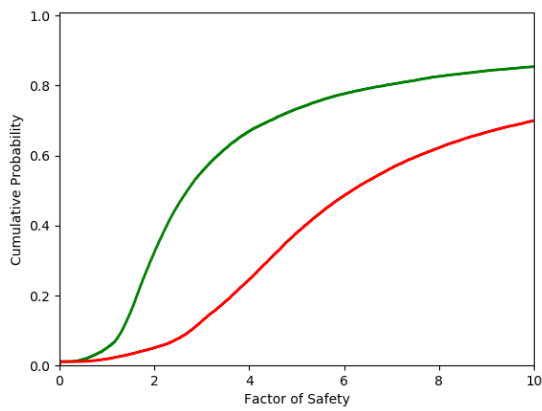
(m) 2017. Piece of information 12
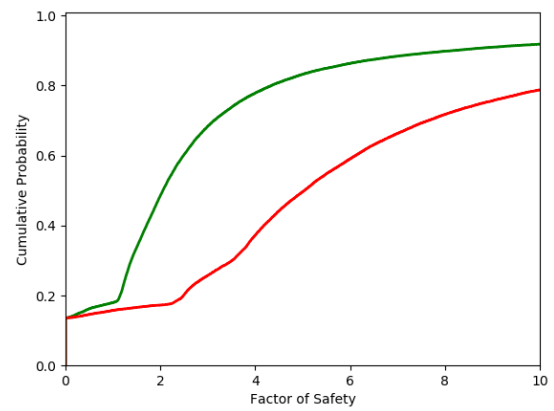
(n) 2017. Piece of information 13
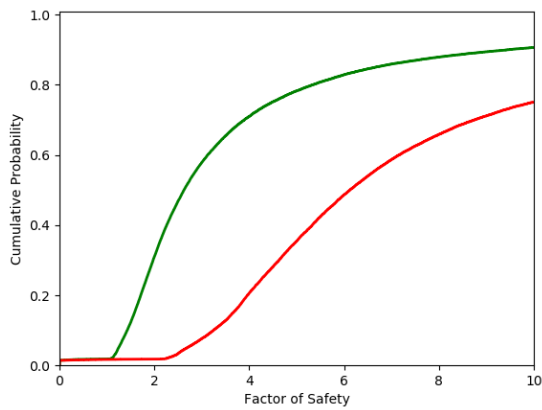
(ñ) 2017. Piece of information 14
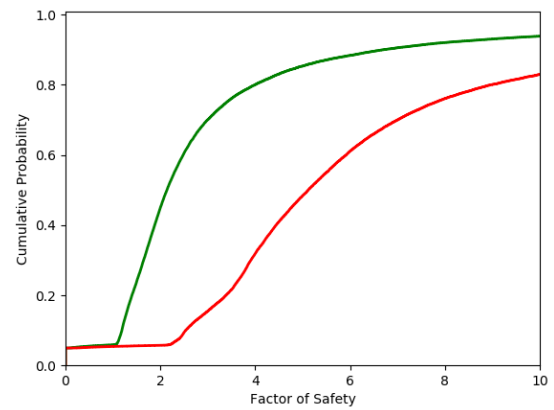
(o) 2017. Piece of information 15

**Figure C-2**.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable

Wedge formed by bedding and joint set 2 planes (Scenario 2)
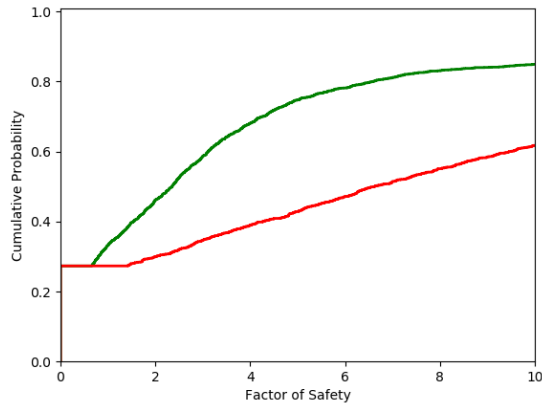
(a) 2017. Piece of information 0
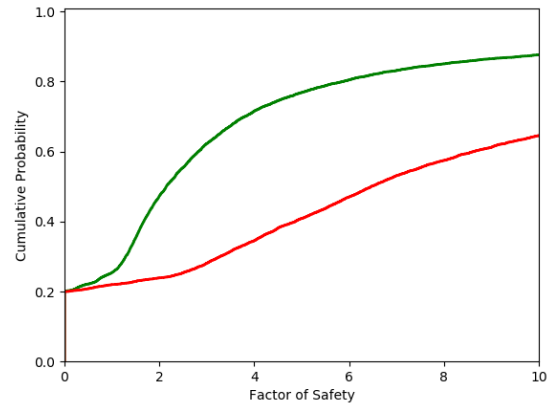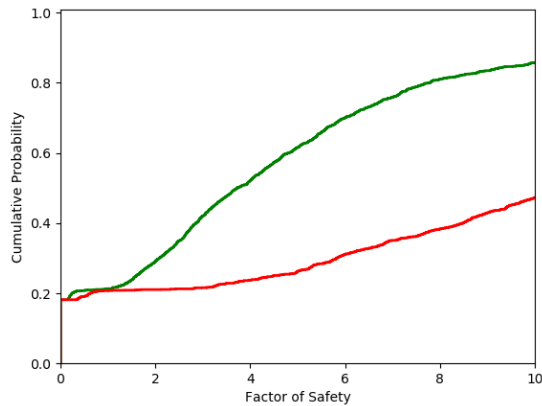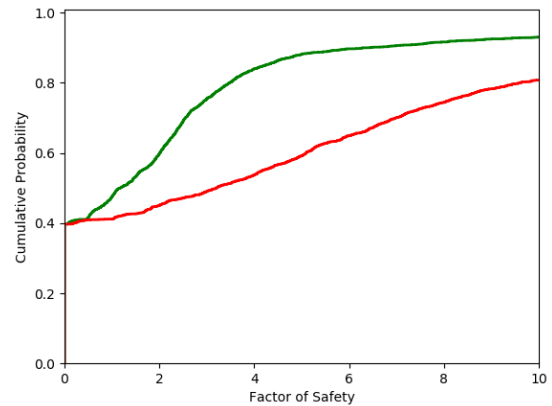
(b) 2017. Piece of information 1

(c) 2017. Piece of information 2

(d) 2017. Piece of information 3
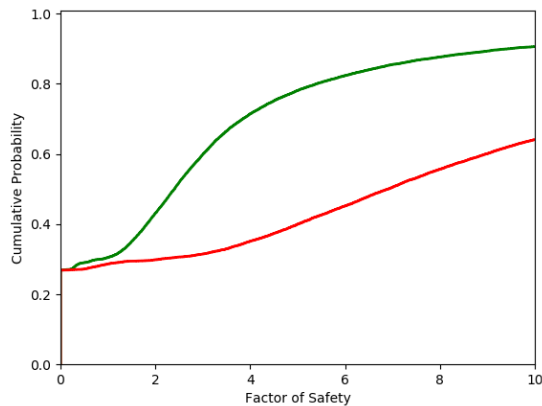
(e) 2017. Piece of information 4
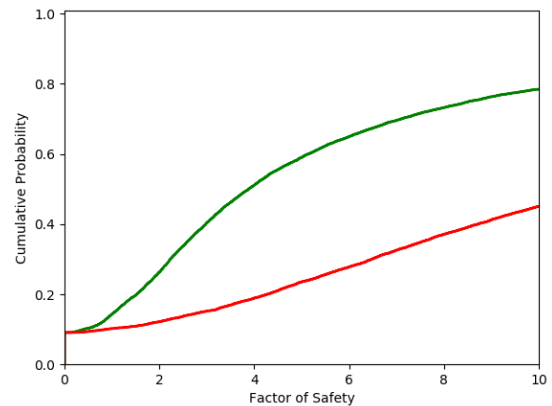
(f) 2017. Piece of information 5

Figure C-3.: DSS for the factor of safety computed for the combinations of measured planes
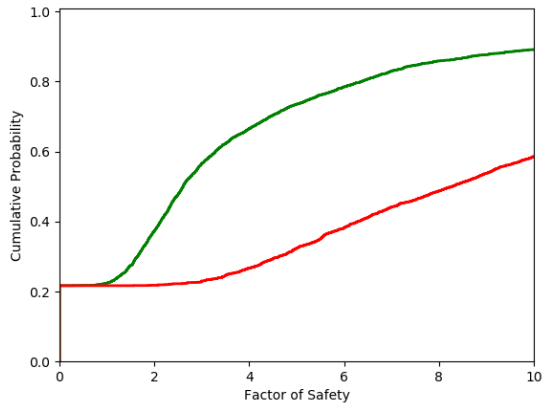
(g) 2017. Piece of information 6
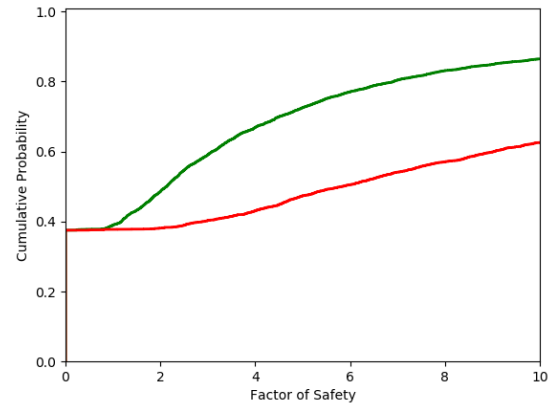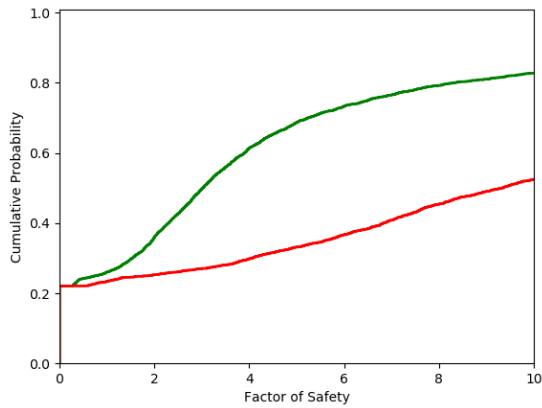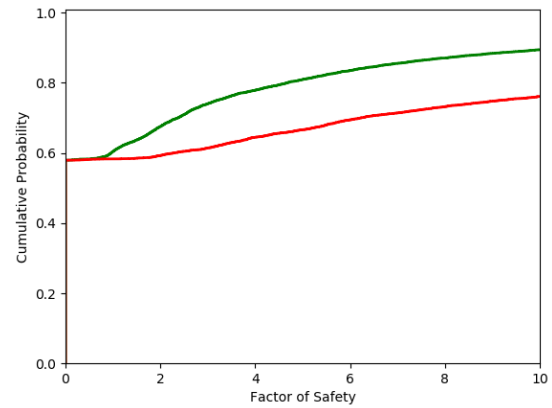
(h) 2017. Piece of information 7

(i) 2017. Piece of information 8

(j) 2017. Piece of information 9
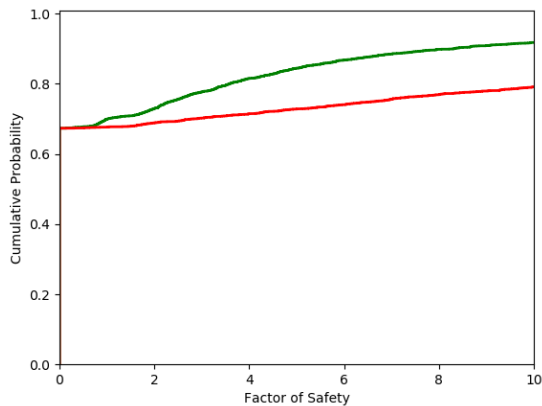
(k) 2017. Piece of information 10
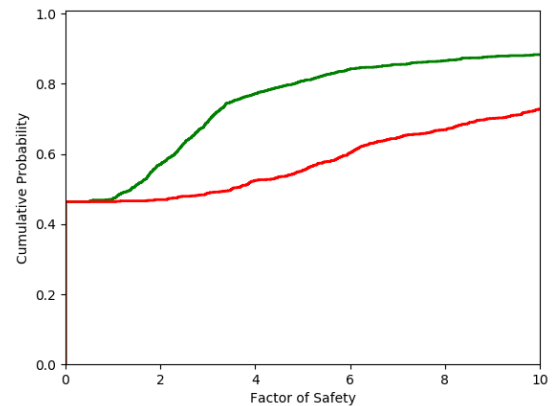
(l) 2017. Piece of information 11

Figure C-3.: DSS for the factor of safety computed for the combinations of measured planes
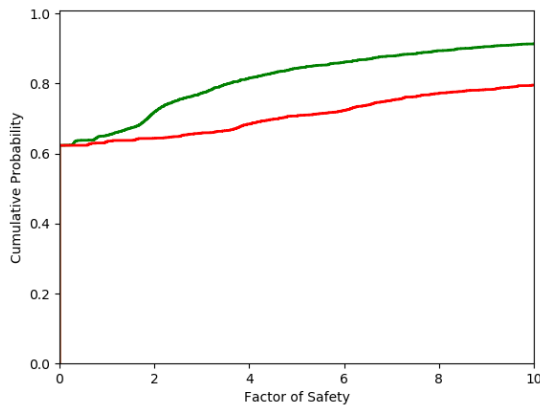
Wedge formed by joint sets 1 and 2 planes (Scenario 3)

(m) 2017. Piece of information 12
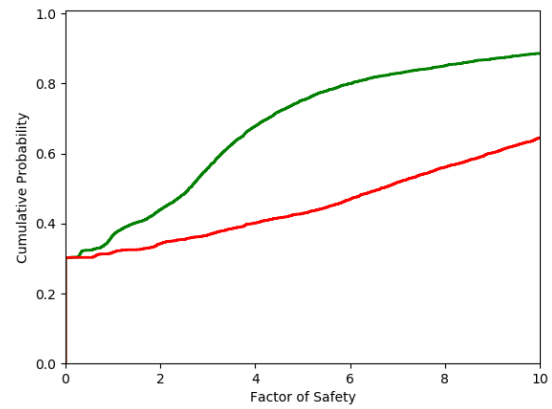
(n) 2017. Piece of information 13
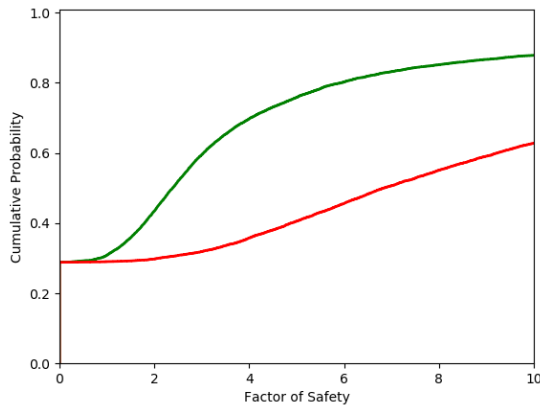
(ñ) 2017. Piece of information 14
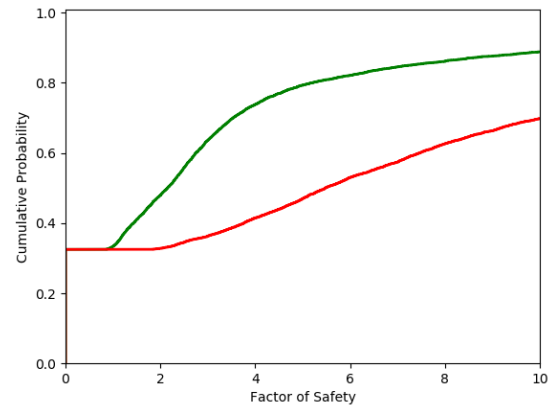
(o) 2017. Piece of information 15

**Figure C-3**.: DSS for the factor of safety computed for the combinations of measured planes

Wedge formed by joint sets 1 and 2 planes (Scenario 3)
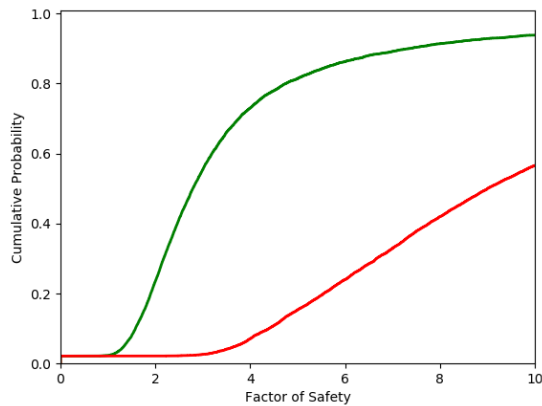
(a) 2017. Piece of information 0
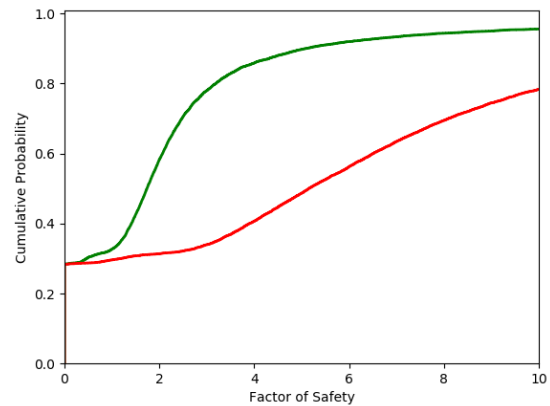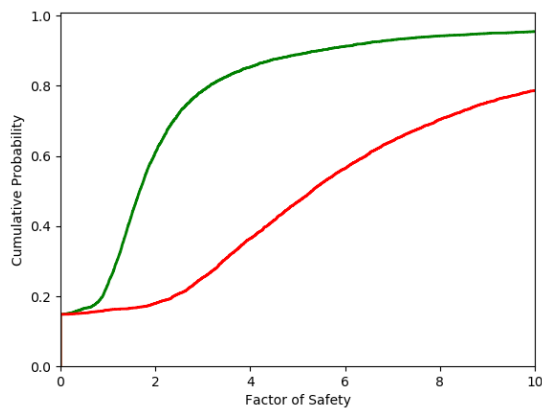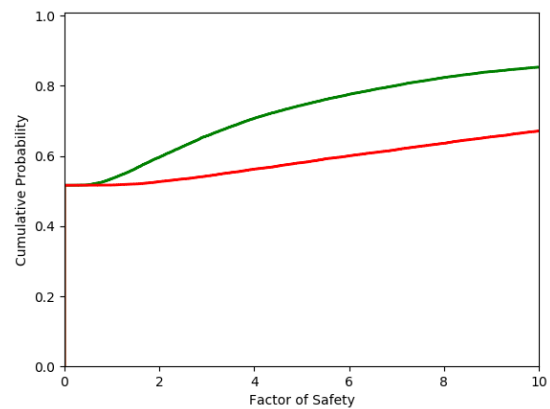
(b) 2017. Piece of information 1

(c) 2017. Piece of information 2

(d) 2017. Piece of information 3
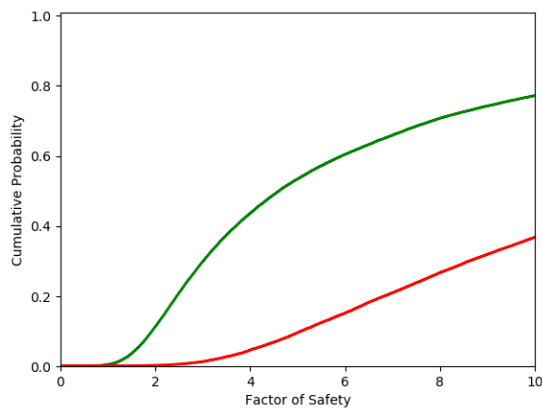
(e) 2017. Piece of information 4
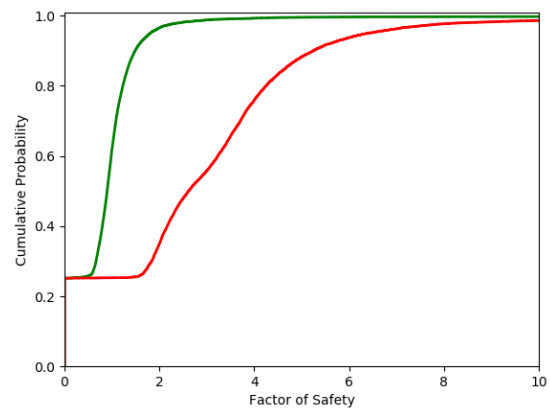
(f) 2017. Piece of information 5

Figure C-4.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable
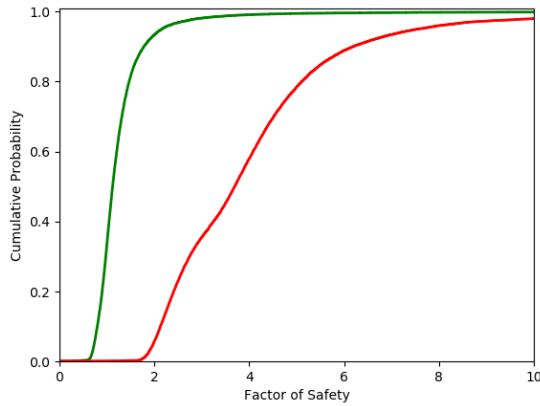
(g) 2017. Piece of information 6
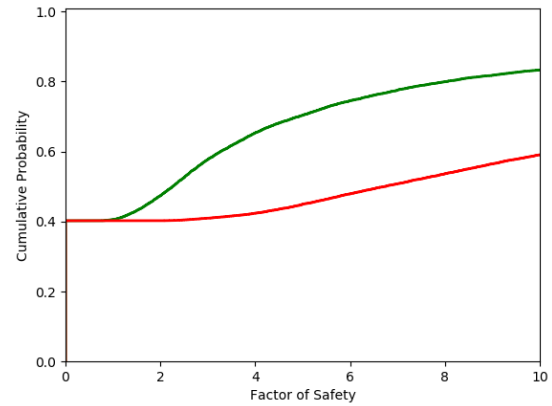
(h) 2017. Piece of information 7
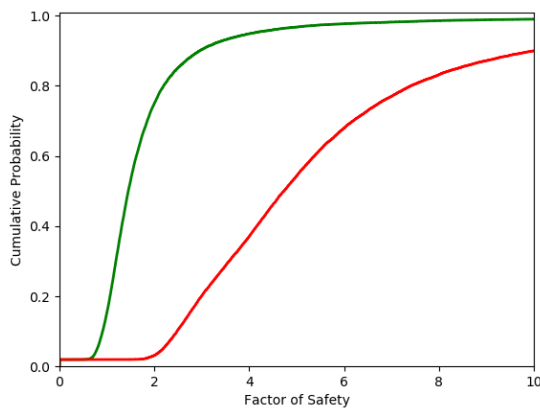
(i) 2017. Piece of information 8
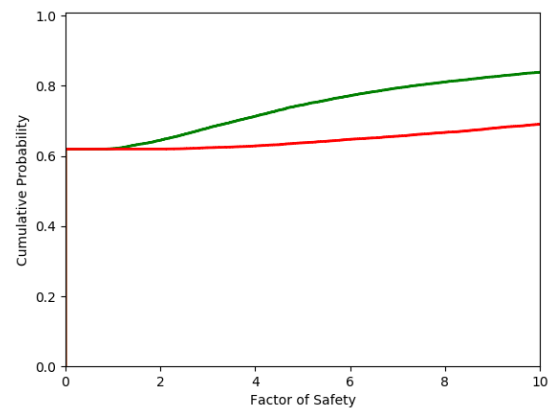
(j) 2017. Piece of information 9
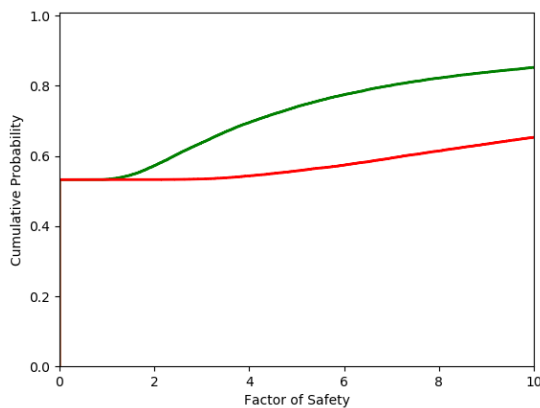

(k) 2017. Piece of information 10
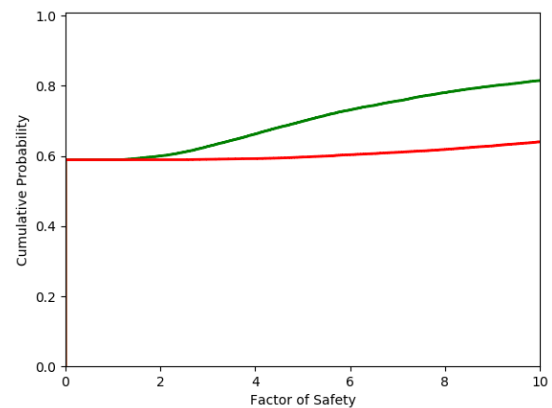
(l) 2017. Piece of information 11

Figure C-4.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable
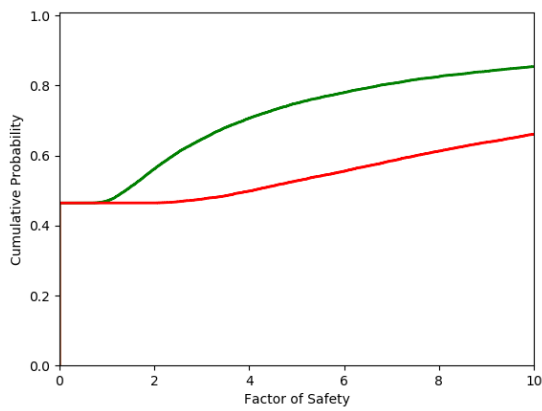
(m) 2017. Piece of information 12
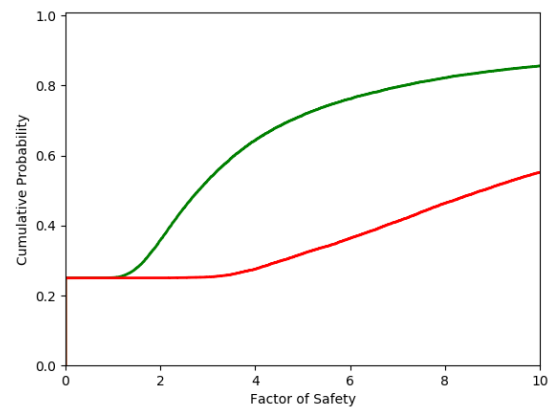
(n) 2017. Piece of information 13-4
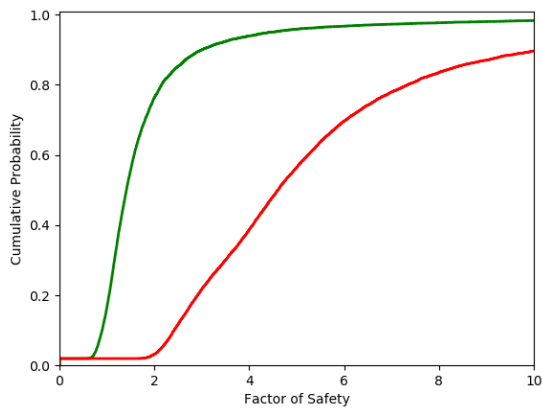
(ñ) 2017. Piece of information 14
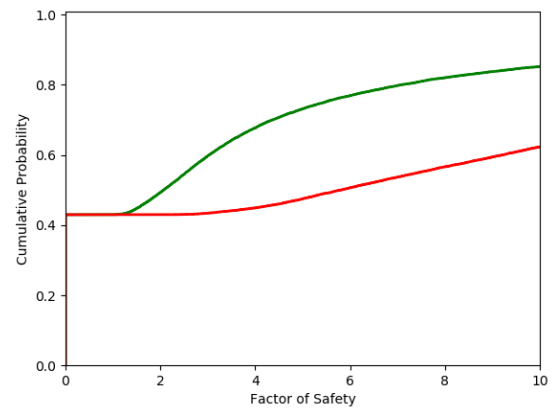
(o) 2017. Piece of information 15

**Figure C-4**.: DSS for the wedge factor of safety for the orientation simulated as Kent distributed random variable

Wedge formed by joint sets 1 and 2 planes (Scenario 4)

D. Appendix: Scripts

In this Appendix the scripts generated for the algorithms proposed in this research project are included. The algorithms suggested were programmed in Python 3.6 available at www.python.org, using the integrated development environment, IDE, PyCharm 2018.1 available at www.jetbrains.com.

```
 1  '''
 2
 3  Research Project:
 4  Reliability Assessment of Rock Slopes by Evidence Theory
 5  Universidad Nacional de Colombia sede Bogota
 6  Author: Rodrigo Hernandez-Carrillo
 7  2/8/18
 8
 9  # Copyright (c) Rodrigo Hernandez-Carrillo 2018
10
11  _____
12
13
14  This class computes the main geometruical features of an input set of
15  directional data expressed in terms of a 2 columns array. The first
16  column corresponds to the dip direction, while the second one to the dip
17  of a plane.
18
19  Functions to compute the mean, resultant, orientation matrix and
20  rotation matrix are provided.
21
22  '''
23
24  import numpy as np
25
26
27  # Function to iterate input files
28  # Function to read input matrix
29  # Output: array of planes expressed as dip direction and dip in degrees
30
31  def open_array(filename, suffix, i):
32      input_file = '{}{}{}{}'.format(filename, i, '.', suffix)
33      input_array = np.loadtxt(input_file, delimiter=',', skiprows=1)
34      return input_array
35
36
37  # When required, the following function modifies the angles to
38  # longitude and colaitude. Recommended for computing Kent distribution
39  # paraneters.
40  # Input: array in dip and dip direction
41  # Output: inout angles in terms of colatitude and longitude
42
43
44  def angle_adjustment(input_angle):
45      input_angle[:, 1] = 90 - input_angle[:, 1]
46      #print(input_angle)
47      return input_angle
48
49
50  # Function to compute the direction cosines.
51  # this transformation can be utilized when the axis are defined as follows:
52  # 1. Angles given in colatitude and longitude.
53  # 2. Colatitude measure with respct to axis x1.
54  # 3. The  longitude is measured clockwise from x2 that coincides with north
55  # See Kasarapu 2015
56  # Input: input array of planes expressed as dip direction and dip in
```

```
57 # degrees
58 # Output: number of planes, vector wih coordinates direction
59 # cosines, orientation matrix.
60 # Ref. Fisher, Lewis and Embleton, 1987
61
62 def direction_cosines(in_array):
63     n = np.ma.size(in_array, axis=0)  # number of measured planes
64     cos_x1 = np.cos(np.radians(in_array[:, 1]))
65     cos_x2 = (np.cos(np.radians(in_array[:, 0])) *
66                np.sin(np.radians(in_array[:, 1])))
67     cos_x3 = (np.sin(np.radians(in_array[:, 0])) *
68                np.sin(np.radians(in_array[:, 1])))
69     orientation_matrix = (np.sum(cos_x1 ** 2), np.sum(cos_x1 * cos_x2),
70                           np.sum(cos_x1 * cos_x3), np.sum(cos_x1 * cos_x2),
71                           np.sum(cos_x2 ** 2),
72                           np.sum(cos_x2 * cos_x3), np.sum(cos_x1 * cos_x3),
73                           np.sum(cos_x2 * cos_x3),
74                           np.sum(cos_x3 ** 2))
75     orientation_matrix = np.asarray(orientation_matrix).reshape(3, 3)
76     #print("This is the orientation matrix", orientation_matrix)
77     return n, cos_x1, cos_x2, cos_x3, orientation_matrix
78
79 # Function to compute orientation matrix, when
80 # direction cosines are already known
81 # Useful for Mixture fitting
82
83
84 def orientation_matrix(cos_x1, cos_x2, cos_x3):
85     orientation_matrix = (np.sum(cos_x1 ** 2), np.sum(cos_x1 * cos_x2),
86                           np.sum(cos_x1 * cos_x3), np.sum(cos_x1 * cos_x2),
87                           np.sum(cos_x2 ** 2),
88                           np.sum(cos_x2 * cos_x3), np.sum(cos_x1 * cos_x3),
89                           np.sum(cos_x2 * cos_x3),
90                           np.sum(cos_x3 ** 2))
91     orientation_matrix = np.asarray(orientation_matrix).reshape(3, 3)
92     return orientation_matrix
93
94 # Function to aggregate direction_cosines into one list
95 # It is useful for Mixture Model
96
97
98 def cosines_matrix (cos_x1, cos_x2, cos_x3):
99     cos_mat = [cos_x1, cos_x2, cos_x3]
100     return cos_mat
101
102
103 # Function to compute the mean direction in dip and dip directions
104 # Input: direction cosines
105 # Output: Resultant length R, mean resultant length, mean unit vector,
106 # mean dip direction and dip in degrees
107
108 def sample_mean(cos_x1, cos_x2, cos_x3):
109     n = np.ma.size(cos_x1, axis=0)
110     sum_x1 = np.sum(cos_x1)
111     sum_x2 = np.sum(cos_x2)
112     sum_x3 = np.sum(cos_x3)
```

```
113      resultant = np.sqrt(sum_x1 ** 2 + sum_x2 ** 2 + sum_x3 ** 2)
114      mean_resultant = resultant / n
115      mean_unit_vector = (sum_x1 / resultant, sum_x2 /
116                          resultant, sum_x3 /
117                          resultant)
118
119      if mean_unit_vector[1] >= 0 and mean_unit_vector[2] >= 0:
120          Q = 0
121      else:
122          if mean_unit_vector[1] < 0 and mean_unit_vector[2] >= 0:
123              Q = (-2 * np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
124                                              mean_unit_vector[
125                                              1]))) + 180)
126          else:
127              if mean_unit_vector[1] >= 0 and mean_unit_vector[2] < 0:
128                  Q = (-2 * np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
129                                              mean_unit_vector[
130                                              1]))) + 360)
131              else:
132                  Q = 180
133
134      mean_longitude = (np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
135                                              mean_unit_vector[
136                                              1]))) + Q)
137      mean_collatitude = (np.degrees(np.arccos(mean_unit_vector[0])))
138
139      return (resultant, mean_resultant, mean_unit_vector, mean_longitude,
140              mean_collatitude)
141
142
143 # Function to convert the intersection line, into trend
144 # and plunge
145 # Input: Intersection line direction cosines
146 # Output: Intersection line trend and plundge
147
148 def rectangular_to_cilindrical2(x1, x2, x3):
149      if x2 >= 0 and x3 >= 0:
150          Q = 0
151      else:
152          if x2 < 0 and x3 >= 0:
153              Q = (-2 * np.degrees(np.arctan(np.abs(x3 /
154                                              x2))) + 180)
155          else:
156              if x2 >= 0 and x3 < 0:
157                  Q = (-2 * np.degrees(np.arctan(np.abs(x3 /
158                                              x2))) + 360)
159              else:
160                  Q = 180
161
162      dip_dir = (np.degrees(np.arctan(np.abs(x3 /
163                                              x2))) + Q)
164      dip = 90-(np.degrees(np.arccos(np.abs(x1))))
165
166      return (dip_dir , dip)
167
168
```

```python
169 # Function to transfer inputs from dip direction/dip to strike and dip.
170 # It keeps the convention that dip direction is located 90 degrees
171 # counter clockwise from the strike
172
173 def ddirection_to_pole(dipdir, dip):
174     plundge = 90 - dip
175     trend= np.zeros(len(dipdir))
176     i = np.where(dipdir <= 180)
177     trend[i] = dipdir[i] + 180
178     i = np.where(dipdir > 180)
179     trend [i] = dipdir[i] - 180
180     plundge = 90 - dip
181     return trend, plundge
182
183 def ddirection_to_strike(dipdir, dip):
184     dip=dip
185     azimut = np.zeros(len(dipdir))
186     i = np.where(dipdir < 90)
187     azimut[i] = dipdir[i] + 270
188     i = np.where(dipdir >= 90)
189     azimut[i] = dipdir[i] - 90
190     return azimut, dip
191
192
193 # Function to compute the rotation matrix (transformation matrix)
194 # Input: Dip and dip direction of the new axis in degrees
195 # Output: rotation matrix
196 # Ref. Fisher, Lewis and Embleton, 1987
197
198 def rotation_matrix(ang1, ang2):
199     rotation_matrix = (np.cos(np.radians(ang2)), -np.sin(np.radians(
200         ang2)), 0, np.sin(np.radians(ang2)) * np.cos(np.radians(ang1)),
201                     np.cos(np.radians(ang1)) * np.cos(np.radians(ang2)),
202                     -np.sin(np.radians(ang1)),
203                     np.sin(np.radians(ang2)) * np.sin(np.radians(ang1)),
204                     np.sin(np.radians(ang1)) * np.cos(np.radians(ang2)),
205                     np.cos(np.radians(ang1)))
206     rotation_matrix = np.asarray(rotation_matrix).reshape(3, 3)
207     return rotation_matrix
208
209
210
211
212 if __name__ == "__main__":
213     exit
```

```
 1 '''
 2
 3
 4 Research Project:
 5 Reliability Assessment of Rock Slopes by Evidence Theory
 6 Universidad Nacional de Colombia sede Bogota
 7 Author: Rodrigo Hernandez-Carrillo
 8 2/8/18
 9
10 # Copyright (c) Rodrigo Hernandez-Carrillo 2018
11
12 _____
13
14 This script computes the Factor of Safety of a wedge
15 based on the formulation of Low 1997.
16 This is an explicit formulation
17
18 The program reads a vector, that includes the variables ordered as:
19
20 The variables of the problem are:
21
22 Joint dip and dip direction, from these the parameters beta and delta are computed
23 Joints mechanical properties:
24     1. phi1: friction angle joint 1
25     2. phi2: friction angle joint 2
26     3. c1: cohesion joint 1
27     4. c2: cohesion joint 2
28     5. uweight: unit weight
29     5. density: specific density of the rock
30 Slope geometry:
31     1. h: wedge height
32     2. betat: slope dip
33     3. alp
34     3. Omega: upper slope dip
35 Normalized Water Pressure, Gw1 and Gw2, which depen on the pressue distribution
36 assumption
37
38 '''
39
40 import numpy as np
41
42
43 # Function to compute the factor of safety.
44 # Water condition variable, if water == 0, the
45 # assumed water condition is pyramidal
46 # if water == 1, average water pressure is
47 # assigned on contact planes
48
49 # Function to compute strike
50
51 def dipdir_to_strike(angle):
52     if angle < 90:
53         strike = (angle - 90) + 360
54     else:
55         strike = angle - 90
56     return strike
```

```
 57
 58
 59 def dipdir_to_strike_sum(angle):
 60     if angle > 270:
 61         strike = (angle + 90) - 360
 62     else:
 63         strike = angle + 90
 64     return strike
 65
 66 # Function to compute the rock wedge factor of safety
 67 # Inputs: Joints dip and and dip direction, slope geometry, rock joint strength
 68 # parameters
 69 # Output: wedge factor of safety and mode of failure
 70
 71 def factor_of_safety(alpha_1, beta_1, alpha_2, beta_2, alphat, betat, phi_1, c_1,
 72                      phi_2, c_2, h, omega, density, gammawater, water):
 73
 74     strt = dipdir_to_strike_sum(alphat)
 75
 76     # Analysis for horizontal triangle: Resorts to the fact that plane orientation
 77     # is a vector, and that the dip direction is defined 90 degrees
 78     # counter-clockwise from azimut
 79
 80     # Define vector based on azimut
 81     vector_1 = (np.sin(np.radians(alpha_1)), np.cos(np.radians(alpha_1)), 0)
 82     vector_2 = (np.sin(np.radians(alpha_2)), np.cos(np.radians(alpha_2)), 0)
 83
 84     # Cross prodcunt to order 'vector 1' and 'vector 2'
 85     cross_1_2 = np.cross(vector_1, vector_2)
 86     print(cross_1_2)
 87
 88     # Reaasign order of vector
 89     if cross_1_2[2] > 0:
 90         alpha1 = alpha_1
 91         beta1 = beta_1
 92         alpha2 = alpha_2
 93         beta2 = beta_2
 94         phi1 = phi_1
 95         phi2 = phi_2
 96         c1 = c_1
 97         c2 = c_2
 98
 99
100     else:
101         alpha1 = alpha_2
102         beta1 = beta_2
103         alpha2 = alpha_1
104         beta2 = beta_1
105         phi1 = phi_2
106         phi2 = phi_1
107         c1 = c_2
108         c2 = c_1
109
110     str1 = dipdir_to_strike(alpha1)
111     str2 = dipdir_to_strike_sum(alpha2)
112
```

```
113      # str1 = 145
114      # str2 = 195
115
116      print('str1', str1)
117      print('str2', str2)
118      print('strt', strt)
119
120      # Check conditions to define horizontal triangle
121
122      angle1 = 0.0001
123      angle2 = 0.0001
124      delta1 = 0.0001
125      delta2 = 0.0001
126
127      if 180 <= str1 < 360:
128          if 180 <= str2 <= 360:
129              if (str2 <= strt < 0) or (0 <= strt < str1 - 180):
130                  if (str2 <= strt < 0):
131                      angle1 = str1 - (strt - 180)
132                      angle2 = strt - str2
133                      delta1 = beta1
134                      delta2 = beta2
135                  else:
136                      angle1 = (str1 - 180) - strt
137                      angle2 = strt - str2 + 360
138                      delta1 = beta1
139                      delta2 = beta2
140              if str1 - 180 <= strt <= str2 - 180:
141                  angle1 = strt + 180 - str1
142                  angle2 = str2 - (strt + 180)
143                  delta1 = beta1
144                  delta2 = 180 - beta2
145              if str1 <= strt < str2:
146                  angle1 = strt - str1
147                  angle2 = str2 - strt
148                  delta1 = 180 - beta1
149                  delta2 = beta2
150          if 0 <= str2 < 180:
151              if str2 <= strt < str1 - 180:
152                  angle1 = (str1 - 180) - strt
153                  angle2 = strt - str2
154                  delta1 = beta1
155                  delta2 = beta2
156              if str1 - 180 <= strt < 180 or 180 <= strt < str2 + 180:
157                  if 180 <= strt < str2 + 180:
158                      angle1 = strt - (str1 - 180)
159                      angle2 = str2 + 180 - strt
160                      delta1 = beta1
161                      delta2 = 180 - beta2
162                  else:
163                      angle1 = (strt + 180) - str1
164                      angle2 = (str2 + 180) - strt
165                      delta1 = beta1
166                      delta2 = 180 - beta2
167              if (0 <= strt < str2) or (str1 <= strt <= 360):
168                  if 0 <= strt < str2:
```

```
169                         angle1 = strt - str1 + 360
170                         angle2 = str2 - strt
171                         delta1 = 180 - beta1
172                         delta2 = beta2
173                     else:
174                         angle1 = strt - str1
175                         angle2 = str2 + 360 - strt
176                         delta1 = 180 - beta1
177                         delta2 = beta2
178
179         if 0 <= str1 < 180:
180             if 0 <= str2 < 180:
181                 if str2 <= strt < str1 + 180:
182                     angle1 = (str1 + 180) - strt
183                     angle2 = strt - str2
184                     delta1 = beta1
185                     delta2 = beta2
186                 if str1 + 180 <= strt < str1 + 180:
187                     angle1 = strt - (str1 + 180)
188                     angle2 = (str2 + 180) - strt
189                     delta1 = beta1
190                     delta2 = 180 - beta2
191                 if str1 <= strt < str2:
192                     angle1 = strt - str1
193                     angle2 = str2 - strt
194                     delta1 = 180 - beta1
195                     delta2 = beta2
196             if 180 <= str2 < 360:
197                 if str2 <= strt < str1 + 180:
198                     angle1 = (str1 + 180) - strt
199                     angle2 = strt - str2
200                     delta1 = beta1
201                     delta2 = beta2
202                 if str1 + 180 <= strt <= 360 or 0 <= strt <= str2 - 180:
203                     if str1 + 180 <= strt < 360:
204                         angle1 = strt - (str1 + 180)
205                         angle2 = (str2 - 180) - strt
206                         delta1 = beta1
207                         delta2 = 180 - beta2
208                     else:
209                         angle1 = strt - str1 + 180
210                         angle2 = (str2 - 180) - strt
211                         delta1 = beta1
212                         delta2 = 180 - beta2
213                 if str1 <= strt < str2:
214                     angle1 = strt - str1
215                     angle2 = str2 - strt
216                     delta1 = 180 - beta1
217                     delta2 = beta2+0.01
218
219         print(angle1, angle2)
220         print(delta1, delta2)
221
222
223         angle1 = np.radians(angle1)
224         angle2 = np.radians(angle2)
```

```
225    delta1 = np.radians(delta1)
226    delta2 = np.radians(delta2)
227    uweight = gammawater * density
228    f1 = np.radians(phi1)
229    f2 = np.radians(phi2)
230    dipT = np.radians(betat)
231    dipupper = np.radians(omega)
232
233    # Computation of paramters of the model
234
235    sin_si = np.abs(np.sqrt(1 - np.power(
236        np.sin(delta1) * np.sin(delta2) * np.cos(angle1 + angle2) + np.cos(
237            delta1) * np.cos(delta2), 2)))
238    tan_ep = np.sin(angle1 + angle2) / (
239            np.sin(angle1) * (1 / np.tan(delta2)) + np.sin(angle2) * (
240            1 / np.tan(delta1)))
241    a0 = sin_si / (
242            np.power(np.sin(angle1 + angle2) * np.sin(delta1) * np.sin(delta2),
243                    2) * ((1 / tan_ep) - (1 / np.tan(dipT))))
244    b1 = a0 * np.sin(angle2) * np.sin(delta2)
245    b2 = a0 * np.sin(angle1) * np.sin(delta1)
246    a1 = (np.sin(delta2) * (1 / np.tan(delta1)) - np.cos(delta2) * np.cos(
247        angle1 + angle2)) / (sin_si * np.sin(angle1 + angle2))
248    a2 = (np.sin(delta1) * (1 / np.tan(delta2)) - np.cos(delta1) * np.cos(
249        angle1 + angle2)) / (sin_si * np.sin(angle1 + angle2))
250    k = (1 - np.tan(dipupper) / np.tan(dipT)) / (1 - np.tan(dipupper) / tan_ep)
251    Z = np.cos(delta1) * np.cos(delta2) + np.sin(delta1) * np.sin(delta2) * np.cos
252        angle1 + angle2)
253
254    # Water conditions
255
256    if water == 0:
257        Gw1 = 0.5 * k  # Pyramidal water pressure distribution
258        Gw2 = 0.5 * k  # Pyramidal water pressure distribution
259    else:
260        u = gammawater * h * k / 6
261        Gw1 = 3 * u / (gammawater * h)
262        Gw2 = 3 * u / (gammawater * h)
263
264    # Computation of FoS depending on the sliding surface. cte1 and cte2 has to be
265    #   checked
266
267    cte1 = (a1 - b1 * Gw1 / density)
268    cte2 = (a2 - b2 * Gw2 / density)
269    plane1 = cte1 + Z * cte2
270    plane2 = cte2 + Z * cte1
271
272    if cte1 >= 0 and cte2 >= 0:
273        FS = cte1 * np.tan(f1) + cte2 * np.tan(f2) + 3 * b1 * c1 / (
274                uweight * h) + 3 * b2 * c2 / (uweight * h)
275        failure = "Failure along intersection line"
276    else:
277        if cte2 < 0 and plane1 > 0:
278            FS = (plane1 * np.tan(f1) + 3 * b1 * c1 / (uweight * h)) / np.sqrt(
279                    1 + (((b2 * Gw2 / density) - a2) * sin_si) ** 2)
280            failure = "Failure along Plane 1"
```

```
281         else:
282             if cte1 < 0 and plane2 > 0:
283                 FS = ((plane2) * np.tan(f2) + 3 * b2 * c2 / (uweight * h)) / np.sq
284                     1 + (((b1 * Gw1 / density) - a1) * sin_si) ** 2)
285             failure = "Failure along Plane 2"
286         else:
287             if plane1 < 0 and plane2 < 0:
288                 FS = 0
289                 failure = "Contact lost on both planes"
290
291
292     return FS, failure
293
294 if __name__ == "__main__":
295
```

```
 1 '''
 2
 3 Research Project:
 4 Reliability Assessment of Rock Slopes by Evidence Theory
 5 Universidad Nacional de Colombia sede Bogota
 6 Author: Rodrigo Hernandez-Carrillo
 7 2/8/18
 8
 9 # Copyright (c) Rodrigo Hernandez-Carrillo 2018
10
11 _____
12
13
14 This script defines the fucntions requiered to compute the parameters of a Kent
15 distribution.
16
17 As input it requires the rotation and orientation matrices.
18
19 Procedure followed as described by Fisher, Lewis and Embleton, 1987
20 '''
21
22
23 import numpy as np
24
25
26 # Function to compute the parameters of Kent distribution, from a given
27 # set of data
28 # Input: orientation matrix, rotation matrix, number of planes and mean
29 # resultant
30 # Output: kappa: concentration parameter. Beta: Ovalness. Major and minor
31 # axis.
32 # Ref. Fisher, Lewis and Embleton, 1987. Section 5.3
33
34
35 def estimation_kent_parameters(ori_mat, rot_mat, n, mean_res):
36     S = ori_mat * 1 / n
37     B = np.dot(np.transpose(rot_mat), np.dot(S, rot_mat))
38     psi = 0.5 * np.arctan(2 * B[1, 2] / (B[1, 1] - B[2, 2]))
39     k = ((1, 0, 0, 0, np.cos(psi), -np.sin(psi), 0, np.sin(psi),
40           np.cos(psi)))
41     k = np.asarray(k).reshape(3, 3)
42     g = np.dot(rot_mat, k)
43     v = np.dot(np.transpose(g), np.dot(S, g)) # Orthogonal Axis Matrix
44     q = v[1, 1] - v[2, 2]
45     kappa = 1 / (2 - 2 * mean_res - q) + 1 / (2 - 2 * mean_res + q)
46     beta = 0.5 * (1 / (2 - 2 * mean_res - q) - 1 / (2 - 2 * mean_res +
47                                                     q))
48
49     # Limit the value of kappa to 15. Otherwise, simulation is pointless.
50
51     if n < 10 and kappa < 15:
52         beta = 0
53         kappa = 15
54
55     # beta has to be positive, otherwise, it assumed as zero, and data are
56     # modelled as Fisher distribution with concetration kappa
```

```
57      if beta >= 0:
58          beta = beta
59      else:
60          beta = 0
61
62      return g, kappa, beta, v
63
64 # Function to estimate the normalization constant for large
65 # values of kappa. Useful for mixtures
66
67 def estimation_ck (kappa, beta):
68      ck = np.exp(-kappa) * np.sqrt(kappa*kappa -4*beta*beta) / 2*np.pi
69      return ck
70
71 # This function simulates a Kent distribution by a Acceptance /
72 # Rejection algorithm. A bilateral exponential distribution is used
73 # as bounding distribution
74
75 def simulation_kent(kappa, beta, n=1000):
76      lamda = 2 * beta / kappa
77      a = 4 * kappa - 8 * beta
78      b = 4 * kappa + 8 * beta
79      gamma = 8 * beta
80      c1 = 1
81      lamda1 = np.sqrt((a + 2 * np.sqrt(gamma)))
82      c2 = b / (2 * (b - gamma))
83      lamda2 = np.sqrt(b)
84
85      # solve cubic equations and maximize h(x) = f(x)/t*g(x)
86
87      r1 = [-2 * gamma, 0, -a, lamda1]
88      t1c = np.roots(r1)
89      root1 = np.abs(-2 * gamma * np.power(t1c, 3) - a * t1c + lamda1)
90      t1c = t1c[np.argmin(root1)]
91      htc1 = np.exp(-0.5*(a*t1c**2 + gamma*t1c**4) + lamda1*np.abs(t1c))
92      r2 = [2 * gamma, 0, -b, lamda2]
93      t2c = np.roots(r2)
94      root2 = np.abs(2 * gamma * np.power(t2c, 3) - b * t2c + lamda2)
95      t2c = t2c[np.argmin(root2)]
96      htc2 = np.exp(-0.5*(b*t2c**2 - gamma*t2c**4) + lamda2*np.abs(t2c))
97      sim_t1 = list()  # Simulation vector for t1
98      sim_t2 = list()  # Simulation vector for t2
99      stop = 0
100
101     while stop < n:
102
103         u1 = np.random.random()
104         u2 = np.random.random()
105
106         t1 = np.log(np.absolute(u1)) / -lamda1
107         t2 = np.log(np.absolute(u2)) / -lamda2
108
109         ft1 = np.exp(
110             -0.5 * (a * np.power(t1, 2) + gamma * np.power(t1, 4)))
111         ft2 = np.exp(
112             -0.5 * (b * np.power(t2, 2) - gamma * np.power(t2, 4)))
```

```
113
114          gt1 = np.exp(-lamda1 * np.absolute(t1))
115          gt2 = np.exp(-lamda2 * np.absolute(t2))
116
117          ht1 = ft1 / (htc1 * gt1)
118          ht2 = ft2 / (htc2 * gt2)
119
120          u1h = np.random.random()
121          u2h = np.random.random()
122
123          if u1h <= ht1 and u2h <= ht2 and t1**2 + t2**2 < 1:
124              u1abs = np.random.random()  ## Random number to account for
125              # the bilateral wrapping (absolute value)
126              u2abs = np.random.random()
127              if u1abs >= 0.5:
128                  sim_t1.append(t1)
129              else:
130                  sim_t1.append(-t1)
131              if u2abs >= 0.5:
132                  sim_t2.append(t2)
133              else:
134                  sim_t2.append(-t2)
135              stop = len(sim_t1)
136
137      sim_t1 = np.asarray(sim_t1)
138      sim_t2 = np.asarray(sim_t2)
139      return sim_t1, sim_t2
140
141
142 def rot_kent(rot, x1, x2, x3):
143      cos_matrix =np.vstack((x1, x2, x3)).T
144      rot_cos = np.einsum("jk, ik -> ij", rot, cos_matrix)
145      return rot_cos
146
147 # Routine lo align the major axis of the Kent simulation with the E-W axes.
148 # Rotate 90 degrees counterclockwise
149
150 def major_axis_alignment(dipdir, dip):
151      dip=dip
152      azimut = np.zeros(len(dipdir))
153      i = np.where(dipdir > 270)
154      azimut[i] = dipdir[i] - 270
155      i = np.where(dipdir <= 270)
156      azimut[i] = dipdir[i] + 90
157      return azimut, dip
158
159
160 '''
161      sim = np.array(sim_vector).reshape(n, 2)
162      sim = sim[:, np.arctan(sim[:, 1] / sim[:, 0])]
163      sim = sim[ 2*np.arcsin(sim[0, :]/np.cos(sim[:, 1])), :]
164      sim_kent = np.deegres(sim)
165      '''
166
167 if __name__ == "__main__":
168      exit
```

```
 1  '''
 2
 3  Research Project:
 4  Reliability Assessment of Rock Slopes by Evidence Theory
 5  Universidad Nacional de Colombia sede Bogota
 6  Author: Rodrigo Hernandez-Carrillo
 7  2/12/18
 8
 9  # Copyright (c) Rodrigo Hernandez-Carrillo 2018
10
11  _____
12
13
14  This script resorts to the KentDistribution script to  computes the input random
15  sets corresponding to the structural information collected. Considers:
16
17  1. Parameters computation according to Kent distribution
18
19  '''
20
21  import numpy as np
22  import KentDistribution1 as kd
23  import VectorMean as vm
24  import matplotlib.pyplot as plt
25  import mplstereonet
26
27  if __name__ == "__main__":
28
29      # Computation of Kent parameters. Looping as many times as pieces of informatio
30      # are analysed
31
32      n_pieces_evidence = 58
33
34      statistics_summary = list()  # List to save parameters result
35      for i in range(n_pieces_evidence):
36
37          dipdir_dip = vm.open_array('InputSet', 'csv', i)
38          dipdir_dip = vm.angle_adjustment(dipdir_dip)
39          n, x_cos, y_cos, z_cos, orientation = vm.direction_cosines(dipdir_dip)
40          r_length, r_mean, mean_vector, dipdir_mean, dip_mean = (vm.sample_mean(x_co
41                                                                  y_cos, z_cos))
42          print(dip_mean)
43          rotation = vm.rotation_matrix(dipdir_mean, dip_mean)
44          kent_matrix, k, beta, v = \
45              kd.estimation_kent_parameters(orientation, rotation, n, r_mean)
46          kappa_fisher = (n) / (n - r_length)
47          print("kappa =", k, "beta=", beta, "mean",
48                dipdir_mean, "/", dip_mean)
49
50          # Simulation of Kent distribution.
51
52          n_sim = 10000
53          sim_t1, sim_t2 = kd.simulation_kent(k, beta, n_sim)
54          sim_longitude, sim_collatitude = (vm.rectangular_to_cilindrical(
55              sim_t1, sim_t2))
56          sim_longitude, sim_collatitude = (kd.major_axis_alignment(sim_longitude,
```

```
 57                                                                   sim_collatitude)
 58         sim_centered = (np.vstack((sim_longitude,
 59                                     sim_collatitude))).T
 60
 61         # Plotting data about the origin
 62
 63         fig = plt.figure()
 64         ax = fig.add_subplot(111, projection='stereonet')
 65         ax.pole(sim_longitude, sim_collatitude, 'g^', markersize=1)
 66         ax.grid()
 67
 68         # Rotation of simulated data about the mean
 69
 70         n, x1, x2, x3, rot = vm.direction_cosines(sim_centered)
 71         rotation = vm.rotation_matrix(dipdir_mean, 90 - dip_mean)
 72         rotated_data = kd.rot_kent(rotation, x1, x2, x3)
 73         dipdir, dip = (vm.rectangular_to_cilindrical1(rotated_data[:, 0],
 74                                                        rotated_data[:, 1],
 75                                                        rotated_data[:, 2]))
 76
 77         # Check if plane has a dip angle higher than 90, then it is adjusted and
 78         # dip direction is rotated 180
 79
 80         for j in range(n_sim):
 81             if dip[j] >= 90:
 82                 dip[j] = 180 - dip[j]
 83                 if dipdir[j] <= 180:
 84                     dipdir[j] = dipdir[j] + 180
 85                 else:
 86                     dipdir[j] = dipdir[j] - 180
 87
 88         # Data are transformed to azimut dip, to be plotted using the package
 89         # mplstereonet
 90
 91         kent_azimut, kent_dip = vm.ddirection_to_strike(dipdir, dip)
 92
 93         # Data are concatenated in an array and then save it in a csv file
 94         # KentSimulationi.csv. This one file, per source of info.
 95
 96         sim_rotated = (np.vstack((dipdir, dip))).T
 97         np.savetxt('{}{}{}{}'.format("JointKentSimulation", i, '.', "csv"),
 98                     sim_rotated,
 99                     delimiter=",", header="Dip direction, Dip", comments='')
100
101         # Plotting rotated planes
102
103         fig1 = plt.figure()
104         ax1 = fig1.add_subplot(111, projection='stereonet')
105         ax1.pole(kent_azimut, kent_dip, 'b^', markersize=1)
106         ax1.grid()
107         result = (i, dipdir_mean, 90 - dip_mean, k, beta)
108         statistics_summary.append(result)
109
110     np.savetxt("SphericalStatistics.csv", statistics_summary, delimiter=",",
111                 header="ID, Mean Dip direction, Mean Dip, kappa, beta", comments=''
112
```

```
 1 '''
 2
 3 Research Project:
 4 Reliability Assessment of Rock Slopes by Evidence Theory
 5 Universidad Nacional de Colombia sede Bogota
 6 Author: Rodrigo Hernandez-Carrillo
 7 2/12/18
 8
 9 # Copyright (c) Rodrigo Hernandez-Carrillo 2018
10
11 _____
12
13 Script to compute:
14 Geomechanical params expressed as random sets
15 Joint planes. Measured joint planes are combined
16 Slope modelled as Kent, with several scenarios (Different values of kappa and beta)
17 For illustration purposes, the slope was
18 considere as deterministic, but the script allows to model it as random variable
19
20 Sequence is as follows:
21
22 For each piece of evidence on geomechanical parameters (usually less info)
23
24 1. Random set input are read.
25
26 2. Combinations of focal elements and probability assignments by using itertools
27
28 Start loop for each piece of evidence on joint sets
29
30 3. Read joint plane input data (separately for each plane)
31
32 4. Define combinations of joint planes
33
34 5. Define lists for storing FoS and start counters
35
36 For each scenario defined for slope plane as Kent
37
38 6. Read the current slope simulated planes
39
40 Loop for each combination of random sets
41
42 Loop for each realization (total number of realizations is controlled by the
43 total amount of combinations of input planes)
44
45 7. Compute the probability assignment of the combination of inputs
46
47 8. Compute deterministically the factor of safety
48
49 9. Define the number of potentially unstable blocks, the weitght the probability of
50 each FOS, by the current probability assignment
51
52 10. Store the factor of safety, step-wise, for all the combination of random sets,
53 of the current scenario of slope simualtions.
54
55 11. This defines the random set for the reponse, for that specific slope scenario.
56
```

```
 57  12. Organize information according to random sets procedure (class RandomSets)
 58
 59  '''
 60
 61  import WedgeFoS as fs
 62  import numpy as np
 63  import RandomSets as rs
 64  import VectorMean as vm
 65  import matplotlib.pyplot as plt
 66  import mplstereonet
 67  import Combinations1 as cb
 68  import time
 69  import itertools as it
 70
 71
 72  if __name__ == "__main__":
 73
 74      # Variable to measure execution time
 75      start = time.time()
 76      print("I am working, there are many computations in progress")
 77
 78      # Read values leading to upper FoS (not upper params)
 79
 80      up_dipdir_s = vm.open_array('Upper', 'csv', 'SlopeDipDirection')
 81      up_dipdir_s = np.asarray(up_dipdir_s)
 82      up_dip_s = vm.open_array('Upper', 'csv', 'SlopeDip')
 83      up_phi_1 = vm.open_array('Upper', 'csv', 'Friction')
 84      up_c_1 = vm.open_array('Upper', 'csv', 'Cohesion')
 85      up_phi_2 = vm.open_array('Upper', 'csv', 'Friction')
 86      up_c_2 = vm.open_array('Upper', 'csv', 'Cohesion')
 87      up_h = vm.open_array('Upper', 'csv', 'Height')
 88      up_omega = vm.open_array('Upper', 'csv', 'Omega')
 89      up_spec_grav = vm.open_array('Upper', 'csv', 'UnitWeight')
 90
 91      # Read values leading to lower FoS (not lower params)
 92
 93      low_dipdir_s = vm.open_array('Lower', 'csv', 'SlopeDipDirection')
 94      low_dip_s = vm.open_array('Lower', 'csv', 'SlopeDip')
 95      low_phi_1 = vm.open_array('Lower', 'csv', 'Friction')
 96      low_c_1 = vm.open_array('Lower', 'csv', 'Cohesion')
 97      low_phi_2 = vm.open_array('Lower', 'csv', 'Friction')
 98      low_c_2 = vm.open_array('Lower', 'csv', 'Cohesion')
 99      low_h = vm.open_array('Lower', 'csv', 'Height')
100      low_omega = vm.open_array('Lower', 'csv', 'Omega')
101      low_spec_grav = vm.open_array('Lower', 'csv', 'UnitWeight')
102      print((up_phi_1, up_c_1, up_phi_2, up_c_2))
103      print((low_phi_1, low_c_1, low_phi_2, low_c_2))
104
105
106      # Read probability assignment
107
108      pa_phi_1 = vm.open_array('PA', 'csv', 'Friction')
109      pa_phi_2 = vm.open_array('PA', 'csv', 'Friction')
110      pa_c_1 = vm.open_array('PA', 'csv', 'Cohesion')
111      pa_c_2 = vm.open_array('PA', 'csv', 'Cohesion')
112
```

```
113     # Generate combinations for analysis, if required
114
115     upper_inputs = np.asarray(list(it.product(up_phi_1, up_c_1, up_phi_2, up_c_2))
116     lower_inputs = np.asarray(
117         list(it.product(low_phi_1, low_c_1, low_phi_2, low_c_2)))
118
119     print('lower inputs', lower_inputs)
120
121     # Generate combination of probability assignments
122
123     pa_comb = np.asarray(list(it.product(pa_phi_1, pa_c_1, pa_phi_2, pa_c_2)))
124     n_focal_elements =  np.size(lower_inputs[:, 0])
125     print("Number of focal elements", n_focal_elements)
126
127     target_FOS = 1.0  # Allowable FS
128     n_pieces_evidence = 16  # Number of pieces of evidence to compute FOS
129
130     for i in range(n_pieces_evidence):
131
132         # Read joint planes, expressed as dip and dip_direction
133
134         set1 = vm.open_array('StochasticInputSet', 'csv', 2 * i)
135         set2 = vm.open_array('StochasticInputSet', 'csv', 2 * i + 1)
136
137         # Define combinations of planes
138
139         dipdir_dip = cb.stochastic_joints(set1, set2)
140
141         # Number of sim, depends on the number of combinations of planes
142         n_sim = len(dipdir_dip[:, 0])
143
144         up_n_stable = 0  # Counter for stable blocks
145         up_n_failed = 0  # Counter for block with FOS lower than target
146         up_n_blocks = 0  # Counter for potential blocks
147         low_n_stable = 0  # Counter for stable blocks
148         low_n_failed = 0  # Counter for block with FOS lower than target
149         low_n_blocks = 0  # Counter for potential blocks
150
151         # Lists to store FoS and mode of failure
152
153         up_FOS = list()
154         up_mode_failure = list()
155         low_FOS = list()
156         low_mode_failure = list()
157         PA_FoS = list()
158         RST_Area = list()  # Area between bounds
159         upper_PoF = list()
160         lower_PoF = list()
161
162         # Slope modeled as Kent, with several scenarios (different kappa, beta)
163
164         n_scenarios = 1
165
166         for l in range(n_scenarios):
167
168             #in_slope_params = vm.open_array('SlopeSimulation', 'csv', l)
```

```
169
170            for ii in range(n_focal_elements):
171
172                for j in range(n_sim):
173
174                    # Probability assignment
175
176                    pa_FoS = (pa_comb[ii,0] * pa_comb[ii,1] * pa_comb[ii,2] *
177                              pa_comb[ii, 3])
178
179                    # FOR UPPER BOUND
180
181                    # Simulation of inputs according to assumed probability
182                    # distribution
183
184                    #dipdir_s = in_slope_params[j, 0]
185                    dipdir_s = 330
186                    #dip_s = in_slope_params[j, 1]
187                    dip_s = 70
188                    phi_1 = upper_inputs[ii, 0]
189                    #phi_1 = 32
190                    c_1 = upper_inputs[ii, 1]
191                    phi_2 = upper_inputs[ii, 0]
192                    # phi_2 = 32
193                    c_2 = upper_inputs[ii, 1]
194                    # h = up_h
195                    h = 15
196                    u_water = 9.81
197                    spec_gravity = up_spec_grav
198
199                    # Computation of intersection
200                    # Based on the cross product of normal vectors
201
202                    planes = (np.asarray([[dipdir_dip[j, 0], dipdir_dip[j, 1]],
203                                          [dipdir_dip[j, 2], dipdir_dip[j, 3]],
204                                          [dipdir_s, dip_s]]))
205                    # print(planes)
206                    n, x_cos, y_cos, z_cos, orientation = vm.direction_cosines(
207                        planes)
208                    vector1_cos = [x_cos[0], y_cos[0], z_cos[0]]
209                    vector2_cos = [x_cos[1], y_cos[1], z_cos[1]]
210                    slope = [x_cos[2], y_cos[2], z_cos[2]]
211
212                    # Intersection line plunge and trend is computed with cross
213                    # product.
214                    # Verification of the relative position of planes 1 and 2
215                    # Component X1 is checked. If positive, the cross product is
216                    # corrected (commuted)
217
218                    intersection = np.cross(vector1_cos, vector2_cos)
219                    if intersection[0] <= 0:
220                        intersection = intersection
221                    else:
222                        intersection = np.cross(vector2_cos, vector1_cos)
223
224                    x1i = intersection[0]
```

```
225                         x2i = intersection[1]
226                         x3i = intersection[2]
227                         inter_dipdir, inter_dip = vm.rectangular_to_cilindrical2(x1i,
228                                                                       x2i,
229                                                                       x3i)
230
231                         # Before computing the factor of safety, the kinematic conditi
232                         # for the wedge to form are checked.
233                         # Conditions are:
234                         # Angle between intersection trend and slope dip direction
235                         # has to be less than 90 degrees
236                         # Intersection plunge less than apparent slope dip
237
238                         # Conditional to calculate the angle between
239                         # intersection and slope
240
241                         if np.abs(inter_dipdir - dipdir_s) < 180:
242                             angle_inter_slope = np.abs(inter_dipdir - dipdir_s)
243                         else:
244                             angle_inter_slope = 360 - np.abs(inter_dipdir - dipdir_s)
245
246                         aparent_slope_dip = (
247                             np.degrees((np.arctan(np.tan(np.radians(dip_s))
248                                             * np.cos(
249                                 np.radians(angle_inter_slope))))))
250
251                         if (angle_inter_slope <= 90) and (inter_dip < aparent_slope_di
252                             up_FoS, up_failure = (fs.factor_of_safety(dipdir_dip[j, 0]
253                                                                       dipdir_dip[j, 1]
254                                                                       dipdir_dip[j, 2]
255                                                                       dipdir_dip[j, 3]
256                                                                       dipdir_s,
257                                                                       dip_s,
258                                                                       phi_1, c_1, phi_
259                                                                       c_2, h,
260                                                                       0, spec_gravity,
261                                                                       u_water, 0))
262                             if up_FoS >= 0:
263                                 up_FoS = up_FoS
264                                 up_FOS.append(up_FoS)
265                                 up_mode_failure.append(up_failure)
266                                 up_n_blocks = up_n_blocks + 1
267                             else:
268                                 up_FoS = np.abs(up_FoS)
269                                 up_FOS.append(up_FoS)
270                                 up_n_blocks = up_n_blocks + 1
271
272                             # Assess blocks with FOS lower than target
273                             if up_FoS < target_FOS:
274                                 up_n_failed = up_n_failed + 1
275
276                         else:
277                             up_n_stable = up_n_stable + 1
278                             # FoS = 100000
279                             # FOS.append(FoS)
280                             up_failure = "There is no failure"
```

```
281                             up_mode_failure.append(up_failure)
282
283                     # FOR THE LOWER BOUND
284
285                     # Input lower bound
286
287                     phi_1 = lower_inputs[ii, 0]
288                     c_1 = lower_inputs[ii, 1]
289                     phi_2 = lower_inputs[ii, 0]
290                     # phi_2 = 32
291                     c_2 = lower_inputs[ii, 1]
292                     # h = low_h
293                     h = 15
294                     u_water = 9.81
295                     spec_gravity = low_spec_grav
296
297                     # FOS computation
298
299                     if (angle_inter_slope <= 90) and (inter_dip < aparent_slope_di
300                         low_FoS, low_failure = (fs.factor_of_safety(dipdir_dip[j,
301                                                                     dipdir_dip[j,
302                                                                     dipdir_dip[j,
303                                                                     dipdir_dip[j,
304                                                                     dipdir_s,
305                                                                     dip_s, phi_1,
306                                                                     c_1, phi_2, c_
307                                                                     h, 0,
308                                                                     spec_gravity,
309                                                                     u_water, 0))
310                         if low_FoS >= 0:
311                             low_FoS = low_FoS
312                             low_FOS.append(low_FoS)
313                             low_mode_failure.append(low_failure)
314                             low_n_blocks = low_n_blocks + 1
315                         else:
316                             low_FoS = np.abs(low_FoS)
317                             low_FOS.append(low_FoS)
318                             low_n_blocks = low_n_blocks + 1
319
320
321
322                          #Assess blocks with FOS lower than target
323                         if low_FoS < target_FOS:
324                             low_n_failed = low_n_failed + 1
325
326
327                     else:
328                         low_n_stable = low_n_stable + 1
329                         low_failure = "There is no failure"
330                         low_mode_failure.append(low_failure)
331
332             # Probabilities
333             # Probability of each realization
334             up_prob = pa_FoS*(1 / up_n_blocks)
335
336             # lists to stores weighted FOS
```

```
337
338                    for jj in range(up_n_blocks):
339                        PA_FoS.append(up_prob)
340
341                    # Restart counters
342
343                    up_n_blocks = 0
344                    up_prob = 0
345
346            low = np.asarray(low_FOS)
347            up = np.asarray(up_FOS)
348            pa = np.asarray(PA_FoS)
349
350            DS_FOS = np.zeros((np.size(low), 3))
351            DS_FOS[:, 0] = low[:]
352            DS_FOS[:, 1] = up[:]
353            DS_FOS[:, 2] = pa[:]
354
355            lower_FOS, lower_prob, upper_FOS, upper_prob, A = rs.random_sets(DS_FO
356
357            # Finding the probability of failure
358
359            x = np.where(lower_FOS >= target_FOS)
360            lower_FOS_PoF = lower_prob[np.min(x)]
361
362            x = np.where(upper_FOS >= target_FOS)
363            upper_FOS_PoF = upper_prob[np.min(x)]
364
365            lower_PoF.append(lower_FOS_PoF)
366            upper_PoF.append(upper_FOS_PoF)
367
368            RST_Area.append(A)
369
370            print("Area between bounds is: ", A, "for scenario", l,
371                  "for source,", i)
372            print("Lower POF is", lower_FOS_PoF, "for scenario", l,
373                  "for source,", i)
374            print("Upper POF is", upper_FOS_PoF  , "for scenario", l,
375                  "for source,", i)
376
377            # Finding the probability of having a FoS lower than 1, 2, 3, 4, 5
378            All_FOS = 10    # FOS to consider
379
380            #####################
381            ##  LOWER BOUND #####
382            #################33####
383
384            lower_FOS_AllPoF = list()  # List to storage FoS lower than integer
385            for h in range(All_FOS):
386                y = np.where(lower_FOS >= h+1)
387                lower_FOS_All = lower_prob[np.min(y)]
388                lower_FOS_AllPoF.append(lower_FOS_All)
389
390            np.savetxt(
391                '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerALLPoF",
392                                          "Scenarios", l,
```

```
393                                             "Source", i, '.', "csv"), lower_FOS_AllP
394                 delimiter=",", header="PoF", comments='')
395
396             lower_FOS_AllPoF = list()  # Re start list
397
398             ######################
399             ##  UPPER BOUND  #####
400             ###############33####
401
402             upper_FOS_AllPoF = list()  # List to storage FoS lower than integer
403             for h in range(All_FOS):
404                 y = np.where(upper_FOS >= h + 1)
405                 upper_FOS_All = upper_prob[np.min(y)]
406                 upper_FOS_AllPoF.append(upper_FOS_All)
407
408             np.savetxt(
409                 '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperALLPoF",
410                                          "Scenarios", l,
411                                          "Source", i, '.', "csv"), upper_FOS_AllP
412                 delimiter=",", header="PoF", comments='')
413
414             upper_FOS_AllPoF = list()  # Re start lis
415
416             np.savetxt(
417                 '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerFOSplot",
418                                          "Scenarios", l,
419                                          "Source", i, '.', "csv"), lower_FOS[1:],
420                 delimiter=",", header="Area", comments='')
421
422             np.savetxt(
423                 '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerFOSProb",
424                                          "Scenarios", l,
425                                          "Source", i, '.', "csv"), lower_prob[:-1
426                 delimiter=",", header="Area", comments='')
427
428             np.savetxt(
429                 '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperFOSplot",
430                                          "Scenarios", l,
431                                          "Source", i, '.', "csv"), upper_FOS[1:],
432                 delimiter=",", header="Area", comments='')
433
434             np.savetxt(
435                 '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperFOSProb",
436                                          "Scenarios", l,
437                                          "Source", i, '.', "csv"), upper_prob[:-1
438                 delimiter=",", header="Area", comments='')
439
440             # Plot
441
442             plt.figure()
443             plt.plot(lower_FOS[1:], lower_prob[:-1], 'g-', linewidth=2.0)
444             plt.plot(upper_FOS[1:], upper_prob[:-1], 'r-', linewidth=2.0)
445             plt.ylim(0, 1.01)
446             plt.xlim(0, 10)
447             plt.xlabel('Factor of Safety')
448             plt.ylabel('Cumulative Probability')
```

```
449                #plt.show()
450
451                # Re start lists to store
452
453                PA_FoS = list()
454                low_FOS = list()
455                up_FOS = list()
456                up_failed = list()
457                low_failed = list()
458
459                plt.savefig('{}{}{}{}{}{}{}{}'.format('StochasticKentRandom','Probabilit
460                                               'Function',
461                                               'Scenario', l, "Source", i))
462
463         np.savetxt(
464            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "Area", "Scenarios",
465                                  "Source", i, '.', "csv"), RST_Area,
466            delimiter=",", header="Area", comments='')
467
468         np.savetxt(
469            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerPoF",
470                                  "Scenarios",
471                                  l, "Source", i, '.', "csv"), lower_PoF,
472            delimiter=",", header="Probabilidad de falla", comments='')
473
474         np.savetxt(
475            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperPoF",
476                                  "Scenarios",
477                                  l, "Source", i, '.', "csv"), upper_PoF,
478            delimiter=",", header="Probabilidad de falla", comments='')
479
480 end = time.time()
481 time_seconds = end -start
482 time_min = time_seconds / 60
483
484 print("I am done computing the factor of safety for",n_sim, " wedges ")
485
486 print("It took me", time_seconds, "seconds or", time_min, " minutes to complete "
487        "the", n_sim*n_scenarios*n_focal_elements*n_pieces_evidence,
488      "wedges FOS computation")
```

```
 1 '''
 2
 3 Research Project:
 4 Reliability Assessment of Rock Slopes by Evidence Theory
 5 Universidad Nacional de Colombia sede Bogota
 6 Author: Rodrigo Hernandez-Carrillo
 7 2/12/18
 8
 9 # Copyright (c) Rodrigo Hernandez-Carrillo 2018
10
11 _____
12
13 Script to compute:
14 Geomechanical params expressed as random sets
15 Joint planes. Measured joint planes are combined
16 Slope modelled as Kent, with several scenarios (Different values of kappa and beta)
17 For illustration purposes, the slope was
18 considered as deterministic, but the script allows to model it as random variable
19
20 Sequence is as follows:
21
22 For each piece of evidence on geomechanical parameters (usually less info)
23
24 1. Random set input are read.
25
26 2. Combinations of focal elements and probability assignments by using itertools
27
28 Start loop for each piece of evidence on joint sets
29
30 3. Read joint plane input data (separately for each plane)
31
32 4. Define combinations of joint planes
33
34 5. Define lists for storing FoS and start counters
35
36 For each scenario defined for slope plane as Kent
37
38 6. Read the current slope simulated planes
39
40 Loop for each combination of random sets
41
42 Loop for each realization (total number of realizations is controlled by the
43 total amount of combinations of input planes)
44
45 7. Compute the probability assignment of the combination of inputs
46
47 8. Compute deterministically the factor of safety
48
49 9. Define the number of potentially unstable blocks, the weitght the probability of
50 each FOS, by the current probability assignment
51
52 10. Store the factor of safety, step-wise, for all the combination of random sets,
53 of the current scenario of slope simualtions.
54
55 11. This defines the random set for the reponse, for that specific slope scenario.
56
```

```
 57  12. Organize information according to random sets procedure (class RandomSets)
 58
 59  '''
 60
 61  import WedgeFoS as fs
 62  import numpy as np
 63  import RandomSets as rs
 64  import VectorMean as vm
 65  import matplotlib.pyplot as plt
 66  import mplstereonet
 67  import Combinations1 as cb
 68  import time
 69  import itertools as it
 70
 71
 72  if __name__ == "__main__":
 73
 74      # Variable to measure execution time
 75      start = time.time()
 76      print("I am working, there are many computations in progress")
 77
 78      # Read values leading to upper FoS (not upper params)
 79
 80      up_dipdir_s = vm.open_array('Upper', 'csv', 'SlopeDipDirection')
 81      up_dipdir_s = np.asarray(up_dipdir_s)
 82      up_dip_s = vm.open_array('Upper', 'csv', 'SlopeDip')
 83      up_phi_1 = vm.open_array('Upper', 'csv', 'Friction')
 84      up_c_1 = vm.open_array('Upper', 'csv', 'Cohesion')
 85      up_phi_2 = vm.open_array('Upper', 'csv', 'Friction')
 86      up_c_2 = vm.open_array('Upper', 'csv', 'Cohesion')
 87      up_h = vm.open_array('Upper', 'csv', 'Height')
 88      up_omega = vm.open_array('Upper', 'csv', 'Omega')
 89      up_spec_grav = vm.open_array('Upper', 'csv', 'UnitWeight')
 90
 91      # Read values leading to lower FoS (not lower params)
 92
 93      low_dipdir_s = vm.open_array('Lower', 'csv', 'SlopeDipDirection')
 94      low_dip_s = vm.open_array('Lower', 'csv', 'SlopeDip')
 95      low_phi_1 = vm.open_array('Lower', 'csv', 'Friction')
 96      low_c_1 = vm.open_array('Lower', 'csv', 'Cohesion')
 97      low_phi_2 = vm.open_array('Lower', 'csv', 'Friction')
 98      low_c_2 = vm.open_array('Lower', 'csv', 'Cohesion')
 99      low_h = vm.open_array('Lower', 'csv', 'Height')
100      low_omega = vm.open_array('Lower', 'csv', 'Omega')
101      low_spec_grav = vm.open_array('Lower', 'csv', 'UnitWeight')
102      print((up_phi_1, up_c_1, up_phi_2, up_c_2))
103      print((low_phi_1, low_c_1, low_phi_2, low_c_2))
104
105
106      # Read probability assignment
107
108      pa_phi_1 = vm.open_array('PA', 'csv', 'Friction')
109      pa_phi_2 = vm.open_array('PA', 'csv', 'Friction')
110      pa_c_1 = vm.open_array('PA', 'csv', 'Cohesion')
111      pa_c_2 = vm.open_array('PA', 'csv', 'Cohesion')
112
```

```
113      # Generate combinations for analysis, if required
114
115      upper_inputs = np.asarray(list(it.product(up_phi_1, up_c_1, up_phi_2, up_c_2))
116      lower_inputs = np.asarray(
117          list(it.product(low_phi_1, low_c_1, low_phi_2, low_c_2)))
118
119      print('lower inputs', lower_inputs)
120
121      # Generate combination of probability assignments
122
123      pa_comb = np.asarray(list(it.product(pa_phi_1, pa_c_1, pa_phi_2, pa_c_2)))
124      n_focal_elements =  np.size(lower_inputs[:, 0])
125      print("Number of focal elements", n_focal_elements)
126
127      target_FOS = 1.0  # Allowable FS
128      n_pieces_evidence = 16  # Number of pieces of evidence to compute FOS
129
130      for i in range(n_pieces_evidence):
131
132          # Read joint planes, expressed as dip and dip_direction
133
134          set1 = vm.open_array('StochasticInputSet', 'csv', 2 * i)
135          set2 = vm.open_array('StochasticInputSet', 'csv', 2 * i + 1)
136
137          # Define combinations of planes
138
139          dipdir_dip = cb.stochastic_joints(set1, set2)
140
141          # Number of sim, depends on the number of combinations of planes
142          n_sim = len(dipdir_dip[:, 0])
143
144          up_n_stable = 0  # Counter for stable blocks
145          up_n_failed = 0  # Counter for block with FOS lower than target
146          up_n_blocks = 0  # Counter for potential blocks
147          low_n_stable = 0  # Counter for stable blocks
148          low_n_failed = 0  # Counter for block with FOS lower than target
149          low_n_blocks = 0  # Counter for potential blocks
150
151          # Lists to store FoS and mode of failure
152
153          up_FOS = list()
154          up_mode_failure = list()
155          low_FOS = list()
156          low_mode_failure = list()
157          PA_FoS = list()
158          RST_Area = list()  # Area between bounds
159          upper_PoF = list()
160          lower_PoF = list()
161
162          # Slope modeled as Kent, with several scenarios (different kappa, beta)
163
164          n_scenarios = 1
165
166          for l in range(n_scenarios):
167
168              #in_slope_params = vm.open_array('SlopeSimulation', 'csv', l)
```

```
169
170            for ii in range(n_focal_elements):
171
172                for j in range(n_sim):
173
174                    # Probability assignment
175
176                    pa_FoS = (pa_comb[ii,0] * pa_comb[ii,1] * pa_comb[ii,2] *
177                             pa_comb[ii, 3])
178
179                    # FOR UPPER BOUND
180
181                    # Simulation of inputs according to assumed probability
182                    # distribution
183
184                    #dipdir_s = in_slope_params[j, 0]
185                    dipdir_s = 330
186                    #dip_s = in_slope_params[j, 1]
187                    dip_s = 70
188                    phi_1 = upper_inputs[ii, 0]
189                    #phi_1 = 32
190                    c_1 = upper_inputs[ii, 1]
191                    phi_2 = upper_inputs[ii, 0]
192                    # phi_2 = 32
193                    c_2 = upper_inputs[ii, 1]
194                    # h = up_h
195                    h = 15
196                    u_water = 9.81
197                    spec_gravity = up_spec_grav
198
199                    # Computation of intersection
200                    # Based on the cross product of normal vectors
201
202                    planes = (np.asarray([[dipdir_dip[j, 0], dipdir_dip[j, 1]],
203                                          [dipdir_dip[j, 2], dipdir_dip[j, 3]],
204                                          [dipdir_s, dip_s]]))
205                    # print(planes)
206                    n, x_cos, y_cos, z_cos, orientation = vm.direction_cosines(
207                        planes)
208                    vector1_cos = [x_cos[0], y_cos[0], z_cos[0]]
209                    vector2_cos = [x_cos[1], y_cos[1], z_cos[1]]
210                    slope = [x_cos[2], y_cos[2], z_cos[2]]
211
212                    # Intersection line plunge and trend is computed with cross
213                    # product.
214                    # Verification of the relative position of planes 1 and 2
215                    # Component X1 is checked. If positive, the cross product is
216                    # corrected (commuted)
217
218                    intersection = np.cross(vector1_cos, vector2_cos)
219                    if intersection[0] <= 0:
220                        intersection = intersection
221                    else:
222                        intersection = np.cross(vector2_cos, vector1_cos)
223
224                    x1i = intersection[0]
```

```
225                        x2i = intersection[1]
226                        x3i = intersection[2]
227                        inter_dipdir, inter_dip = vm.rectangular_to_cilindrical2(x1i,
228                                                                              x2i,
229                                                                              x3i)
230
231                        # Before computing the factor of safety, the kinematic conditi
232                        # for the wedge to form are checked.
233                        # Conditions are:
234                        # Angle between intersection trend and slope dip direction
235                        # has to be less than 90 degrees
236                        # Intersection plunge less than apparent slope dip
237
238                        # Conditional to calculate the angle between
239                        # intersection and slope
240
241                        if np.abs(inter_dipdir - dipdir_s) < 180:
242                            angle_inter_slope = np.abs(inter_dipdir - dipdir_s)
243                        else:
244                            angle_inter_slope = 360 - np.abs(inter_dipdir - dipdir_s)
245
246                        aparent_slope_dip = (
247                            np.degrees((np.arctan(np.tan(np.radians(dip_s))
248                                                 * np.cos(
249                                np.radians(angle_inter_slope))))))
250
251                        if (angle_inter_slope <= 90) and (inter_dip < aparent_slope_di
252                            up_FoS, up_failure = (fs.factor_of_safety(dipdir_dip[j, 0]
253                                                                     dipdir_dip[j, 1]
254                                                                     dipdir_dip[j, 2]
255                                                                     dipdir_dip[j, 3]
256                                                                     dipdir_s,
257                                                                     dip_s,
258                                                                     phi_1, c_1, phi_
259                                                                     c_2, h,
260                                                                     0, spec_gravity,
261                                                                     u_water, 0))
262                            if up_FoS >= 0:
263                                up_FoS = up_FoS
264                                up_FOS.append(up_FoS)
265                                up_mode_failure.append(up_failure)
266                                up_n_blocks = up_n_blocks + 1
267                            else:
268                                up_FoS = np.abs(up_FoS)
269                                up_FOS.append(up_FoS)
270                                up_n_blocks = up_n_blocks + 1
271
272                            # Assess blocks with FOS lower than target
273                            if up_FoS < target_FOS:
274                                up_n_failed = up_n_failed + 1
275
276                        else:
277                            up_n_stable = up_n_stable + 1
278                            # FoS = 100000
279                            # FOS.append(FoS)
280                            up_failure = "There is no failure"
```

```
281                              up_mode_failure.append(up_failure)
282
283                      # FOR THE LOWER BOUND
284
285                      # Input lower bound
286
287                      phi_1 = lower_inputs[ii, 0]
288                      c_1 = lower_inputs[ii, 1]
289                      phi_2 = lower_inputs[ii, 0]
290                      # phi_2 = 32
291                      c_2 = lower_inputs[ii, 1]
292                      # h = low_h
293                      h = 15
294                      u_water = 9.81
295                      spec_gravity = low_spec_grav
296
297                      # FOS computation
298
299                      if (angle_inter_slope <= 90) and (inter_dip < aparent_slope_di
300                          low_FoS, low_failure = (fs.factor_of_safety(dipdir_dip[j,
301                                                                      dipdir_dip[j,
302                                                                      dipdir_dip[j,
303                                                                      dipdir_dip[j,
304                                                                      dipdir_s,
305                                                                      dip_s, phi_1,
306                                                                      c_1, phi_2, c_
307                                                                      h, 0,
308                                                                      spec_gravity,
309                                                                      u_water, 0))
310                              if low_FoS >= 0:
311                                  low_FoS = low_FoS
312                                  low_FOS.append(low_FoS)
313                                  low_mode_failure.append(low_failure)
314                                  low_n_blocks = low_n_blocks + 1
315                              else:
316                                  low_FoS = np.abs(low_FoS)
317                                  low_FOS.append(low_FoS)
318                                  low_n_blocks = low_n_blocks + 1
319
320
321
322                                  #Assess blocks with FOS lower than target
323                              if low_FoS < target_FOS:
324                                  low_n_failed = low_n_failed + 1
325
326
327                          else:
328                              low_n_stable = low_n_stable + 1
329                              low_failure = "There is no failure"
330                              low_mode_failure.append(low_failure)
331
332                  # Probabilities
333                  # Probability of each realization
334                  up_prob = pa_FoS*(1 / up_n_blocks)
335
336                  # lists to stores weighted FOS
```

```
337
338                      for jj in range(up_n_blocks):
339                          PA_FoS.append(up_prob)
340
341                      # Restart counters
342
343                      up_n_blocks = 0
344                      up_prob = 0
345
346              low = np.asarray(low_FOS)
347              up = np.asarray(up_FOS)
348              pa = np.asarray(PA_FoS)
349
350              DS_FOS = np.zeros((np.size(low), 3))
351              DS_FOS[:, 0] = low[:]
352              DS_FOS[:, 1] = up[:]
353              DS_FOS[:, 2] = pa[:]
354
355              lower_FOS, lower_prob, upper_FOS, upper_prob, A = rs.random_sets(DS_FO
356
357              # Finding the probability of failure
358
359              x = np.where(lower_FOS >= target_FOS)
360              lower_FOS_PoF = lower_prob[np.min(x)]
361
362              x = np.where(upper_FOS >= target_FOS)
363              upper_FOS_PoF = upper_prob[np.min(x)]
364
365              lower_PoF.append(lower_FOS_PoF)
366              upper_PoF.append(upper_FOS_PoF)
367
368              RST_Area.append(A)
369
370              print("Area between bounds is: ", A, "for scenario", l,
371                    "for source,", i)
372              print("Lower POF is", lower_FOS_PoF, "for scenario", l,
373                    "for source,", i)
374              print("Upper POF is", upper_FOS_PoF  , "for scenario", l,
375                    "for source,", i)
376
377              # Finding the probability of having a FoS lower than 1, 2, 3, 4, 5
378              All_FOS = 10    # FOS to consider
379
380              #####################
381              ##  LOWER BOUND #####
382              ###############33####
383
384              lower_FOS_AllPoF = list()  # List to storage FoS lower than integer
385              for h in range(All_FOS):
386                  y = np.where(lower_FOS >= h+1)
387                  lower_FOS_All = lower_prob[np.min(y)]
388                  lower_FOS_AllPoF.append(lower_FOS_All)
389
390              np.savetxt(
391                  '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerALLPoF",
392                                          "Scenarios", l,
```

```
393                                              "Source", i, '.', "csv"), lower_FOS_AllP
394              delimiter=",", header="PoF", comments='')
395
396          lower_FOS_AllPoF = list()  # Re start list
397
398          ######################
399          ##  UPPER BOUND  #####
400          ###############33####
401
402          upper_FOS_AllPoF = list()  # List to storage FoS lower than integer
403          for h in range(All_FOS):
404              y = np.where(upper_FOS >= h + 1)
405              upper_FOS_All = upper_prob[np.min(y)]
406              upper_FOS_AllPoF.append(upper_FOS_All)
407
408          np.savetxt(
409              '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperALLPoF",
410                                       "Scenarios", l,
411                                       "Source", i, '.', "csv"), upper_FOS_AllP
412              delimiter=",", header="PoF", comments='')
413
414          upper_FOS_AllPoF = list()  # Re start lis
415
416          np.savetxt(
417              '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerFOSplot",
418                                       "Scenarios", l,
419                                       "Source", i, '.', "csv"), lower_FOS[1:],
420              delimiter=",", header="Area", comments='')
421
422          np.savetxt(
423              '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerFOSProb",
424                                       "Scenarios", l,
425                                       "Source", i, '.', "csv"), lower_prob[:-1
426              delimiter=",", header="Area", comments='')
427
428          np.savetxt(
429              '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperFOSplot",
430                                       "Scenarios", l,
431                                       "Source", i, '.', "csv"), upper_FOS[1:],
432              delimiter=",", header="Area", comments='')
433
434          np.savetxt(
435              '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperFOSProb",
436                                       "Scenarios", l,
437                                       "Source", i, '.', "csv"), upper_prob[:-1
438              delimiter=",", header="Area", comments='')
439
440          # Plot
441
442          plt.figure()
443          plt.plot(lower_FOS[1:], lower_prob[:-1], 'g-', linewidth=2.0)
444          plt.plot(upper_FOS[1:], upper_prob[:-1], 'r-', linewidth=2.0)
445          plt.ylim(0, 1.01)
446          plt.xlim(0, 10)
447          plt.xlabel('Factor of Safety')
448          plt.ylabel('Cumulative Probability')
```

```
449                #plt.show()
450
451                # Re start lists to store
452
453                PA_FoS = list()
454                low_FOS = list()
455                up_FOS = list()
456                up_failed = list()
457                low_failed = list()
458
459                plt.savefig('{}{}{}{}{}{}{}{}'.format('StochasticKentRandom','Probabilit
460                                        'Function',
461                                        'Scenario', l, "Source", i))
462
463        np.savetxt(
464            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "Area", "Scenarios",
465                                    "Source", i, '.', "csv"), RST_Area,
466            delimiter=",", header="Area", comments='')
467
468        np.savetxt(
469            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "LowerPoF",
470                                    "Scenarios",
471                                    l, "Source", i, '.', "csv"), lower_PoF,
472            delimiter=",", header="Probabilidad de falla", comments='')
473
474        np.savetxt(
475            '{}{}{}{}{}{}{}{}'.format("StochasticKentRandom", "UpperPoF",
476                                    "Scenarios",
477                                    l, "Source", i, '.', "csv"), upper_PoF,
478            delimiter=",", header="Probabilidad de falla", comments='')
479
480 end = time.time()
481 time_seconds = end -start
482 time_min = time_seconds / 60
483
484 print("I am done computing the factor of safety for",n_sim, " wedges ")
485
486 print("It took me", time_seconds, "seconds or", time_min, " minutes to complete "
487        "the", n_sim*n_scenarios*n_focal_elements*n_pieces_evidence,
488      "wedges FOS computation")
```

```
 1 '''
 2
 3 Research Project:
 4 Reliability Assessment of Rock Slopes by Discrete
 5 Elements and Random Sets
 6 Universidad Nacional de Colombia sede Bogota
 7 Author: Rodrigo Hernandez-Carrillo
 8 2/12/18
 9
10 # Copyright (c) Rodrigo Hernandez-Carrillo 2018
11
12 _____
13
14
15 This script performs a mixture fittiing using a Kent
16 distribution as base and a uniform distribution
17 to identify outlines.
18 The Expectation Maximization algorithm is applied
19 using the moment method instead of the maximun
20 likelihood approach
21
22 As input, directed cosines form original
23 clustering along with initial estimation
24 parameters is required
25
26 '''
27
28 import numpy as np
29 import pandas as pd
30 import KentDistribution1 as kd
31 import VectorMean as vm
32 import matplotlib.pyplot as plt
33 import mplstereonet
34 import warnings
35
36 # The following method performs the expectation step of the EM algorithm. It
37 # computes the current posterior probability of component membership or
38 # responsibility.
39 # As inputs it requires an array of unit vector, the mixing proportions, pi,
40 # the kent distribution parameters, ck, k, beta and epsilon and the number of
41 # clusters and points. Returns the reponsability matrix a step i.
42 # Ref: Peel, Whitten and McLachlan 2001
43 # It includes a try statment, to handle overflow errors when responsability
44 # values are too small. Instead it assigns zero
45
46
47
48 def expectation_initialize (x, pi, ck, k, beta, epsilon, n_clusters, n_points,
49                             pi_uniform):
50     matrix_responsability = list()
51     for c in range(n_clusters):
52         uniform_responsability = pi_uniform/(2*np.pi)
53         vector_responsability = list()
54         uniform_responsability_vector = list()
55         uniform_responsability_vector.append(uniform_responsability)
56         for i in range (n_points):
```

```
57              responsability = (pi[c] * ck[c] * np.exp(k[c] * np.dot(x[i,:],
58                                  epsilon[c,:,0]) + beta[c] * (np.dot(x[i,:],
59                                  epsilon[c,:,1]))**2 - beta[c] * (np.dot(x[i,:]
60                                  epsilon[c,:,2]))**2))
61
62              # print(epsilon)
63              #print('REPONSABILITY', responsability)
64              uniform_res= pi_uniform / (2 * np.pi)
65              vector_responsability.append(responsability)
66              uniform_responsability_vector.append(uniform_res)
67          vector_responsability.append(uniform_responsability)
68          vector_res = np.asarray(vector_responsability)
69          vector_res_uniform = np.asarray(uniform_responsability_vector)
70          #print('TAMANO VECTOR UNIFORME', vector_res_uniform.shape)
71          matrix_responsability.append(vector_res)
72
73      matrix_responsability.append(vector_res_uniform)
74      responsability_matrix = np.asarray(matrix_responsability)
75      sum_over_row = np.sum(responsability_matrix, axis = 0)
76      responsability_matrix = responsability_matrix / (sum_over_row)
77 #    print('RESPONSIBILDAD', responsability_matrix)
78      return responsability_matrix.T
79
80 # In the following a set of functions modified to update input parameters accordin
81 # to Kent distribution are given. This corresponds to the maximization step of the
82 # EM algorithm.
83
84 # This funtion computes the mixture coefficients, based on the reponsabilities and
85 # total number of points.
86 # Refs: Yasutomi and Tanaka 2016
87
88
89 def mixture_coefficient (responsability, n_points):
90      mc = np.sum(responsability, axis=0)
91      mixturePi = np.sum(responsability, axis=0) / (n_points + 1)
92      # print('ESTOS SON LOS COIEFICENTES DE AJUSTE', mc, mixturePi)
93      return mc, mixturePi
94
95 # This function modifies the input dip direction (first moment), according to the
96 # responsibilities defined at the expectation step. It is a modified version
97 # of the conventional analysis, to update parameters according to responsibility.
98 # Reminder: Parameters are computed according to moments method.
99 # Refs: Kent, 1982
100
101
102 def sample_mean(cos_x1, cos_x2, cos_x3, responsability, mc):
103      sum_x1 = np.sum(responsability[:-1] * cos_x1)
104      sum_x2 = np.sum(responsability[:-1]* cos_x2)
105      sum_x3 = np.sum(responsability[:-1]  * cos_x3)
106      resultant = np.sqrt(sum_x1 ** 2 + sum_x2 ** 2 + sum_x3 ** 2)
107      mean_resultant = resultant / mc
108      mean_unit_vector = (sum_x1 / resultant, sum_x2 /
109                          resultant, sum_x3 /
110                          resultant)
111
112      if mean_unit_vector[1] >= 0 and mean_unit_vector[2] >= 0:
```

```
113            Q = 0
114       else:
115           if mean_unit_vector[1] < 0 and mean_unit_vector[2] >= 0:
116               Q = (-2 * np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
117                                                 mean_unit_vector[
118                                                   1]))) + 180)
119           else:
120               if mean_unit_vector[1] >= 0 and mean_unit_vector[2] < 0:
121                   Q = (-2 * np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
122                                                 mean_unit_vector[
123                                                   1]))) + 360)
124               else:
125                   Q = 180
126
127       mean_longitude = (np.degrees(np.arctan(np.abs(mean_unit_vector[2] /
128                                             mean_unit_vector[
129                                               1]))) + Q)
130       mean_collatitude = (np.degrees(np.arccos(mean_unit_vector[0])))
131       #print(mean_longitude, mean_collatitude)
132       return (resultant, mean_resultant, mean_unit_vector, mean_longitude,
133               mean_collatitude)
134
135 # This  is a modified version of the conventional orientation matrix (second momen
136 # to update parameters according to responsibility
137 # Reminder: Parameters are computed according to moments method.
138 # Refs: Kent, 1982
139
140 def orientation_matrix(cos_x1, cos_x2, cos_x3, responsability,  mc):
141     cos_x1 = responsability[:-1] * cos_x1
142     cos_x2 = responsability[:-1] * cos_x2
143     cos_x3 = responsability[:-1] * cos_x3
144     orientation_matrix = (np.sum(cos_x1 ** 2)/(mc ),
145                            np.sum(cos_x1 * cos_x2)/(mc ),
146                         np.sum(cos_x1 * cos_x3)/(mc ),
147                         np.sum(cos_x1 * cos_x2)/(mc ),
148                         np.sum(cos_x2 ** 2)/(mc ),
149                         np.sum(cos_x2 * cos_x3)/(mc ),
150                         np.sum(cos_x1 * cos_x3)/(mc ),
151                         np.sum(cos_x2 * cos_x3)/(mc ),
152                         np.sum(cos_x3 ** 2)/(mc ))
153     orientation_matrix = np.asarray(orientation_matrix).reshape(3, 3)
154     return orientation_matrix
155
156 # This  is a modified version of the conventional Kent parameters computation by
157 # moments methods, to update parameters according to responsibility
158 # Function to compute the parameters of Kent distribution, from a given
159 # set of data
160 # Input: orientation matrix, rotation matrix, number of planes and mean
161 # resultant
162 # Output: kappa: concentration parameter. Beta: Ovalness. Major and minor
163 # axis.
164 # Ref. Fisher, Lewis and Embleton, 1987. Section 5.3
165
166 def estimation_kent_parameters(ori_mat, rot_mat, mean_res):
167     S = ori_mat
168     B = np.dot(np.transpose(rot_mat), np.dot(S, rot_mat))
```

```
169     psi = 0.5 * np.arctan(2 * B[1, 2] / (B[1, 1] - B[2, 2]))
170     k = ((1, 0, 0, 0, np.cos(psi), -np.sin(psi), 0, np.sin(psi),
171          np.cos(psi)))
172     k = np.asarray(k).reshape(3, 3)
173     g = np.dot(rot_mat, k)
174     v = np.dot(np.transpose(g), np.dot(S, g)) # Orthogonal Axis Matrix
175     q = v[1, 1] - v[2, 2]
176     kappa = 1 / (2 - 2 * mean_res - q) + 1 / (2 - 2 * mean_res + q)
177     beta = np.abs(
178         0.5 * (1 / (2 - 2 * mean_res - q) - 1 / (2 - 2 * mean_res +
179                                                    q)))
180     #print('kappa = ', kappa, 'beta = ', beta)
181     return g, kappa, beta, v
182
183 # In the following a function to plot the points on a stereonet is presented.
184 # As inputs the dip and dip direction are required. The functions creates a
185 # dictionary using pandas, then from that dictionary.
186 # From the dictionary, colors are mapped to the points, according
187 # to the labels resulting from the clustering
188 # Previously a figure (fig) has to be created.
189
190
191 def plot_stereonet (dip_direction, dip, label, fig):
192     planes = pd.DataFrame(dict(dip_direction=dip_direction, dip=dip, label=label))
193     ax = fig.add_subplot(111, projection='stereonet')
194     colors = { 0 : 'r', 1:'g', 2: 'b', 3: 'c', 4: 'p', 5:'c' }
195     (ax.pole(planes['dip_direction'], planes['dip'],
196             c=planes['label'].apply(lambda x: colors[x])))
197     plt.show()
198
199 if __name__ == "__main__":
200     exit
```

```
 1  '''
 2
 3  Research Project:
 4  Reliability Assessment of Rock Slopes by Discrete
 5  Elements and Random Sets
 6  Universidad Nacional de Colombia sede Bogota
 7  Author: Rodrigo Hernandez-Carrillo
 8  2/12/18
 9
10  # Copyright (c) Rodrigo Hernandez-Carrillo 2018
11
12  _____
13
14
15  This script performs the reliability updating, as new information is available,
16  using the EM algorithm assuming that planes orientation are Kent distributed
17
18  '''
19
20  import numpy as np
21  import time
22  import KentDistribution1 as kd
23  import VectorMean as vm
24  import EM_Algorithm as em
25  import matplotlib.pyplot as plt
26  import pandas as pd
27  import mplstereonet
28
29  start = time.time()
30
31  if __name__ == "__main__":
32
33      print('I am clustering and updating your orientation information, please wait')
34
35      # First, empty lists for each parameters are created, as well as, for the input
36      # dip direction and dip and direction cosines. Data from different input sets a
37      # the assembled (consolidated in each list. Through this lists, the algorithm
38      # is defined based on arrays in numpy. The number of clusters is an input given
39      # by the user
40
41      n_clusters = 10  # Number of clusters
42      dip_direction_dip = list()  # List for dip direction and dip
43      pi_uniform = 0.63 # reduced noise p=¿0.05
44      cosine = list()  # Empty list for direction cosines
45      k_mat = list()
46      size_mat = list()
47      beta_mat = list()
48      g_mat = list()
49      ck_mat = list()
50      cosine = list()
51      dip_direction_dip = list()
52
53      # The following loop computes the initial parameters given by the initial
54      # clusters. It reads a n_clusters number of csv input files, expressed as dip
55      # and dip direction. Afterwards, it computes and assembles the same number of
56      # clusters
```

```
57
58     for i in range(n_clusters):
59         dipdir_dip = vm.open_array('InputSet', 'csv', i) # All collected info
60         # dipdir_dip = vm.open_array('InputSet', 'csv', 3*i+1) # info collected fo
61         # each joint set dipdir_dip = vm.angle_adjustment(dipdir_dip)
62         n, x_cos, y_cos, z_cos, orientation = vm.direction_cosines(dipdir_dip)
63         r_length, r_mean, mean_vector, dipdir_mean, dip_mean = (vm.sample_mean(x_c
64                                                                 y_c
65                                                                 z_c
66         rotation = vm.rotation_matrix(dipdir_mean, dip_mean)
67         # print("Initial roration", rotation)
68         # print("Initial Orientation", orientation)
69         g, k, beta, v = (kd.estimation_kent_parameters(orientation,
70                                                        rotation, n,
71                                                        r_mean))
72         dip_direction_dip.append(dipdir_dip)
73         cosine.append((x_cos, y_cos, z_cos)),
74         ck = kd.estimation_ck(k, beta)
75         size_mat.append(n)
76         k_mat.append(k)
77         beta_mat.append(beta)
78         g_mat.append(g)
79         ck_mat.append(ck)
80
81     # Lists converted into arrays
82
83     dipdir_mat = np.vstack(dip_direction_dip)
84     cosine = np.hstack(cosine).T
85     k_mat = np.asarray(k_mat)
86     beta_mat = np.asarray(beta_mat)
87     g_mat = np.asarray(g_mat)
88     ck_mat = np.asarray(ck_mat)
89     size_mat = np.asarray(size_mat)
90
91     # number of total points and initial mixing coefficients, according to number
92     # data at each input cluster
93     n_points = len(cosine[:, 0])
94     mixing = (1 / n_points) * size_mat
95
96
97     # If require, number of clusters can be changed here
98
99
100    # Computation of initial responsibility matrix, according to inputs. Then,
101    # first update of mixing coefficients is carried out
102
103    responsable = em.expectation_initialize(cosine, mixing, ck_mat, k_mat, beta_ma
104                                            g_mat, n_clusters, n_points, pi_unifor
105    mc, pi_c = em.mixture_coefficient(responsable, n_points)
106
107    # Initial canvas and frame to plot
108    fig = plt.figure()
109
110    # While loop, to perform the iterative EM algorithm. Note that Kent parameters
111    # mean values and rotation matrix, are corrected by the responsibilities. Henc
112    # modified function from EM class are used, not those from KentDistribution cl
```

```
113
114      j = 0
115      while j < 1000:
116          mean_dipdir = list()
117          k_mat = list()
118          size_mat = list()
119          beta_mat = list()
120          g_mat = list()
121          ck_mat = list()
122          for c in range(n_clusters):
123              r_length, r_mean, mean_vector, dipdir_mean, dip_mean = em.sample_mean(
124                  cosine[:, 0], cosine[:, 1],
125                  cosine[:, 2],
126                  responsable[:, c], mc[c])
127              rotation = vm.rotation_matrix(dipdir_mean, dip_mean)
128              orientation = em.orientation_matrix(cosine[:, 0], cosine[:, 1],
129                                                  cosine[:, 2], responsable[:, c],
130                                                  mc[c])
131              g, k, beta, v = em.estimation_kent_parameters(orientation,
132                                                            rotation, r_me
133              ck = kd.estimation_ck(k, beta)
134              k_mat.append(k)
135              beta_mat.append(beta)
136              g_mat.append(g)
137              ck_mat.append(ck)
138              mean_dipdir = np.append(dipdir_mean, dip_mean )
139          k_mat = np.asarray(k_mat)
140          beta_mat = np.asarray(beta_mat)
141          g_mat = np.asarray(g_mat)
142          ck_mat = np.asarray(ck_mat)
143          responsable = em.expectation_initialize(cosine, mixing, ck_mat, k_mat,
144                                                  beta_mat, g_mat, n_clusters,
145                                                  n_points, pi_uniform)
146          mc, pi_c = em.mixture_coefficient(responsable, n_points)
147          mixing = pi_c
148          print("COEFICIENTE DE MEZCLA", pi_c)
149          print(mean_dipdir)
150           j = j + 1
151
152
153      # Save a csv with the responsability matrix
154
155      np.savetxt(
156          '{}{}{}{}'.format("REponsabilty", "Matrix",
157                            '.', "csv"), responsable,
158          delimiter=",", header="Responsability Index", comments='')
159
160
161
162      # Extract the column ID from each row, to select the cluster where each
163      # pole belongs to
164
165      max_respons = np.argmax(responsable, axis = 1)
166      max_respons = max_respons + 1
167      clusters = list(max_respons[:-1])
168
```

```
169     # To define a matrix including the dip direction, dip and cluster ID after
170     # mixing by EM algorithm
171
172
173     dip_direction_dip=list(dipdir_mat.T)
174     dip_direction_dip.append(max_respons[0:-1])
175     clustered_poles = np.vstack(dip_direction_dip)
176     clustered_poles = clustered_poles.T
177
178     # To save a file with the orientations linked to the corresponding cluster,
179     # after clustering process
180
181     np.savetxt(
182         '{}{}{}{}'.format("Clustered", "PolesAllJoints",
183                           '.', "csv"), clustered_poles,
184         delimiter=",", header="Dip direction, Dip, Domain", comments='')
185
186     # Routine to save a file with the weithing coefficients assigned to each
187     # joint set
188
189     np.savetxt(
190         '{}{}{}{}'.format("WeightingCoeeficients", "AllJoints",
191                           '.', "csv"), pi_c,
192         delimiter=",", header="Dip direction, Dip, Domain", comments='')
193
194     # To time the computation process
195
196     end = time.time()
197     time_seconds = end - start
198     time_min = time_seconds / 60
199
200     print("It took me", time_seconds, "seconds or", time_min, " minutes to update"
201                                                     "the",
202         n_clusters,  "pieces of information available")
203
204
```