Dimensional and Mechanical Stability Analysis of Caves in Bahia, Brazil

by

Camillo Rossetto

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday December 15, 2020 at 10:30 AM.

Student number: 4951840
Thesis committee: Prof. G. Bertotti,   TU Delft, supervisor
                   Prof. A. Barnhoorn,   TU Delft
                   S. de Hoop,   TU Delft

This thesis is confidential and cannot be made public until December 15, 2020

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

This page is left blank intentionally

Preface

First of all I would like to thank my thesis supervisor Prof. Giovanni Bertotti, for giving me the opportunity to take part of a very stimulating and interesting project, during which he shared his great experience and love for science. His comments and suggestions constantly helped me to develop critical thinking when facing many challenges and I will carry this with me also after my academical years.

A special thank goes to Stephan de Hoop, who daily shared his knowledge and skills, helping me from the first to the last steps of my project which wouldn't be the same without his support.

The fieldwork that took place in Brazil during July/August 2019 was organized and funded by TUDelft, Universidade Federal do Rio Grande do Norte, Stichting Molengraaf Fonds (STMF) and Shell Brazil to which I am immensely grateful since I will never forget those two weeks. The perfect organization made me enjoy an highly safe and productive environment with amazing people in an equally amazing country, which is Brazil.

I want to thank especially my father Roberto, my mother Loreta and my sister Giorgia for their continuous support and encouragement during this amazing experience in the Netherlands that surely changed my life.

Camillo Rossetto
Delft, December 2020

Contents

List of Figures

List of Tables

List of Reactions

1

Abstract

The interest in carbonate lithologies has risen in the last few decades, due to discoveries of giant and supergiant carbonate reservoirs. While studying the properties of this lithology, the existence of cave systems at great depths was highlighted, since their presence can lead to drilling related problems, as well as change the flow direction of fluids in the subsurface. The possibility of predicting their orientations, dimensions and mechanical stability in the subsurface would therefore be of great help in the hydrocarbon business, which is why in this study, thanks to data provided by a Lidar laser scanner, a dimensional analysis of caves' conduits related to their position within the karst system is performed. The origin of caves can be hypogenic or epigenic and they can present different patterns of distribution of conduits. When deformation events cause the generation of orthogonal sets of joints in the rock layers and fractures intersect with each other and with bedding planes, the fluid flow in the subsurface becomes channelized into these features. The combination of this and speleogenetic processes leads to the development of maze caves, whose conduits' dimensions vary within a system as a consequence of differences of continuity and connectivity along a fracture network. Variations in the shapes of caves' conduits can be observed as well, depending on the supply of sediment and the position relative to the water table during dissolution. Mechanical deformation of karst systems can be simulated by using the linear elastic deformation mathematical model. In this way it is possible to study the influence that mechanical properties such as Young's modulus and Poisson's ratio have on the mechanical stability of caves in the subsurface.

Keywords: Carbonate reservoirs, speleogenesis, maze caves, mechanical deformation

2

Introduction

Speleogenesis is the process of formation of cave systems that develop through the flow of fluids along fractures such as joints (Audra and Palmer, 2015). The dominant speleogenetic domains are two:(1) the hypogenic domain consisting of aggressive fluids, usually acid because enriched in CO_2 or H_2S, reaching the cave-forming zone from depth, and (2) the epigenic domain, with water recharge coming from the surface, where the carbonic acid present in the soil and in the atmosphere dissolves into water (Klimchouk, 2012).

Speleogenesis takes place mainly in carbonate rocks, even if karstic systems can be found to be developed in siliciclastic rocks (Vasconcelos et al., 2013) and in evaporite rocks (Palmer, 2011). The general interest in carbonate rocks grew in the last decades, after that many discoveries highlighted the presence of some of the world's largest hydrocarbon fields in this lithology. Carbonate reservoirs drew attention not only for the significant quantities of hydrocarbons, but also because of their hydraulic properties that can allow to obtain good flow rates (Ehrenberg and Nadeau, 2005).

The rising attention for these reservoirs highlighted the presence of cave systems at great depths (Tian et al., 2016; Ardakani et al., 2013), that can lead to drilling and production related troubles since they can cause well collapse and form thief zones, where circulating fluids get lost. However not only negative sides are related to karsts, in fact they can significantly increase the fluid flow in carbonate reservoirs.

Even if at first caves found deep in the subsurface were considered as paleokarsts developed at shallow depths by epigenic processes and then brought to great depths during their geological history, it became clear that this situation applied only to a minimum part of them. Hypogene speleogenesis is the leading process in the majority of cases and therefore an understanding of the mechanisms and conditions involved is crucial in order to make predictions about the properties and the distribution of these features (Klimchouk, 2012).

The aim of this study is to provide geometrical and dimensional data useful to formulate new prediction models for cave systems in the subsurface, since there are a lot of uncertainties related to the forecast of the size and the position of karst systems at great depths. Data are obtained through cross sections analysis of conduits belonging to three different caves, namely Lapinha, Marota and Lapa do Bode. They are located in the Irecê basin, in the northern part of the São Francisco Craton which hosts the Una Group where multiple cave systems are developed in a mainly carbonate lithology. Several studies that investigated the influence of tectonic structures on the development of cave systems had been located in this area, demonstrating how speleogenesis is closely related and influenced by tectonic events (Ennes-Silva et al., 2016; Pontes et al., 2020; Klimchouk et al., 2016; Silva et al., 2017; Klimchouk and Ford, 2000).

The second goal is to perform an analysis of the influence that these features have on reservoirs' mechanical properties, considering that their impact on carbonate reservoirs qualities is still unclear. The main challenge faced in order to reach this goal is to find a software that can simulate the mechanical stability of a cave system in the

subsurface. The simulation therefore has to take into account the lithostatic pressure generated by the overlying rock layers, but also the influence that an internal flow will have on the cave's stability. Due to difficulties in the simulation, this goal can be only partially achieved, as completely realistic conditions are difficult to model. Furthermore, the calculation of geometrical and dimensional parameters is based on the extraction of caves' conduits cross sections from point clouds. One point cloud for each of the caves that has been investigated is available, thanks to the data collection performed in Brazil through a portable laser scanner that allowed to obtain 3D point clouds. The parameters obtained in this way may help to develop a prediction model, which would be an important step since cave features are too small to be clearly distinguished in a seismic section. Their dimension is in fact smaller than the resolution of seismic surveys, which is usually around 10-15 meters at shallow depths, while it is around 20-30 meters at great depths (Kallweit and Wood, 1982). Well data cannot be used as well for their detection, since the heterogeneity and the sizes involved are too large. A new technique based on the identification of amplitude anomalies in seismic surveys aims to highlight the presence of karst features in the subsurface (Basso et al., 2018), however more research is needed in order to have a clear understanding on this subject.

3

Geological setting

The study area is located in northern part of the state of Bahia, Brazil, in the northern part of the São Francisco Craton and more precisely inside the Irecê Basin, where the succession has been well studied. Here the Una Group, which was formed during the Neoproterozoic, presents a thickness of at least 1.2 km in the Irecê area. This Group is considered to be correlative with the Bambuì Group located in the São Francisco Basin, that together with the Irecê Basin are considered to be the residuum of an extended carbonate platform that during the Neo-Proterozoic/Early Paleozoic covered more than 300000 km^2 of the São Francisco Craton . At the basal part of the Una Group a glacial unit is present, overlaid by a thick carbonate sequence (Misi and Veizer, 1998).

Within the Irecê Basin, the glacial unit, called Bebedouro Formation, is represented by deposits such as diamictites and tillites and, when present, is in angular unconformity with the underlying Mesoproterozoic, Paleoproterozoic and Archean units. The overlying formation, the Salitre Formation, is where the caves we investigated formed and it is composed by limestones, dolostones and grainstones which were deposited on a tidal flat environment. The carbonate sequence is or in unconformity with the glaciogenic sequence, or it can be directly in contact with the metamorphosed siliciclastic Mesoproterozoic basement. The basement, represented by the Chapada Diamantina Group, consists of quartzites, metasiltstones, conglomereates and sericiteschists. On the top of the succession a Paleogene-Quaternary cover lies, composed by the Caatinga limestones or by unconsolidated soil derived mainly by the erosion of the Salitre Formation (Borges et al., 2016). A stratigraphic column of the area is presented at figure 3.1.

AGE	GROUP	FORMATION	THICKNESS (m)	LITHOLOGY	LITHOLOGIC DESCRIPTION
TERTIARY & QUATERNARY					ALLUVIAL SAND SILT & CLAY
					DETRITAL COVER OF SAND & CLAY
LATE PRECAMBRIAN	BAMBUÍ (UNA)	SALITRE	30 to 80		LIMESTONE
					LIMESTONE WITH LENSES OF ARGILLITE
					CALCAREOUS DOLOMITE
		BEBEDOURO	1 to 16		DIAMICTITE WITH LENSES OF SANDY-SILTSTONE
					SILTSTONE WITH SCATTERED STONES
	CHAPADA DIAMANTINA	MORRO do CHAPÉU	200		ORTHOQUARTZITE
					SANDY-SILTSTONE & ARGILLITE
					CONGLOMERATE

Figure 3.1: Stratigraphic column of the succession present in the Irecê Basin. From Montes et al., 1985

Around 1.0 Ga in the São Francisco paleocontinent an important glacial event occurred, during which the glaciers started to move from West to East and the Bebedouro Formation was deposited. After the consequent deglaciation between 750 and 650 Ma, when also the Brasiliano Orogeny started, the São Francisco Craton subsided and a general sea level rise occurred, creating the environment suitable for the development of a carbonate platform, which is today represented by the Salitre Formation (Ennes-Silva et al., 2016) dated at 514 ± 33 Ma with a Pb-Pb isochron and rock magnetic and paleomagnetic data (Trindade et al., 2004). During the Brasiliano orogeny, the position of the Irecê Basin close to the edges of the Craton, caused the movement of the Craton to interact with a fold belt located in the northern part, the Riacho do Pontal fold belt (figure 3.2). This interaction had as a consequence the formation of brittle-ductile structures such as thrust faults and folds with NNE-SSW and E-W strike and southward vergence for what concerns the folds. These structures are more deformed the more we go towards the North and they can be observed on figure 3.2 (Trindade et al., 2004; Ennes-Silva et al., 2016; Borges et al., 2016).
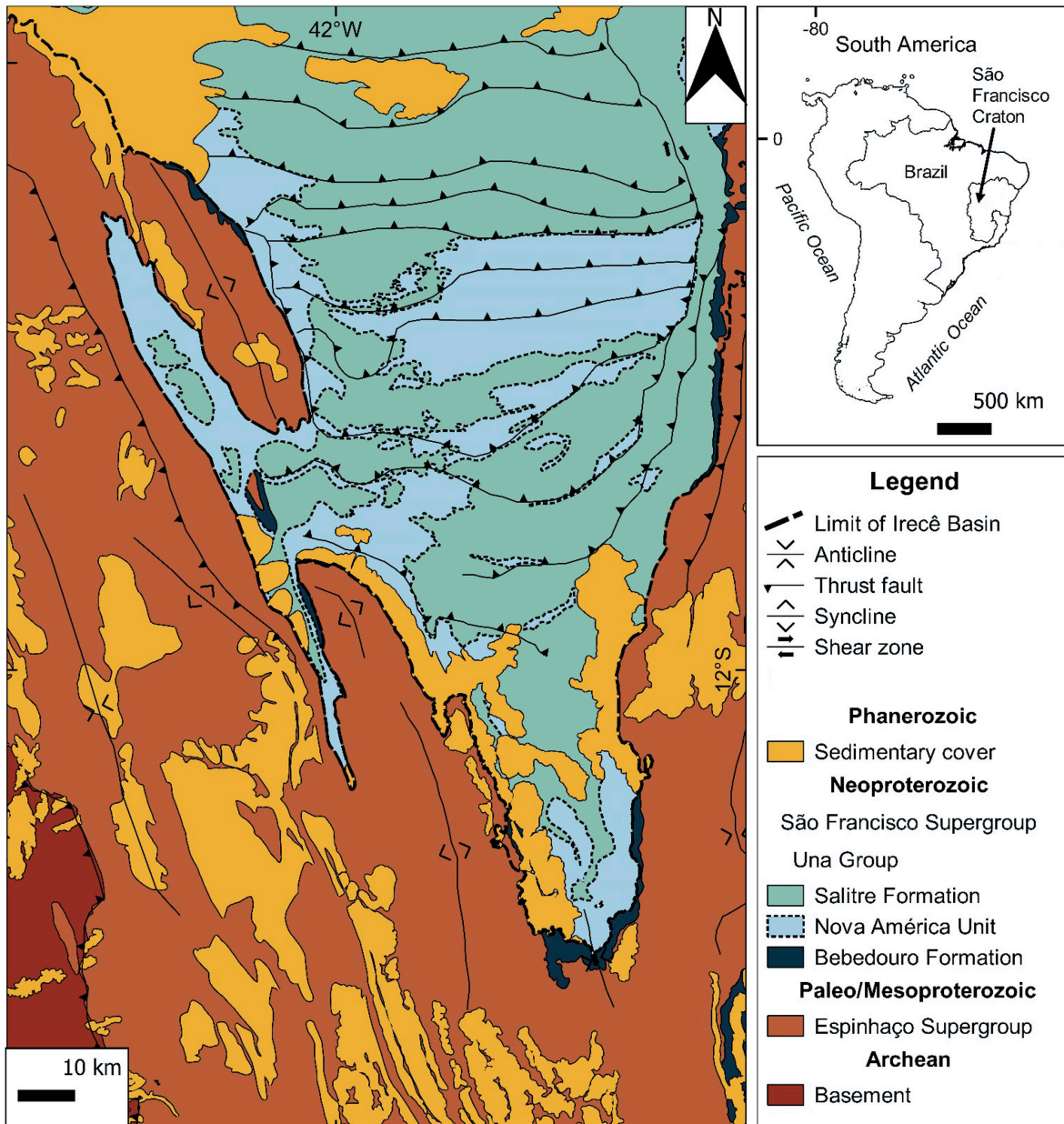
Figure 3.2: A map of the Irecê Basin where the general structural setting can be observed. Modified from Redivo et al., 2019.

4
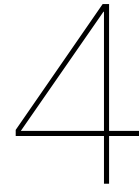
Speleogenesis

Before discussing the data obtained, the processes that lead to the formation of caves are presented in order to understand which are the most important influencing factors that affect speleogenesis. The focus will be in particular on carbonate speleogenesis and on the development of maze caves, specific caves characterized by complex patterns of conduits that are repeatedly connected forming perpendicular or almost perpendicular angles on a plan view.

4.1. Conduits Enlargement Mechanisms

We can start from a broad definition of speleogenesis by citing Andreychouk (1991), who stated that a karst is a "system of processes and phenomena developing and occurring underground and at the Earth's surface as a result of the interaction (dissolution, transport, and deposition of matter) of natural waters with rocks that are soluble in the given situation." (Andreychouk et al., 2009). One of the fundamental factors that lead to the formation of cave systems is thus the presence of a groundwater flow that grants the removal of dissolved material since, in order to keep dissolving the material, water needs to be undersaturated in the solute.

Caves can develop in many different settings, but there are two main domains that can be identified: the epigenic and the hypogenic domains. Epigenic caves are closely related to a water flow coming from the surface, while hypogenic caves owe their formation to a rising flow from below into a cave forming zone into artesian or confined conditions, with a recharge independent from the overlying layers (Klimchouk, 2007). Mazes can develop in both epigene and hypogene settings; in epigene conditions they form thanks to diffuse infiltration of water through an impermeable insoluble layer or thanks to sinking streams that allow to have a water flow. In hypogene conditions instead, they form with the cooperation of sources of deep aggressiveness or through a process that involves meteoric water rising in the distal downflow part of large flow systems and reaching soluble layers within two permeable insoluble layers (transverse hypogenic cave origin). In this last process the insoluble layers are supposed to limit the flow and to allow the fissures to enlarge homogeneously.

Three distinctive patterns of maze caves can be recognized: network caves, that form preferably where a set of intersecting fractures is present, spongework caves, that generate where intergranular porosity is available and anastomotic caves, that develop especially along bedding-planes or low angle faults (figure 4.1). The focus here will be on network mazes, since the caves that we are going to analyze are a result of dissolutional processes of highly fractured zones.

During the earlier stages of development the flow from a recharge point to a discharge point occurs along fractures and fissures, which with the passing of time (10^4 - 10^5 yr) will enlarge and become cave conduits. A slow and uniform dissolution can start the development of a long passage, but not all of the joints will evolve into
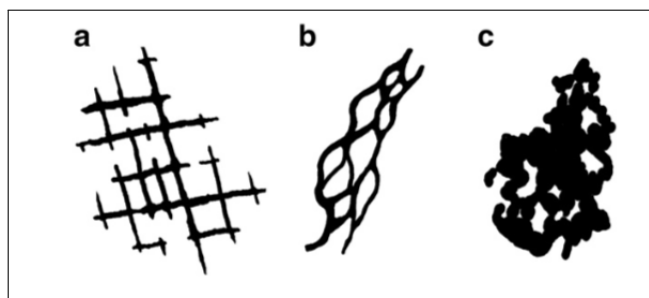
Figure 4.1: Maze caves possible patterns. a) network cave; b) anastomotic cave; c) spongework cave. From Palmer, 2011.

cave conduits. There are in fact preferential evolutional patterns in the subsurface, depending from the fact that the shortest routes with highest discharge and CO_2 content are favoured on the others. The evolution of a conduit usually faces a first stage in which the enlargement rate is proportional to the discharge and a second stage in which it depends only on kinetics, that starts when the flow inside the opening makes the transition from laminar to turbulent regime (Palmer, 1991). When different fracture sets intersect each other, the conduits will follow this pattern and therefore form features called nodes. The nodes connect different conduits and hence may cause different flows to interact with each other.

4.2. Structural Influence

Since the dissolutional development of caves is strongly related to the presence and the direction of joints, it is clear how the structural regional geology plays an important role in speleogenesis. Fracture corridors condition the pathways of fluid flow and therefore of karst systems, but the most favourable setting for cave passages development result to be at the intersections between bedding planes and fractures. Fault damage zones and fold hinges are usually related to highly fractured areas, which is why in these settings it is more likely to find hydraulically connected fracture networks where cave conduits can form (Ennes-Silva et al., 2016; Pontes et al., 2020). Structural analysis at the scale of karst systems may therefore give important information about their orientation.

4.3. Hydrogeological and Chemical Mechanisms

An understanding of the hydrogeology of the upper crust is fundamental, since a change in the hydrogeological setting conditions speleogenesis. This applies especially to hypogene karst, where a comprehension of the hydrodynamic of the upper crust and of the processes that may alter the fluids in the subsurface is crucial. The first step is to distinguish the types of fluids present in the subsurface and to recognize what is their source. There are exogenic fluids, composed by meteoric water and connate water (present in the pores of the rock at the moment of its formation) and endogenic fluids, generated during metamorphism (metamorphic fluids) and during the crystallization of magma (juvenile or magmatic fluids). These fluids can also easily mix, especially in deep parts of basins, generating the so-called formation waters.

Chemical reactions as well must be considered since they are the driving processes that control dissolution in the subsurface. An important influence in the speleogenesis is the transverse flow between different aquifers and strata, that in the subsurface can bring to mix waters with different chemistry, gas content and temperature. The main chemical compounds and mechanisms that bring to dissolution and chemical aggressiveness, especially in deep settings, are the followings. (1) Carbonic acid, that acts both in epigenic and hypogenic speleogenesis. In the former case the CO_2 source comes from meteoric water that, while infiltrating, carries the carbon dioxide absorbed by soil and atmosphere, while in the latter the CO_2 sources are metamorphism, devolatilization reactions, thermal degradation and oxidation of organic matter (Audra and Palmer, 2015; Klimchouk, 2012). When CO_2 mixes with

water, carbonic acid is generated as following

$$CO_2 + H_2O \longrightarrow H_2CO_3 \cdot \tag{1}$$

When carbonic acid makes contact with limestone it causes it to dissolve according to the reaction

$$H_2CO_3 + CaCO_3 \longrightarrow Ca(HCO_3)_2 \cdot \tag{2}$$

(2) Dissolution by cooling thermal water along the rising paths of thermal water, since the solubility of calcite increases with decreasing temperatures. Near the surface however the solubility of carbonates decreases, as well as the CO_2 concentration, causing calcite precipitation. (3) Sulfuric acid and hydrogen sulfide, with the first one formed from oxidation of hydrogen sulfide or of iron sulfides and the second one coming from depth as a result of microbial and thermal reduction when organic carbon is present (Palmer, 2013). (4) Mixing corrosion, caused by the mix of waters with different contents of carbon dioxide, hydrogen sulfide or salts that can increase the dissolution of carbonates. (5) Organic acids, that form especially when organic matter maturation occurs, causing the release of CO_2 and H_2S. (6) Dissolution of evaporites and in mixed carbonate/sulfate strata, with no acidic source necessarily required in the latter case (Klimchouk, 2012).

All the processes and mechanisms explained above are summarized in figure 4.2.
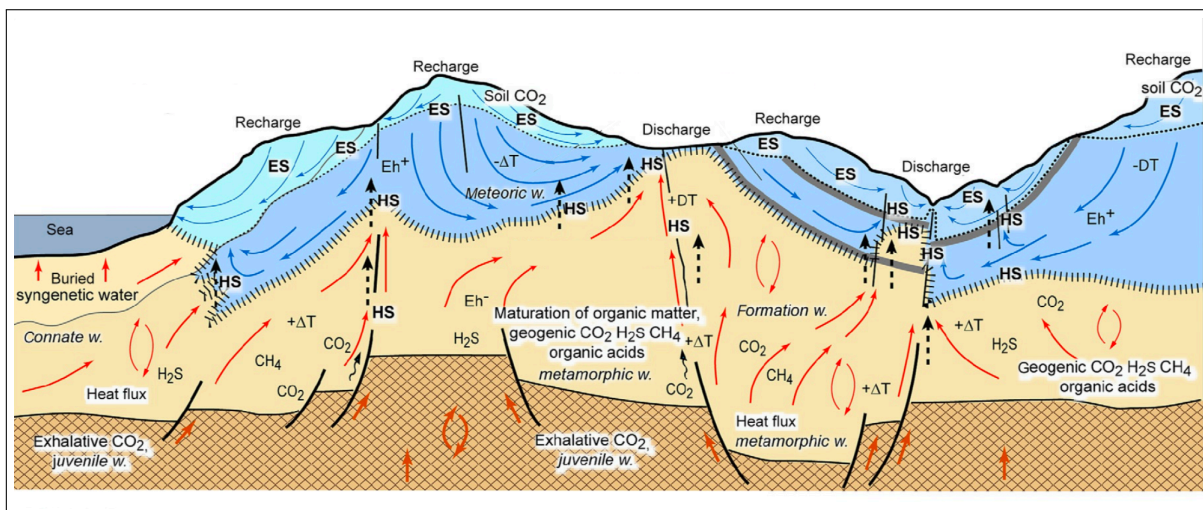
Figure 4.2: Schematic representation of the hydrogeologic settings and chemical processes that may bring to the formation of a cave system. Here several symbols are present and need to be defined: ES=epigene speleogenesis; HS=hypogene speleogenesis; EH[+] and EH[-] are redox conditions respectively oxidizing and reducing; blue arrows indicate meteoric water flow, while red arrows indicate basinal flow. Undulated black arrows represent gas inputs. Modified from Klimchouk, 2012.

.

5

Methodology

The goal of 2019's field work was to map seven caves in total, namely Calisto, Ioiô, Lapa do Bode, Lapinha, Lapinha de Iramaia, Marota and Torrinha (figure 5.1).
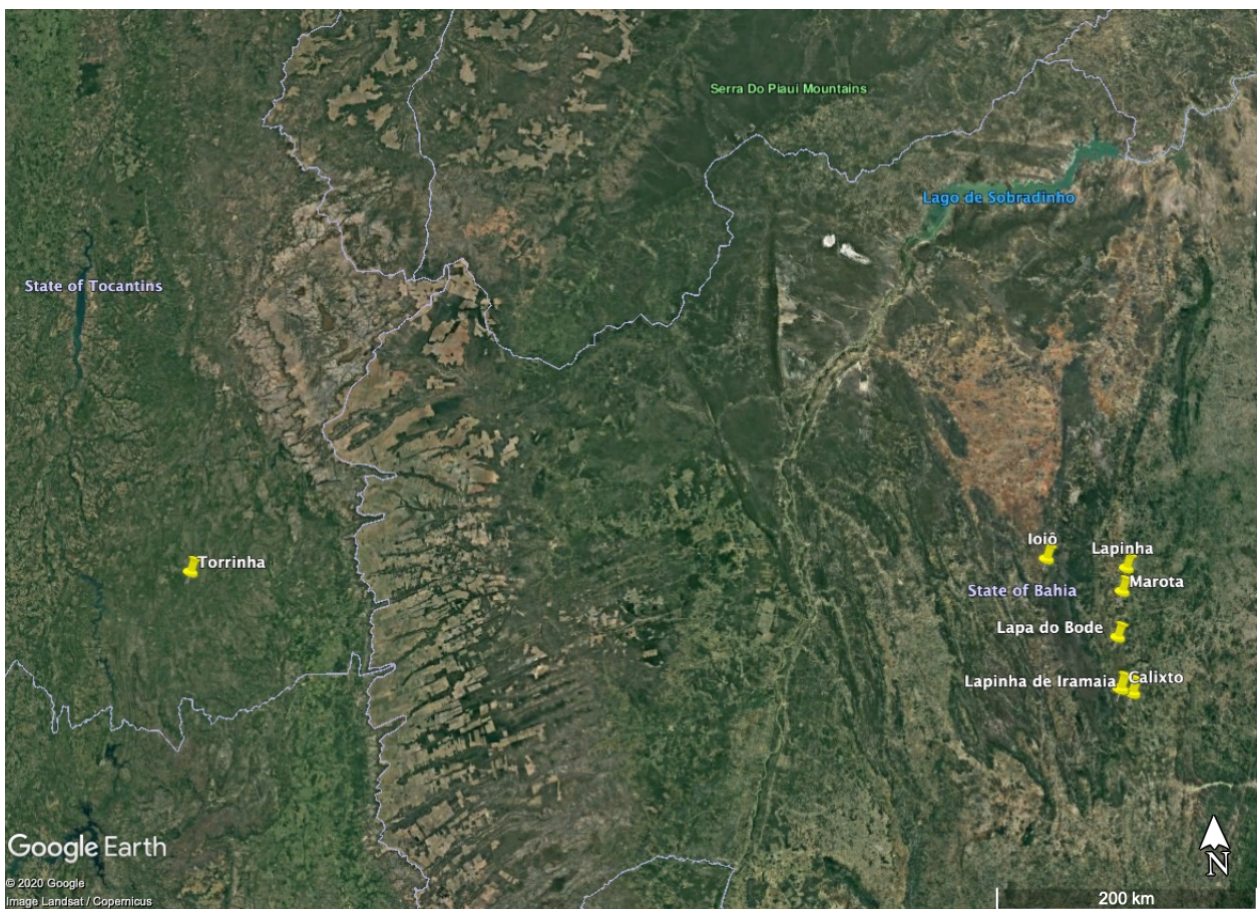
Figure 5.1: Map view of the study area. Each of the indicators represent one of the mapped caves.

The data acquired consist of seven different point clouds, one for each cave, that are 3D representations of these. Since the main focus is on maze caves, only the ones where this pattern can be recognized are studied,

being Lapinha, Lapa do bode and Marota (figure 5.2).
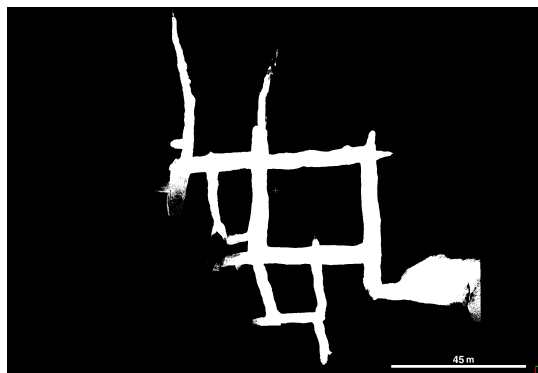
(a) Lapinha cave plan view
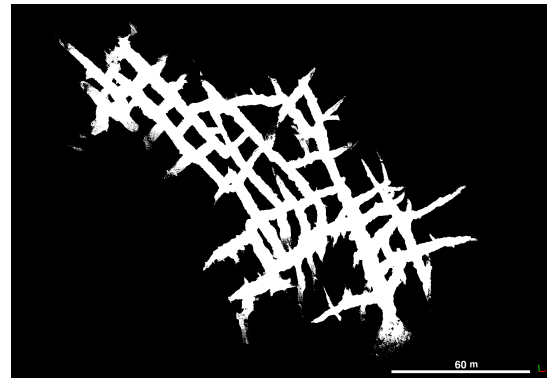
(b) Lapa do bode cave plan view
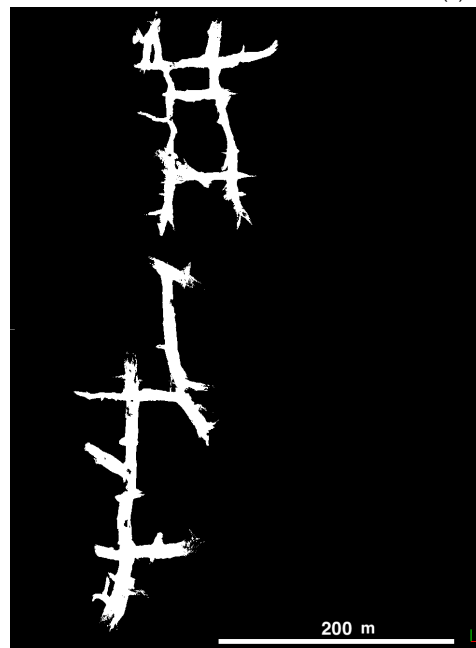
(c) Marota cave plan view

Figure 5.2: Plan views of the caves, where it is possible to recognize a network maze pattern in each of the caves.

Through an analysis of the conduits' shapes and of the orientation of the passages, what we want to achieve is to recognize patterns of variation of the dimensions of the conduits related to their position inside the cave system. Once this is determined, it is possible to make calculations regarding the dissolved volumes in the subsurface related to karst systems. In the end an investigation about the mechanical properties of these features at the variation of the material's properties is performed, even if the conditions of the simulations do not satisfactorily recreate subsurface conditions. Anyways, conclusions regarding the influence that physical properties of the material surrounding a cave have on the deformation of the rock, can still be drawn.

5.1. Data acquisition

Data acquisition was performed through a ZEB-REVO RT portable laser scanner, developed by geoSLAM, which is able to provide accurate 3D point clouds of the areas in which the user is walking. This is made possible by a 2D scanner and an inertial measurement unit which can rotate thanks to a motor drive. It is this motion that gives the third dimension, through which 3D data can be generated. 43200 points are registered every second, with a scan range noise of ±30 mm. The device, thanks to its lightness (it weighs in fact only five kilograms),

is easily maneuverable, which makes it very suitable for caves scanning purposes compared to the traditional terrestrial laser scanners. In order to collect point clouds as accurate as possible, geoSLAM suggests to not perform acquisitions that last more than 30 minutes and, for every acquisition, to create and close a loop while walking, in order to reduce any data drift. This means that for the biggest caves many surveys are required in order to map the entire area of interest.

5.2. Alignment and Denoising

Since for every cave many acquisitions were necessary, the 3D point clouds need to be aligned in order to create a unique point cloud for each cave. The free license software CloudCompare allows to do this after loading in it the .laz file of each point cloud. The first step is to manually align the clouds and then, using the sections that are in common between each cloud, to extract a matrix through which it is possible to automatically align the point clouds. This generates a single point cloud for each cave. The size of the cloud plays an important role, since the bigger the file we work with, the longer are the computational times required to extract data from it. In order to reduce the size it is possible to apply a noise reduction, which consists in creating a filtered point cloud in which outliers and noisy points are removed. This can be done with the tool 'Statistical Outlier Removal' in CloudCompare or with the 'pcdenoise' command in Matlab. In both cases an algorithm calculates first the mean of the average distance to neighbors of all points and then, by specifying a threshold of average distance of a point to its nearest neighbor, the outliers are identified and removed. In our case a value of 10 was chosen for the calculation of the mean of the average distance to neighbors for each point and a value of 3 was chosen as a threshold of average distance of each point to its closest neighbour. Here it is suggested to apply the denoising preferably through CloudCompare than through Matlab, since the computational times are way shorter. For instance, if we consider the smallest sized point cloud between the ones we are analyzing, which is the pointcloud of Lapinha cave (1.89 GB size), the denoising through cloudcompare takes approssimatively 11:35 minutes, while the denoising through matlab takes 35:15 minutes, which is more than three times longer. The size of the point clouds after the denoising process are listed in table 5.1.

Cave's Name	Number of Points	.PLY File Size
Lapinha	62.936.757	1.87 GB
Lapa do Bode	135.574.841	2.14 GB
Marota	162.908.388	3.21 GB

Table 5.1: Caves' files dimensions.

5.3. Skeletonization

In order to obtain the data we want, it is necessary to draw a skeleton for each cave. A skeleton is a set of 1D curves used for simplify 2D and 3D objects and to make it easy to process and visualize data. The ideal skeleton has to show five main qualities: (1) the topology of the object has to be well represented, which in our case means that it has to have the same number of conduits and nodes. (2) It has to be one dimensional and as simple as possible. (3) It must extend along the center of the object, meaning that in our case, for instance thinking about a conduit, the distance from the side walls has to be the same. (4) It has to maintain the geometry of the object. (5) It has to recover the original object (Saha et al., 2017).

At the beginning an attempt to automatically extract the skeleton from each cave was tried, however due to results that were not satisfactory a manual approach has been used. This involves the manual drawing of the skeleton through the software QGIS, which gives a shapefile of it as a result. In order to import into this software a point cloud, it is necessary to convert it into a textfile through CloudCompare. The size of the textfile plays an important role on the proper operation of the QGIS software, since files that are too big cause it to crash. The size can be easily reduced by subsampling the point cloud in CloudCompare where, since we are now only interested in recognizing the shape of the cave from a plan view, a random subsampling can be applied. This consists into

choosing the number of points that we want to keep and random sample them from the point cloud, which can be the original one or the denoised one. The skeletons obtained in this way are displayed in figure 5.3.

(a) Marota's skeleton

(b) Lapinha's skeleton

(c) Lapa do Bode's skeleton
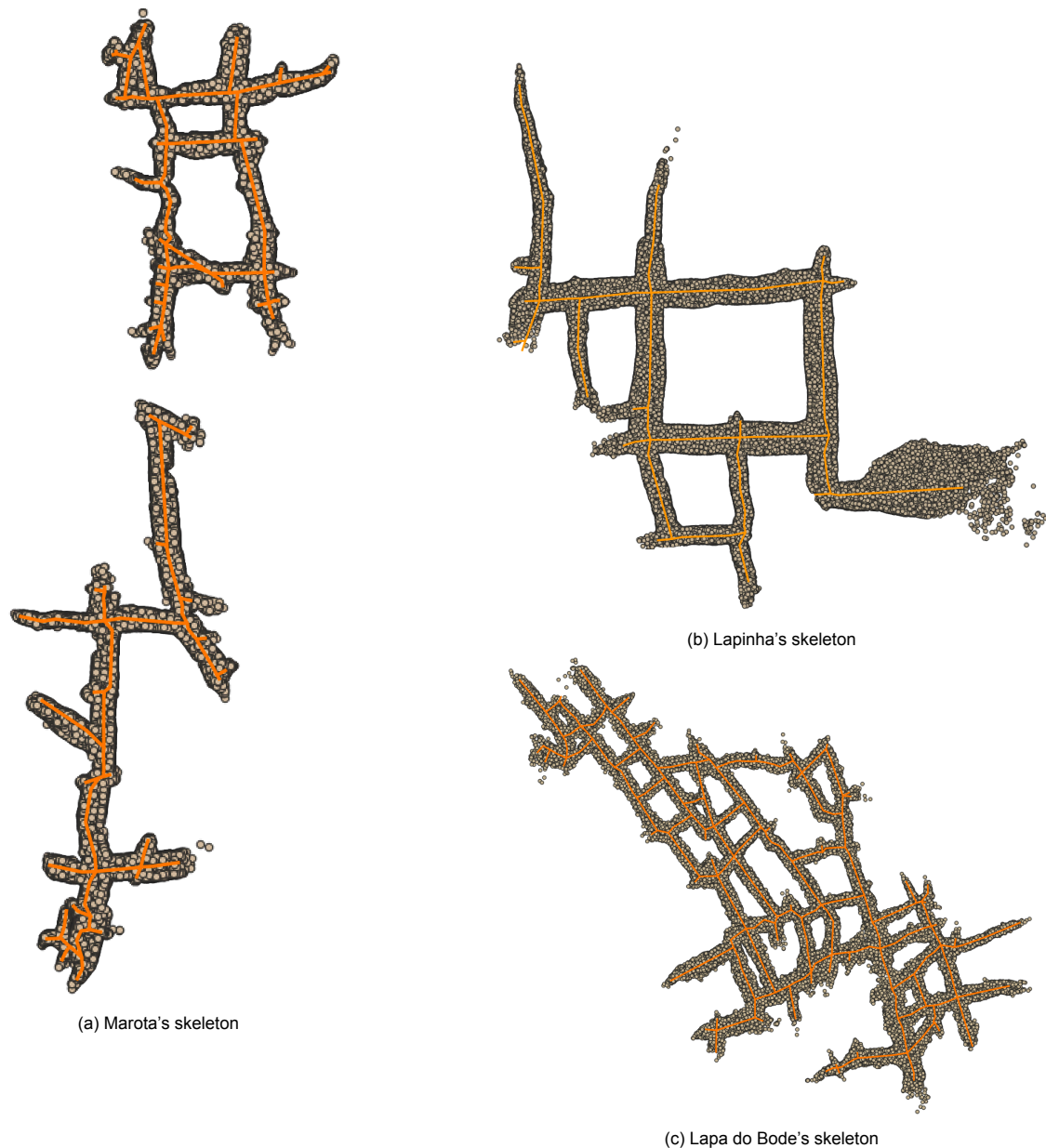
Figure 5.3: This figure displays the skeletons of the three caves in the QGIS software. The orange lines represent the skeletons, while the points are a representation of the random subsampled point clouds imported.

5.4. Cross Sections Analysis

The cross sections extraction and analysis is the fundamental step that allows to obtain our desired data. The base thinking behind this process is that multiple cross sections for every cave's conduit have to be extracted in order to obtain data from them like the width and the height distribution of a conduit along its length. As it is shown in the appendix-B, on average a conduit's shape can resemble an ellipse and therefore this curve can be fitted to all the cross sections in order to calculate their area. Other geometrical parameters are obtained such as width (difference between the largest and the smallest x values, oriented horizontally) and height (difference between the largest and the smallest y values, oriented vertically).

From figure 5.3 it can be noticed that the conduits can intersect each other creating nodes. The shape analysis of

these complex features is beyond the scope of this thesis project, thus cross sections are extracted only along the segments between nodes every 1.5 meters. When a conduit is intersected by multiple nodes, in correspondence of these no data will be extracted, instead the measurements will be executed on the conduit's segments separately and eventually grouped in order to represent the hole conduit. In addition, since conduits that are too short are considered not useful for the extraction of information, all the conduits shorter than 2 meters have not been taken into account. As can be seen also from figure 5.3 not all the extremities of the skeletons coincide perfectly with the ends of the conduits and this is due to the fact that in these areas the point clouds tend to be noisy and not very dense of points. Measurements at the edges would therefore give inaccurate results, which is why it was decided to stop the extension of the skeletons whenever the results started to be unsatisfactory. Anyway the measurements of the total length of the conduits takes into account their whole length observable from the point cloud and not only the skeleton.

A more detailed explanation of the mechanisms that brings to the extraction of the cross sections trough the software Matlab is necessary in order to understand the process step by step. A flowchart of the code is also available at figure 5.4, while in the appendix-C other flowcharts are displayed to ease the understanding of the functions that are within the code.
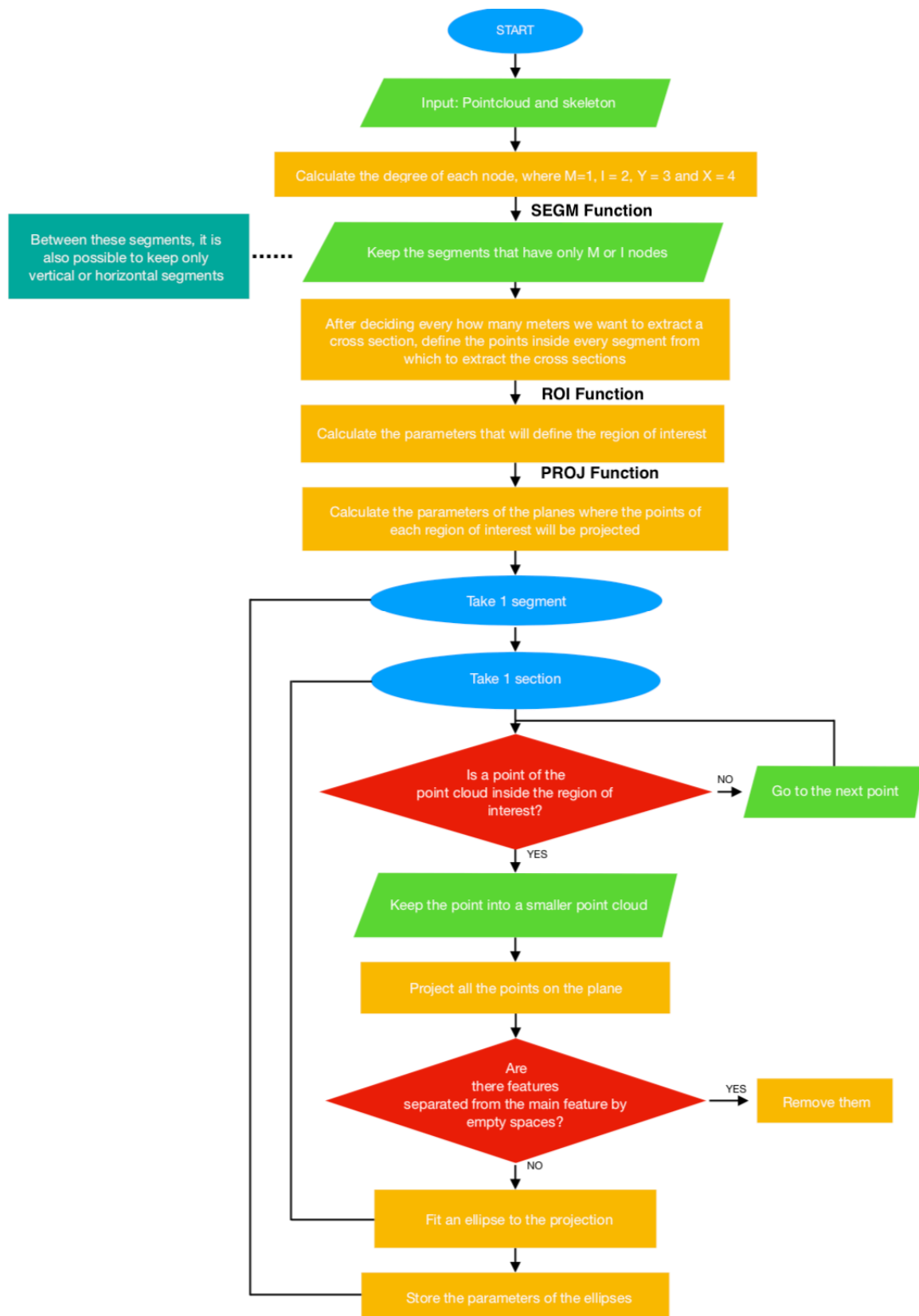
Figure 5.4: Visual representation of how the code works through a flowchart. Blue ovals represent the starting points of the code or of a loop, green parallelograms represent inputs, yellow rectangles represent actions or calculations, red rhombuses represent decisions and turquoise rectangles represent alternatives.

Once a denoised point cloud of a cave and the shapefile of its skeleton are available, we can load them into Matlab. After this a detection of the degree of the skeleton's nodes is applied. These don't have to be confused with the cave's intersection nodes, instead they have to be understood as features corresponding to the linkages of the segments deriving from the drawing of the skeleton. These are divided into M, I, Y, X nodes, where an M node is defined as a node where only one segment ends, an I node as a node between two segments, an Y node as a node where three segments meet and an X node is defined as a node where four segments meet. An example of nodes and their degrees is given at figure 5.5.
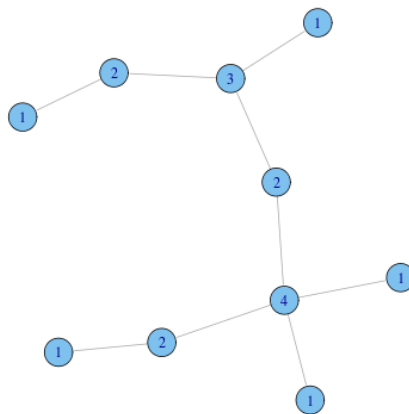
Figure 5.5: This figure shows various nodes and their degrees, represented by the numbers within the nodes.

Considering that we are interested into studying the shape of the conduits and not the shape of the nodes where the conduits meet, only the segments linked by I-M and I-I nodes are studied. 3rd and 4th degree nodes are usually sorrounded by 2nd degree nodes because in this way, since only I-I and I-M linked segments are taken into account, the cross sections extraction is limited to the conduits.

After defining which are the segments we are interested in, within each of these we define equally spaced points (we decided to take 1.5 meters as the distance) from which to extract the cross sections. Regions of interest around each of these points define the xyz coordinates of the slices of point cloud from which the shape of the conduits will be studied. For each of the points, the regions of interest are oriented perpendicular to the segment where the point is, with a thickness of 20 cm in the parallel direction and a width of 80 meters in the perpendicular direction. Planes and their orientations are defined, where all the points of each slice will be projected on. This will give 2D representations of the conduits' shapes, that are easier to study respect to the original 3D slices. A loop is therefore started, working in the following way. For the first point defined within the first segment, the coordinates of its region of interest are taken and every point of the point cloud is checked whether it is inside or outside the region of interest. In the case in which a point falls inside this region, it is kept into a smaller point cloud (slice), otherwise it is skipped. Once the slice is obtained, all the points are projected on the corresponding plane. At this point, on the 2D representation of the conduit's shape just obtained, there is a verification whether the feature centred on the skeleton's segment is separated from other features by empty spaces. If that is the case, the other uncentered features are removed since they would represent other conduits on the sides. An explanatory example is given at figure 5.6. The presence of unwanted features can occur also along the vertical dimension, however the removal of these has been considered too challenging and time consuming in order to be implemented during this project. Therefore the height measurements may have more inaccuracies than the width ones. This may give an idea of how this code can be improved in the future.
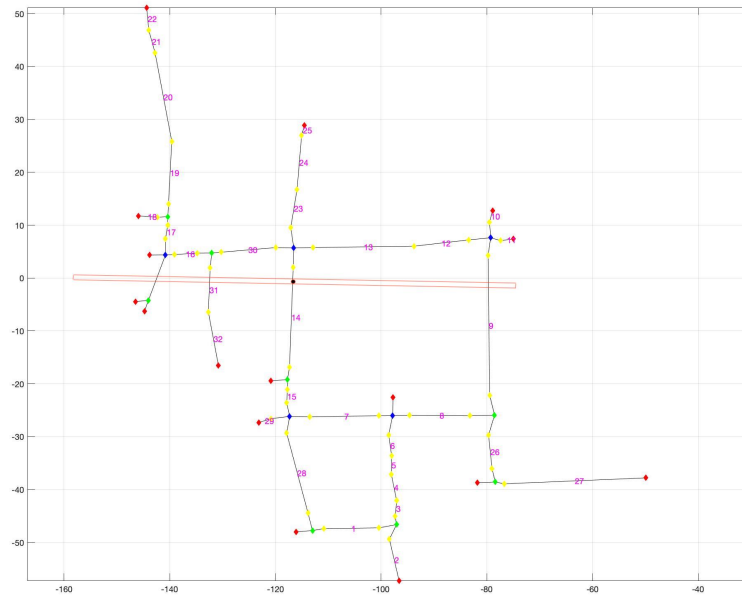
Figure 5.6: This image gives an example of the uncentered features removal. For instance if we imagine that the black spot on the segment number 14 is the point from where the cross section will be extracted and that the red rectangle is the region of interest related to it (imagine a z- value range as well), it is clear that not only the conduit centered on the segment 14 will be taken into account, but the ones centered on the segments 9, 31 and on the one to the left of 31 will be taken into account as well. This is why these features need to be removed.

Once the isolated centered cross section is obtained, an ellipse can be fitted to it and the parameters stored. The error on the ellipse interpolation is calculated and normalized by the number of points, which doesn't give the perfect measure of the deviation, but at least allows to clearly spot badly interpolated ellipses, that usually have interpolation errors greater than values of 0.1-0.15. The process goes on by repeating the same steps for all the points within a segment and, once this is done, another segment is examined till when all the selected segments are studied. The final results can have very different shapes and dimensions as from appendix-B, but an example from Lapinha cave is given at figure 5.7.
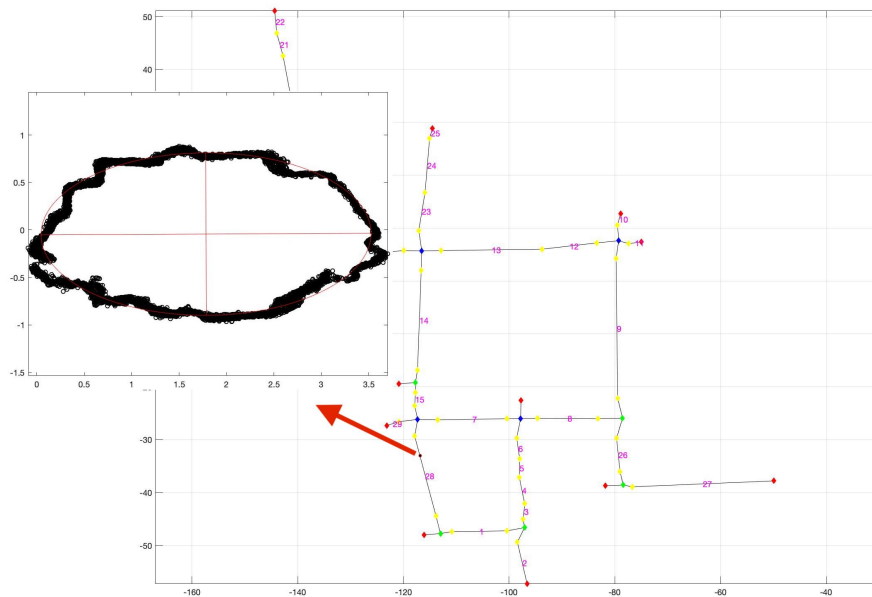
Figure 5.7: Example of a cross section extracted from segment 28 of Lapinha's skeleton. The interpolated ellipse can be seen.

5.5. Conduits Orientation

The dissolutional development of the karst systems is influenced by the study area's structural geology. By visually representing the conduits orientation, it is possible to gather information about the stress fields of the area, since the distribution of fracture systems influences the direction of conduits development. The software QGIS allows to generate rose diagrams by using the caves' skeletons as inputs (Tveite, 2015–2020), which are the most suitable diagrams for this purpose. Here the length of the sectors are weighted to the length of the skeleton's segments and therefore they nicely show the circular distribution of the main development directions.

5.6. Volumetric Calculations

The ellipses interpolated on the cross sections can be used to approximate the cross sectional area of a conduit. If this is combined with the distance between the points where the sections are extracted along a whole conduit, its volume can be calculated with some approximations. For instance when a node is encountered it is assumed that the conduit's shape does not vary through it, as well as for the volume calculation of the part between the extremity of the skeleton and the real end of the conduit, the average of the conduit area is used for the calculation.

When using CloudCompare there is the possibility to show a box that contains the loaded point cloud. The box's dimensions are based on the maximum and minimum x, y and z values, thus a calculation of its volume gives the volume of bedrock interested by the karst system development. By comparing this volume with the cave's volume, it is possible to obtain the dissolved percentage in the subsurface. The rose diagrams previously generated give the main directions of cave development, hence these directions can be combined with the volumetric calculations in order to verify the main patterns of bedrock dissolution. The data obtained in this way can be used in order to formulate hypothesis about the presence in the subsurface of preferential flow directions.

5.7. Mechanical Stability Analysis

The presence of cave systems at depths even greater than 5 kilometers leads us to wonder how these features can maintain stability even if subjected to large lithostatic pressures (0.01~0.2 GPa). If we assume that the deep

karst systems are filled with flowing fluid generating internal pressure, the stability will not depend only on the lithostatic pressure, but on the effective pressure calculated as:

$$P_{eff} = P_{lit} - P_f \tag{5.1}$$

where P_{eff} represents the effective pressure, P_{lit} the lithostatic pressure and P_f the fluid pressure.

The pressure here defined develops stress (σ) acting on the walls of the caves, that consequently generates strain (ϵ) influencing their stability. The most used mathematical model for studying the mechanical deformation of solid objects is linear elasticity, which is based on the assumptions that within the body infinitesimal strains are present and that a linear relationship between stress and strain holds. In a 3D body the stress tensor can be defined as a function of the strain tensor as following:

$$\sigma = 2\mu\epsilon + \lambda tr(\epsilon)I \tag{5.2}$$

where σ represents the stress tensor, μ and λ represents respectively the second and the first Lame's parameters, ϵ represents the strain tensor, tr represents the trace operator and I represents the identity tensor.

The equations involved in this mathematical model are solved through the Finite Element Method (FEM), in particular thanks to the free license software FEniCS Project that enables the automated solution of Partial Differential Equations (PDEs). Before being able to simulate the mechanical stability into FEniCS, a 3D tetrahedral mesh representing the cave system and its surrounding rock matrix has to be generated. For this purpose Lapa do Bode cave was chosen, since it is the one that shows the most developed maze pattern. In order to recreate the conditions that are found in the subsurface, the rock matrix should surround the cave from the sides, the top and the bottom, however the modeling of this configuration fails during the simulation, forcing us to simulate the situation in which the rock matrix is only on the sides in the horizontal direction. This limitation obviously drive the results away from realistic ones. The more realistic model however can still be constructed and used as a starting point for future simulations, therefore the steps needed to generate it are presented. The first step is to obtain an .stl file representing a triangular mesh of the cave through the free license software MeshLab (figure 5.8) and then by loading this into the free license software Gmsh.

When in Gmsh a .geo file can be created where it is possible to build the bounding box that will represent the rock matrix. Since in FEniCS it is necessary to assign different properties to the cave itself and to the surrounding matrix, within the .geo file two different volumes have to be defined, which in this case are called 'cave' and 'matrix'. Another necessary step is to determine the boundaries of the mesh, in order to apply boundary conditions later in FEniCS. Six boundaries are specified, being 'bottom', 'top', 'west', 'east', 'north' and 'south' and through the command '3D' the tetrahedral mesh is finally obtained (figure 5.9) and saved as an .msh file.
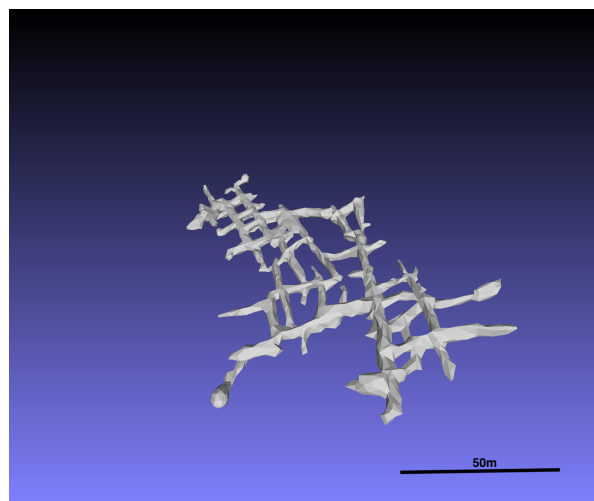
Figure 5.8: This figure shows the triangular mesh of Lapa do Bode cave obtained in MeshLab.
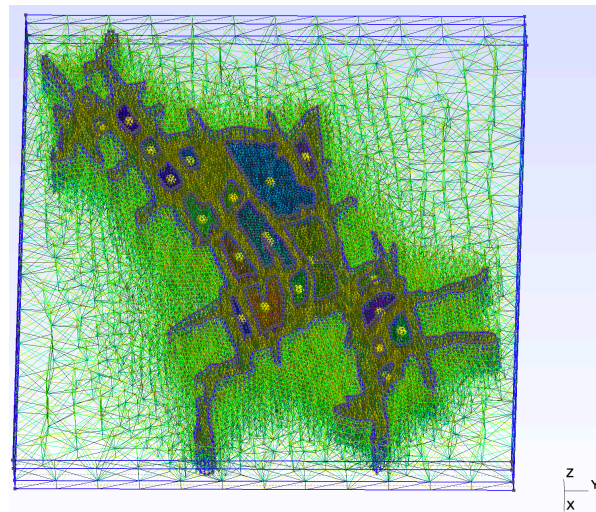
Figure 5.9: This figure shows the tetrahedral mesh of Lapa do Bode cave placed in between a top and a bottom layer, obtained in Gmsh.

As it can be seen from the figure, the middle layer where the cave is placed is in between of a top and a bottom layer. In these two layers, even if the cave's contours can be distinguished, the volumes related to the shape of the cave are grouped into the 'matrix' volume in order to represent the overlying and underlying rock layers.
As already mentioned the simulation of this model doesn't succeed, therefore the model at figure 5.10 is used in order to obtain results.
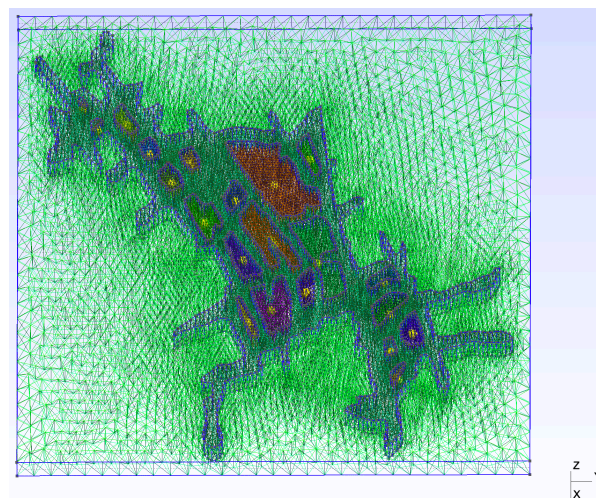
Figure 5.10: This figure shows the tetrahedral mesh of Lapa do Bode cave obtained in Gmsh.

Once the .msh file is acquired it is possible to convert it into three different .xml files to be loaded into FEniCS in order to run the mechanical stability simulations. It is important to explain that the 'cave' volume previously defined is considered as a void space in the simulation, while for the 'matrix' volume two parameters have to be defined, being ν and E. These represent respectively the Poisson's ratio and the Young's modulus of the rock matrix, which define the Lame's parameters as:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{5.3}$$

and

$$\mu = \frac{E}{2(1+\nu)}. \tag{5.4}$$

Likely ν and E values for carbonates are respectively 0.3 and 6 GPa (Madhubabu et al., 2016).

For this first mechanical stability analysis, the boundary conditions applied assign a value of λ and μ to the cave volume and a value of $\frac{\lambda}{1000}$ and null μ to the matrix volume. In addition the stress is always applied vertically on the cave and the most eastern and western sides of the box cannot accommodate deformation in the y-direction, while the other sides are free to move. These boundary conditions don't recreate perfectly plausible natural conditions and should therefore be changed for future simulations.

6

Results

The code for the cross sections extraction was run in Matlab for Lapinha, Lapa do Bode and Marota caves. The results obtained show that conduits dimensions, especially width, stay within ranges that differ for each conduit. That is why dimensional data will not be presented together indiscriminately, on the contrary they will be grouped into different series correspondent to different conduits. Within this section the most meaningful sets of data are displayed on plots, while plots containing all of the data can be found at appendix-E. Another interesting aspect deducible from the results is the relationship between the orientation of the conduits and the orientation of the joints observed in the areas, as well as a preferential dissolutional direction emerging from the volumeteric calculations.

6.1. Computationl Times and Accuracy

The results and the computational times vary between the three caves because of the different sizes and the different environments of formation. On table 6.1 the computational times, the total number of segments, the number of segments investigated and the number of conduits is given.

Cave's Name	Total Segments	Studied Segments	Number of Conduits	Computational Time
Lapinha	75	32	9	247s
Lapa do Bode	322	97	40	532s
Marota	199	69	19	1109s

Table 6.1: Computational times required by the code to run and number of segments and conduits of each cave.

Before presenting the results, it must be specified that the quality of these is related to the assumptions that were made and to the quality of the point clouds which, even after denoising processes, present noise in the data. Therefore errors in the measurements of width and height of conduits occur, as well as inaccuracies in the volumetric measures since an ellipse is not the perfect interpolation of every cross section. Within the code a variable named 'error_interpolation' gives the interpolation error of the ellipse normalized by the number of points. In order to visualize these inaccuracies, at figure 6.1 there is an example of a good interpolation in a non-noisy cross section and other less accurate cases. The examples come from the Lapa do Bode cave and their position is also highlighted. In order to show the quality of the cross section extraction, at figure 6.2 the same slices manually extracted with CloudCompare are displayed in order to compare the automatic method with the manual method used in some of last year's master thesis projects.
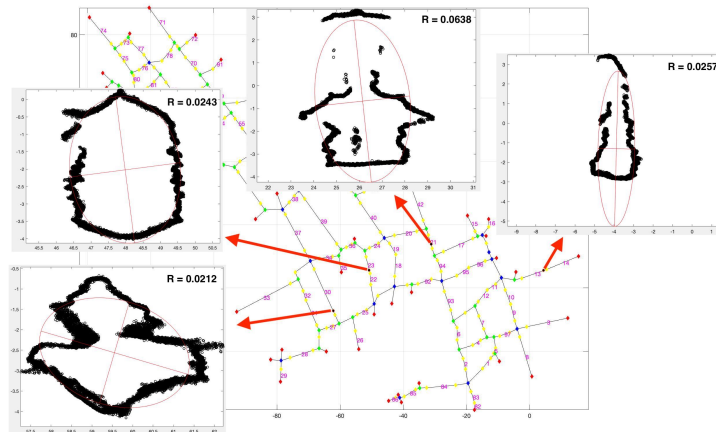
Figure 6.1: This figure shows 4 cross sections extraction examples from the Lapa do Bode cave. The cross section extracted from segment 23 shows a good interpolated ellipse, while the ones from segments 13 and 30 are not perfectly approximated by the ellipses. The example at segment 21 shows a noisy cross section. R = interpolation error.
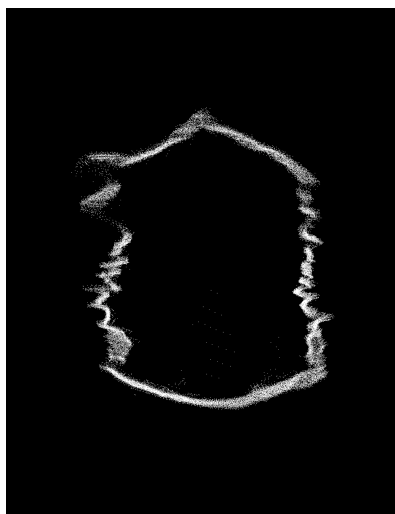
(a) Segment 23 cross section.
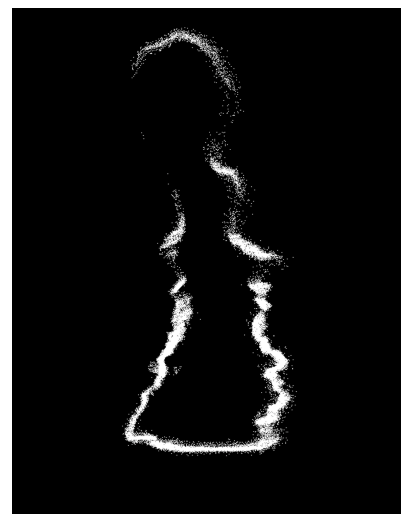
(b) Segment 13 cross section.
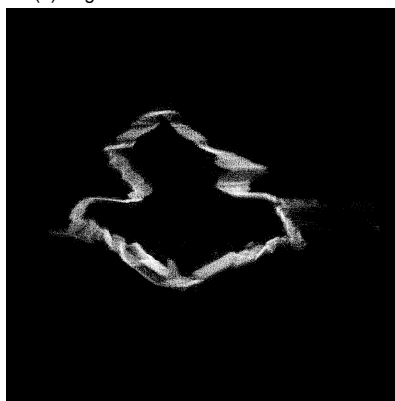
(c) Segment 20 cross section.
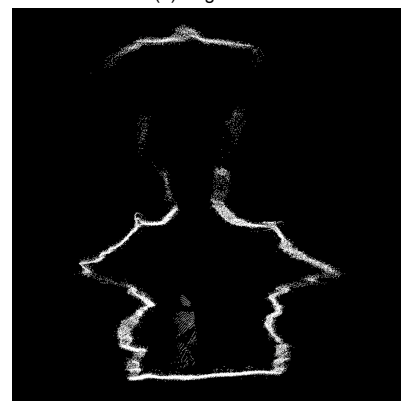
(d) Segment 21 cross section.

Figure 6.2: Cross sections manually extracted from CloudCompare correspondent to the ones at figure 6.1.

Another problem in the dimensions acquisition could lie on how width and height are interpreted. In fact instead of the differences between the maximum and the minimum x and y values, which is a good approximation in most

of the cases, width and height may be seen as the maximum distances found between the two walls respectively in the horizontal and in the vertical direction. This applies especially in the cases in which a cross section can be interpolated by an obliquely oriented ellipse (appendix B). In addition, the sediment depositing on the floor of the conduits in many cases alter the real visible height by filling the bottom part (Farrant and Smart, 2011), therefore this must be taken into account when observing the values and the distribution of the height measurements.

6.2. Lapinha

Lapinha cave, whose name can be translated from portoguese as "Small Cave", is a water-table maze cave that formed in phreatic conditions. Two broad anticlines were recognized at its location: a small one with the axis oriented N-S and a larger one with the axis oriented E-W. Along the axis of these anticlines, fracture corridors have been observed. From a plan view 10 conduits can be identified (figure 6.3), however the most eastern conduit can be ignored since its shapes are derived from a ceiling collapse and not by water flow alone. At first glance a main N-S E-W orientation of the conduits is already visible.
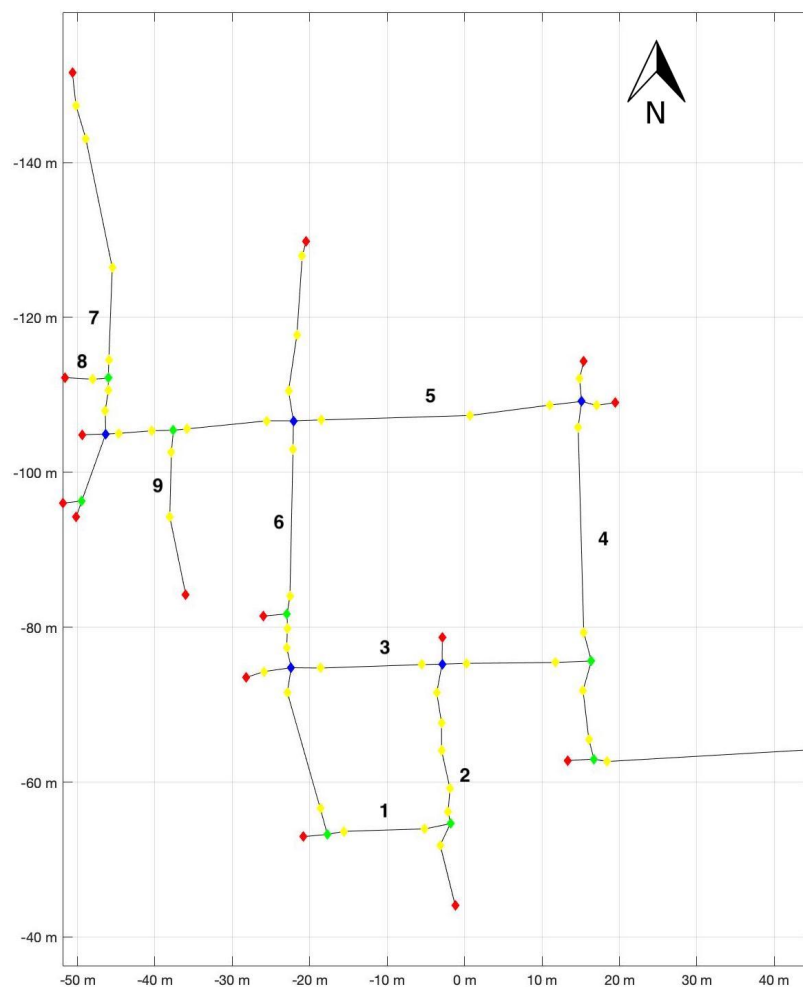
Figure 6.3: Lapinha's skeleton showing the numbers assigned to the different conduits.

6.2.1. Conduits Orientation

The skeleton that was obtained at figure 5.3b can be used to obtain a rose diagram that shows the main orientations of the conduits, as at figure 6.4. From the 75 segments that build up the skeleton, the resulting main orientations are NNW-SSE (32%) and ENE-WSW (40%).
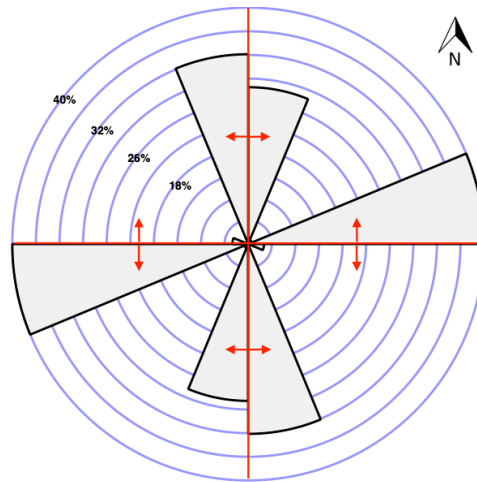
Figure 6.4: Rose diagram generated from Lapinha's skeleton. The orientation of the anticlines' axes is shown in red.

Folds are usually associated with a compressional tectonic phase, however in order to accomodate the deformation, the layers that bend can generate tensional structures such as joints. These usually form in the outer part of the arc described by a fold with an orientation parallel to that of the fold's axis. Since the main orientation of the conduits is NNW-SSE and WSW-ENE and the axis of the folds are oriented N-S and E-W, we can assume that the conduits follow the main directions of development of joints.

6.2.2. Dimensions Analysis

Lapinha's dimensional data were extracted from 171 different cross sections along 9 conduits. Width and height are measured from WSW towards ENE and from NNW towards SSE (figures 6.5 and 6.6) in order to visualize the conduits' dimensions along the two main directions of development of the fracture corridors.

Figure 6.5: WSW-ENE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Lapinha's conduits are listed.
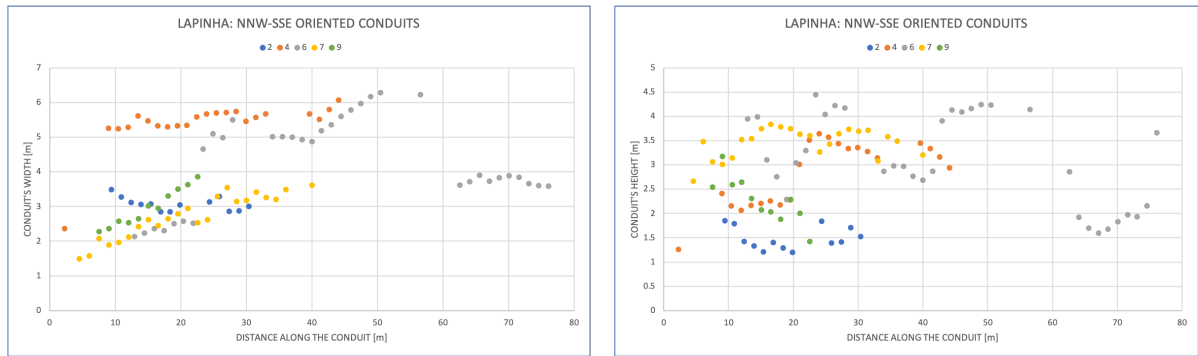
Figure 6.6: NNW-SSE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Lapinha's conduits are listed.

From the two couples of plots one of the first observations that can be made is that conduit's dimensions are not always similar to each other, but sometimes differ of several meters. By observing the single conduits separately, it is possible to recognize that dimensions lie within constant ranges often interrupted by increases or decreases in the values. These changes can be gradual, as for the case of conduit 7 for instance, or improvise as for conduit 3. It is worth to notice that abrupt changes often happen after that nodes are encountered, which can be identified by intervals in which data are absent.

The distribution of the sizes of the cross sections is highlighted at figure 6.7, where all the interpolated ellipses are plotted along Lapinha's skeleton together with a visual representation through colours of conduits' width previously plotted.
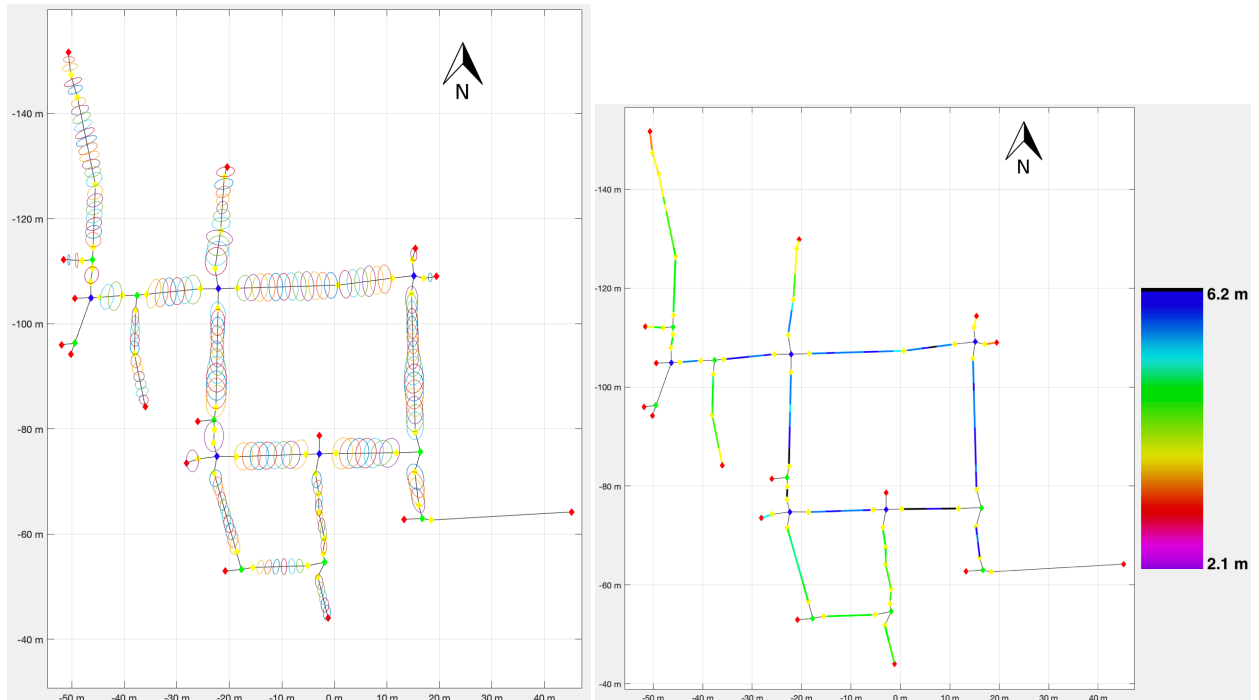
Figure 6.7: Lapinha caves's skeleton showing the ellipses interpolated on the cross sections(left) and the widths values visualized through a colour scale(right).

From figure 6.7 it looks even more clear how the conduits have different dimensions between each other and that within a single conduit the width range can stay constant, as well as vary gradually or abruptly after a node.

In addition a zone with bigger dimensions can be identified at the center of the charts, more precisely for values between -30 and 20 on the horizontal axis and values between -110 and -70 on the vertical axis.

6.2.3. Volumetric Calculations

Lapinha cave's volume is 3513 m^3, with 2068 m^3 extended in the NNW-SSE direction and 1445 m^3 in the ENE-WSW direction. The box obtained through CloudCompare (figure 6.8) has a volume of 71688 m^3, thus only the 4.90% of the total bedrock volume is dissolved, confirming the fact that just a minimal part of the fractures present in the subsurface eventually develop into cave passages. Since a 2.88% is dissolved in the NNW-SSE direction and a 2.02% in the ENE-WSW direction, the first can be considered the preferential direction of dissolution in this area.
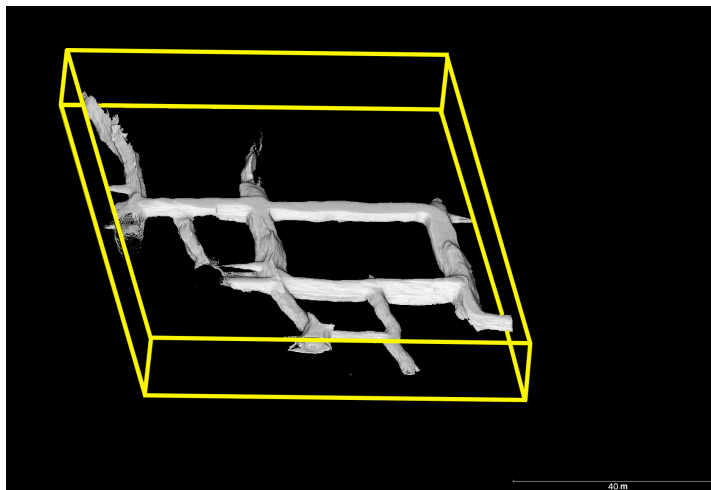
Figure 6.8: Lapinha's bounding box obtained from CloudCompare.

6.3. Marota

Marota cave is for some aspects similar to Lapinha. It is in fact a water-table maze cave that owes its formation to water flowing horizontally in phreatic conditions, as well as it presents fracture corridors locally related to anticlines, along which the conduits develop. 19 conduits form Marota cave (figure 6.9), which look for the most part oriented N-S and E-W.
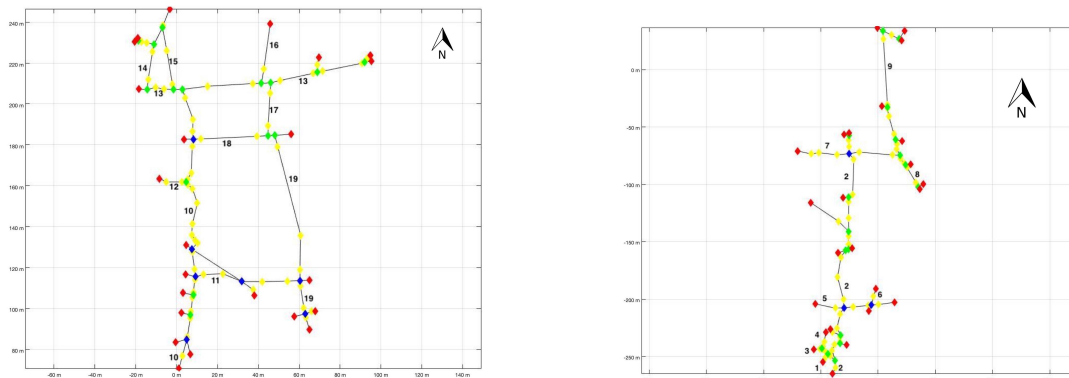
Figure 6.9: Marota's skeleton (northern part on the left and southern part on the right) showing the numbers assigned to the different conduits.

6.3.1. Conduits Orientation

Through the skeleton obtained at figure 5.3a the rose diagram shown at figure 6.10 can be built. The main orientations that emerge, extracted from 199 segments, are NNW-SSE (25%) and ENE-WSW (24%), which reflect the orientations of the fracture corridors observed inside the cave.
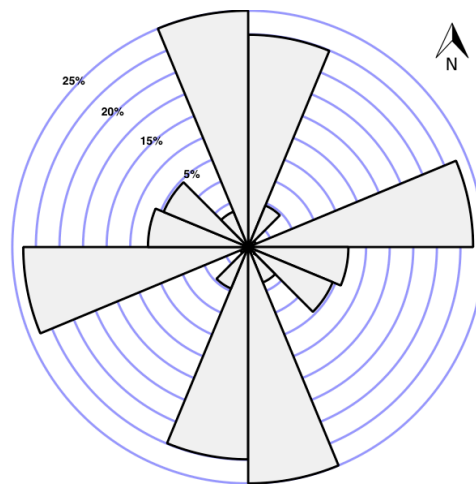
Figure 6.10: Rose diagram generated from Marota's skeleton.

6.3.2. Dimensions Analysis

The dimensional data obtained for Marota cave were acquired from 496 cross sections along 19 conduits. Width and height are measured from NNW towards SSE and from WSW towards ENE (figures 6.11 and 6.12).

Figure 6.11: NNW-SSE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Marota's conduits are listed.
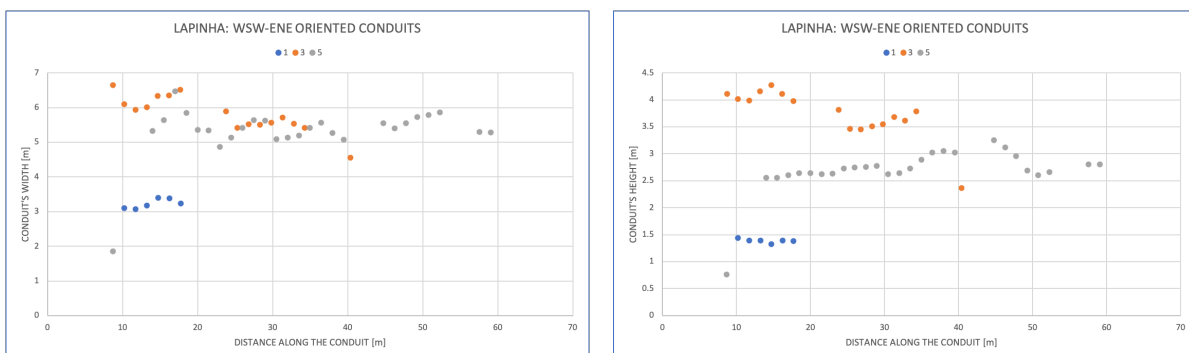
Figure 6.12: WSW-ENE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Marota's conduits are listed.

Marota cave's conduits are much longer than Lapinha's ones, in fact a bigger dataset is available. Even if the data coming from the NNW-SSE oriented passages look noisy, it is still possible to make some observations. Firstly it can be noticed that different conduits have often different dimensions ranges. In addition the longest conduits show us nicely that within a conduit several changes from rising dimensions to decreasing ones occur, and these often coincide with empty intervals in the datasets, i.e. cave's nodes.

At figures 6.13 and 6.14 it is possible to observe the distribution of the ellipses interpolated on the cross sections and a representation of the conduits' width previously plotted through colours that confirm the behaviours described above.
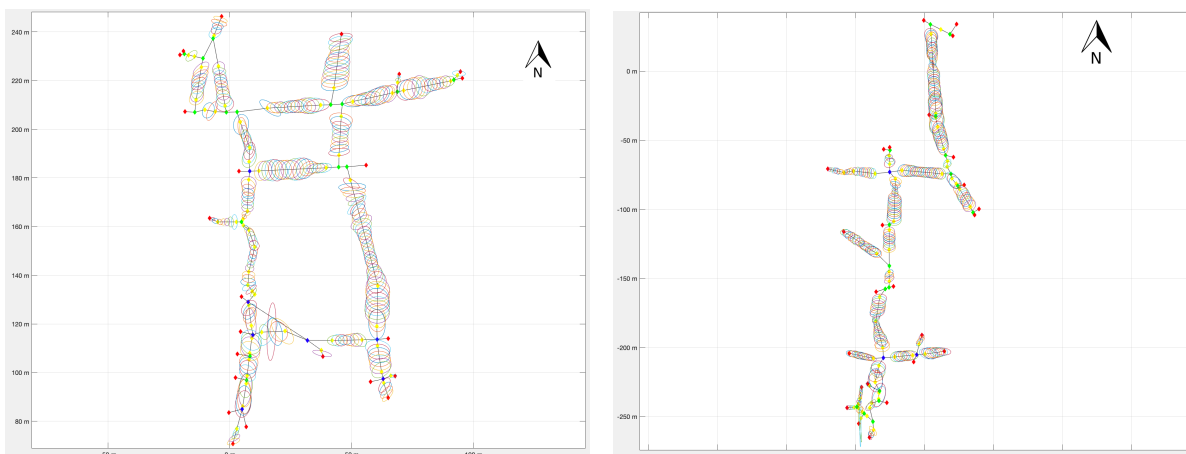
Figure 6.13: Marota's skeleton (northern part on the left and southern part on the right) with the interpolated ellipses plotted.
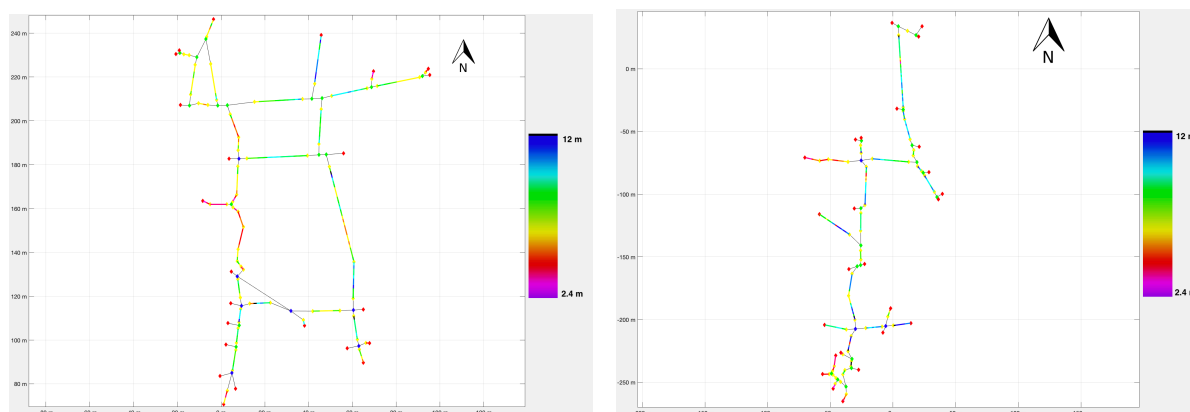
Figure 6.14: Marota's skeleton (northern part on the left and southern part on the right) where the width values are visualized following a colour scale.

The observation of the longest conduits is fundamental in order to notice the dimensions changes of conduits after they are interrupted by nodes. In many cases in fact the conduits maintain fairly constant values of width in the areas between two nodes, while once encountered one of these features the dimensions change.

6.3.3. Volumetric Calculations

The calculated volume for Marota cave is 38916 m^3, with 24432 m^3 elongated in the NNW-SSE direction and 14484 m^3 in the ENE-WSW direction. The boxes that surround Marota's south and north sides (figure 6.15) have a volume of 1517652 m^3, therefore the dissolved percentage is barely the 2.56% of the total bedrock volume, highlighting that caves systems, when present, compose a minimal part of the subsurface. Within this 2.56%, a 1.61% is dissolved towards the NNW-SSE direction and a 0.95% in the ENE-WSW direction, making the first the main direction of dissolution.
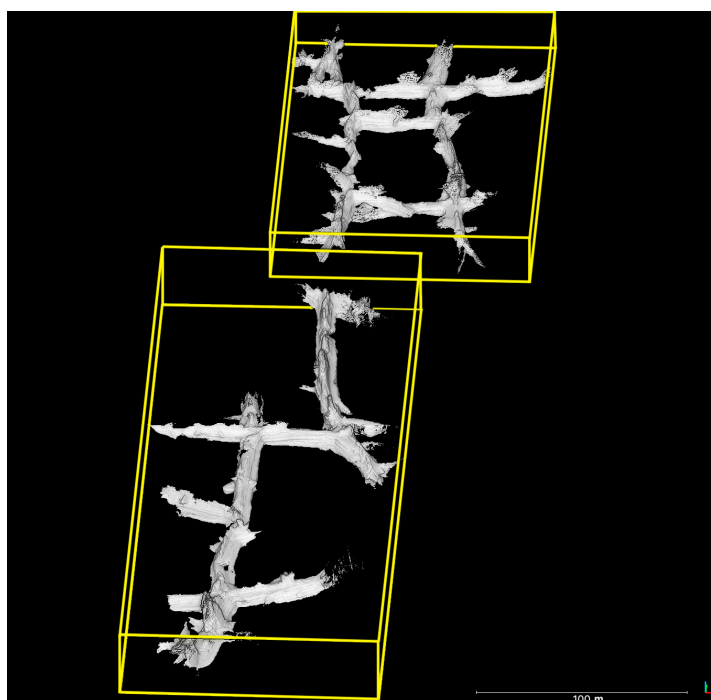
Figure 6.15: Marota cave's bounding boxes obtained through CloudCompare.

6.4. Lapa do Bode

Lapa do Bode, which literally means "Cave of the Goat", is a maze cave whose formation processes are still unclear. The most supported hypothesis sees it as a floodwater maze cave, meaning that its formation is related to the floodings of the close by river. These events occasionally furnish acidic water that flows horizontally in phreatic conditions into the cave. The water source in Lapa do Bode therefore is episodic, while in Marota and Lapinha it is continuous. In addition to this a general dip orientation of the area towards NW was observed. Lapa do Bode, even if smaller in size than Marota, counts 40 conduits with numerous closed loops, making it a nice example of network maze cave. From figure 6.16 at a first sight the conduits show NW-SE and SW-NE orientations.
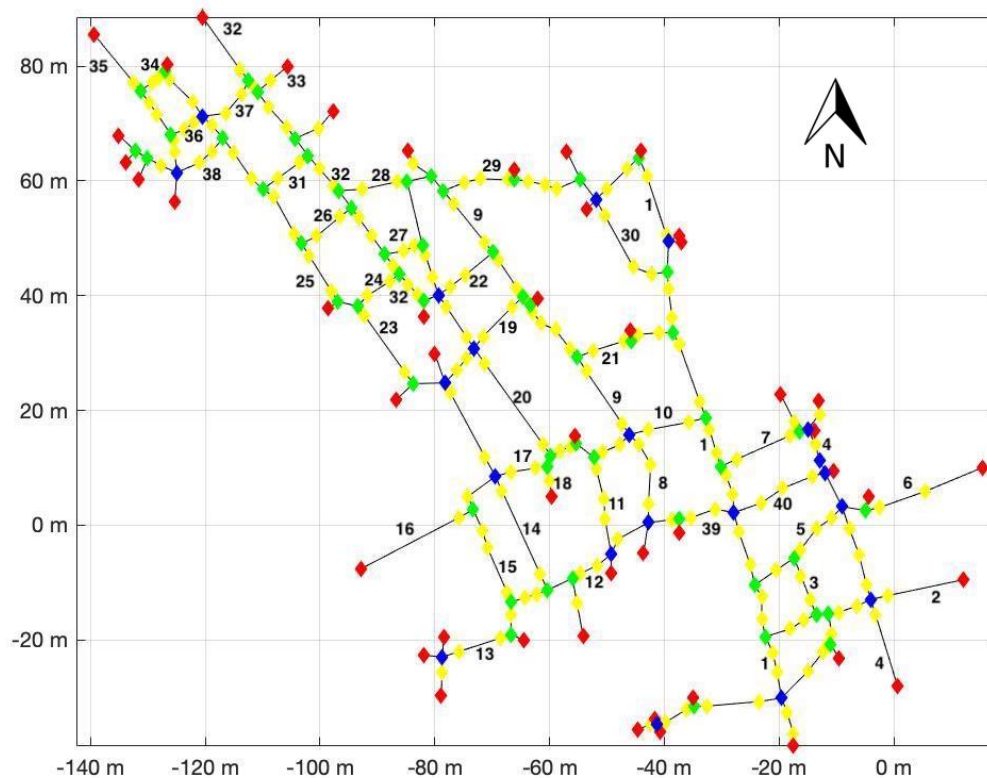
Figure 6.16: Lapa do Bode's skeleton showing the numbers assigned to the different conduits.

6.4.1. Conduits Orientation

From the skeleton showed at figure 5.3c, the rose diagram at figure 6.17 is obtained, showing that the conduits are mainly oriented NW-SE and NE-SW.
As it can be observed form the rose diagram and the dimension of the sectors, the conduits tend to develop more in the NW-SE direction than on the NE-SW one, following the general dip direction.
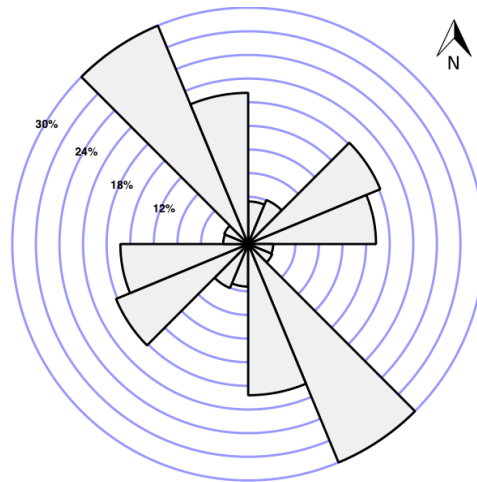
Figure 6.17: Rose diagram generated from Lapa do Bode's skeleton.

6.4.2. Dimensions Analysis

In order to obtain Lapa do Bode's dimensional data 309 cross sections from 40 conduits were analyzed. The plots obtained at figures 6.18 and 6.19 are the results of a dimensional analysis performed in NW-SE and SW-NE directions.
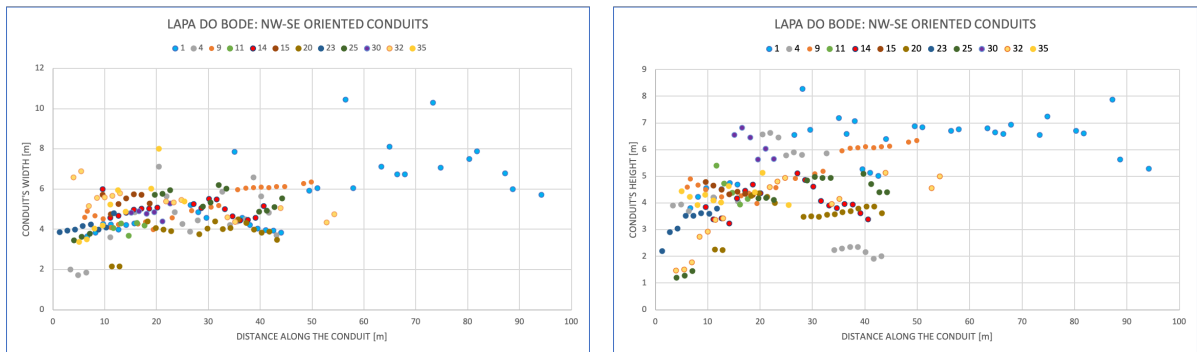
Figure 6.18: NW-SE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Lapa do Bode's conduits are listed.
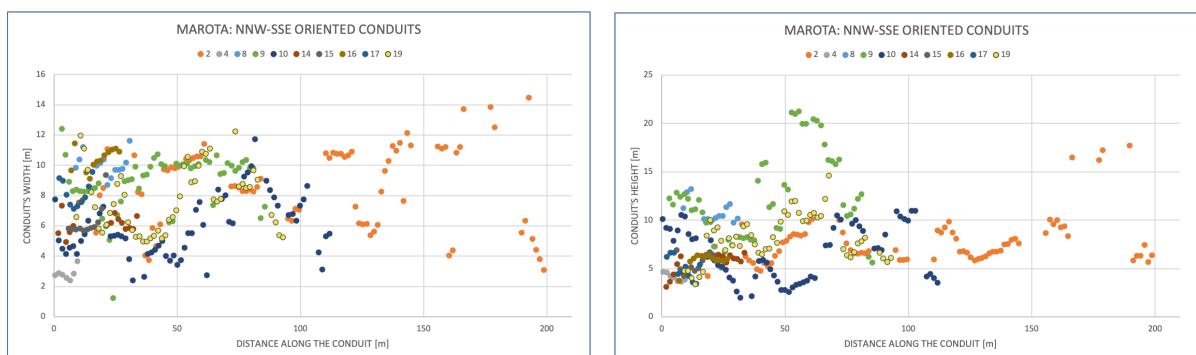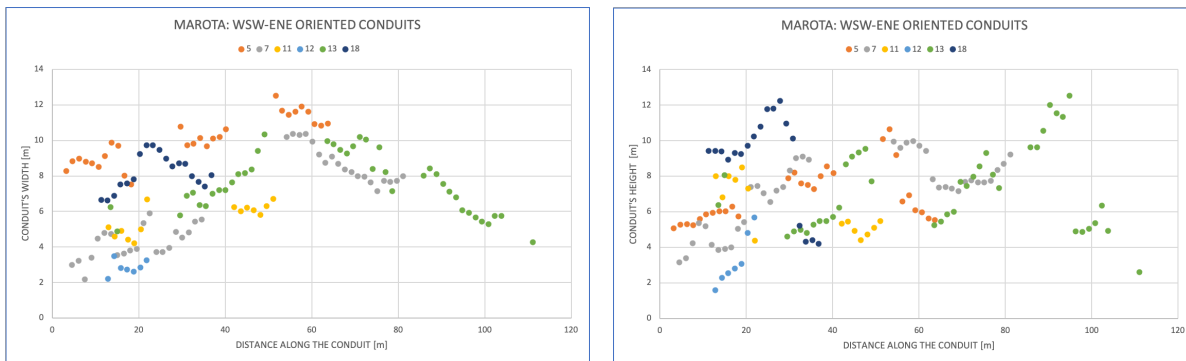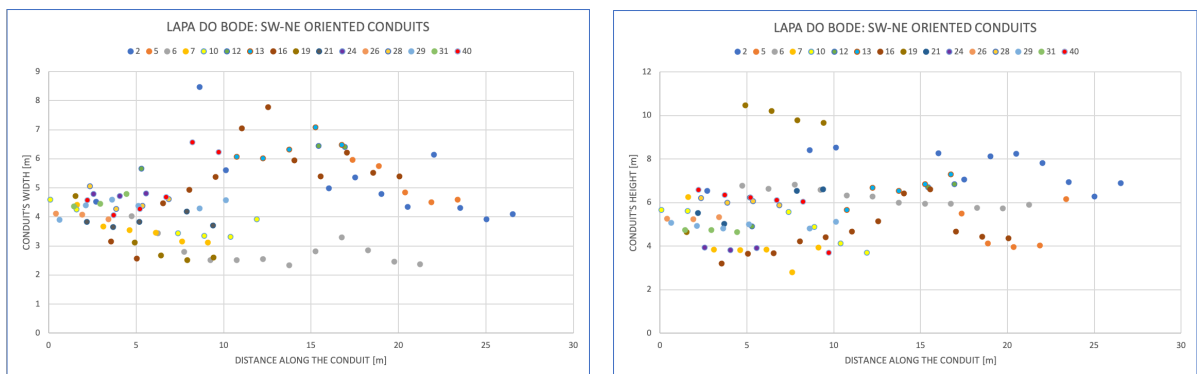

Figure 6.19: SW-NE plots of conduit's width vs distance along the conduit(left) and conduit's height vs distance along the conduit(right). On the legend the numbers of Lapa do Bode's conduits are listed.

Lapa do Bode's number of conduits is large compared to the previous two caves, with lengths comparable to Lapinha's. The maximum values on the horizontal axis of the four plots highlights what was already clear from the rose diagram, which is that NW-SE oriented passages are more developed than those oriented SW-NE. Within the dataset it is still possible to recognize the behaviours already described for Lapinha and Marota's dimensional data. The conduits' in fact can show different dimensional ranges between them, as well as gradual or improvise increases and decreases in width and height.

Since the number of conduits is considerable, the plots may result unclear, which is why at figure 6.20 a visual representation of the conduits' dimensions is given.
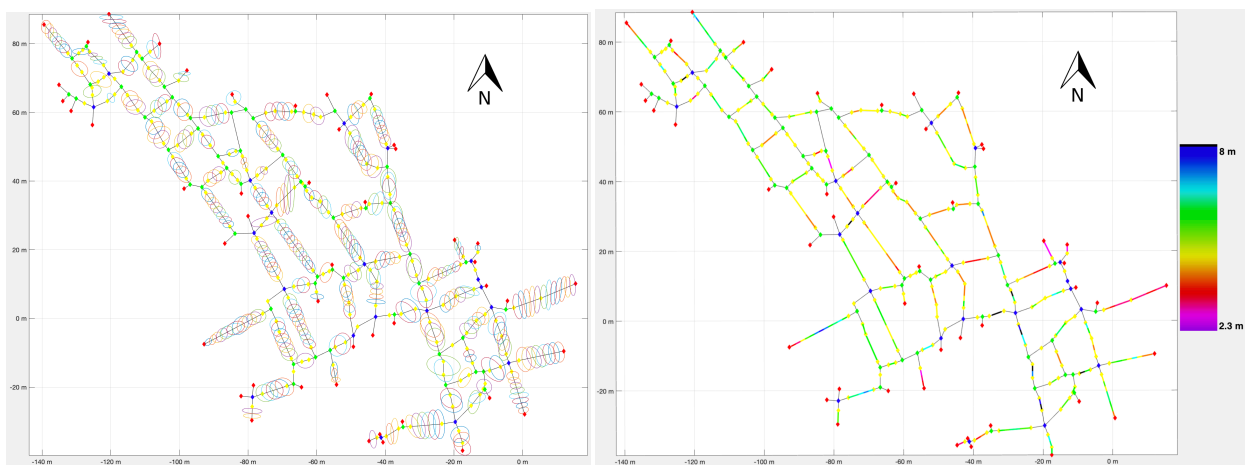
Figure 6.20: Lapa do Bode cave's skeleton showing the ellipses interpolated on the cross sections(left) and the widths' values visualized through a colour scale(right).

It is possible to observe that the NW-SE oriented conduits share similar width values between them on the left part of the graph (around 5 meters), however by considering the side of the graph to the right of the horizontal axis values of -30, several changes and substantial width differences are present. SW-NE oriented conduits show a different behaviour, since their dimensions vary across all of the interested area.

6.4.3. Volumetric Calculations

The total volume of Lapa do Bode cave is 15008 m^3, with 9004 m^3 expanded in the NW-SE direction and 6004 m^3 in the NE-SW direction. The volume of Lapa do Bode's bounding box is 375905 m^3 and therefore the dissolved percentage is just the 3.99%, confirming that not all the joints form cave passages, but preferential dissolutional patterns exist. 2.35% of the volume is dissolved in the NW-SE direction, while 1.64% is dissolved in the NE-SW direction, making the first the preferential direction of dissolutional development.
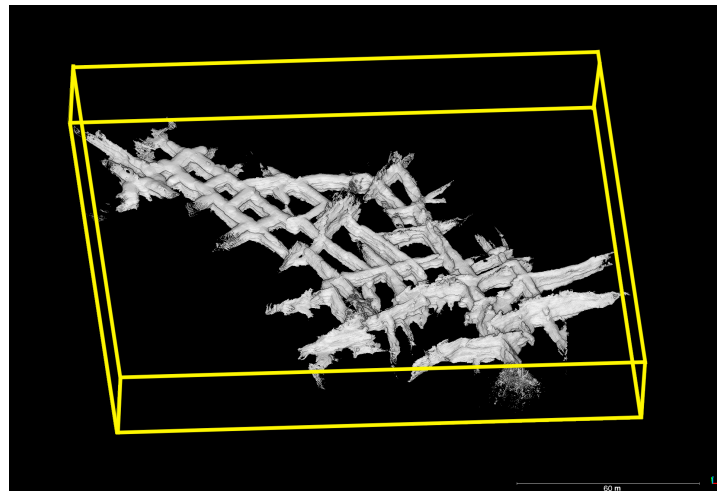
Figure 6.21: Lapa do Bode cave's bounding box.

6.5. Mechanical Stability Analysis

The results of the simulations obtained through FEniCS show how the particle displacement, and therefore the deformation, reacts to changes of the two main influencing parameters previously identified, ν and E. There is the possibility to display the distribution of the stress applied, as well as the distribution of the displacement of the particles (u) over the area of interest. Since the stress applied on Lapa do Bode is the same in every simulation (figure 6.22), the maps of the displacements are shown in order to observe the different mechanical behaviours in the different cases listed at table 6.2.

Simulation n°	E[GPa]	ν
1	6	0.3
2	2	0.3
3	16	0.3
4	6	0.2
5	6	0.4

Table 6.2: List of the simulated cases with the corresponding Young's moduli and Poisson's ratios.
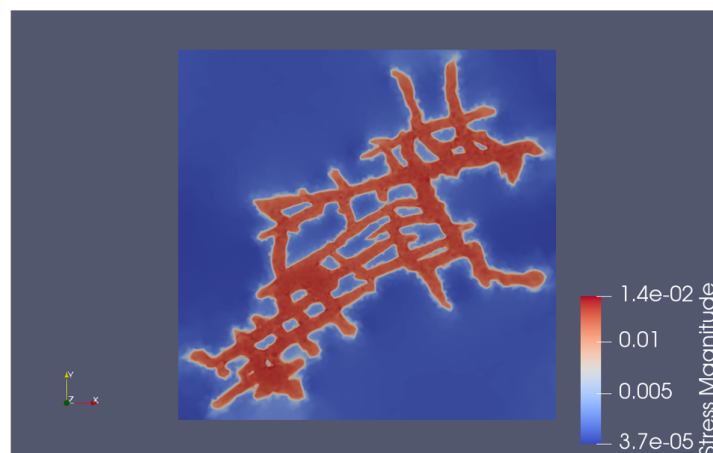
Figure 6.22: Plan view representation of the stress distribution over the area of interest.
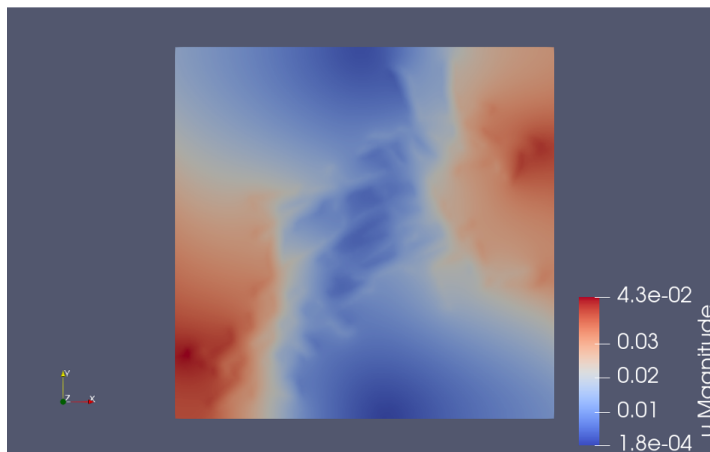
Figure 6.23: Plan view representation of the particles' displacement distribution over the area of interest of case 1.
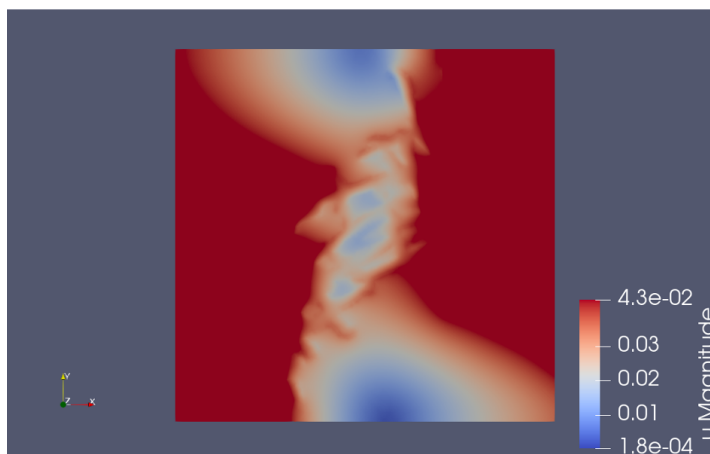
Figure 6.24: Plan view representation of the particles' displacement distribution over the area of interest of case 2.
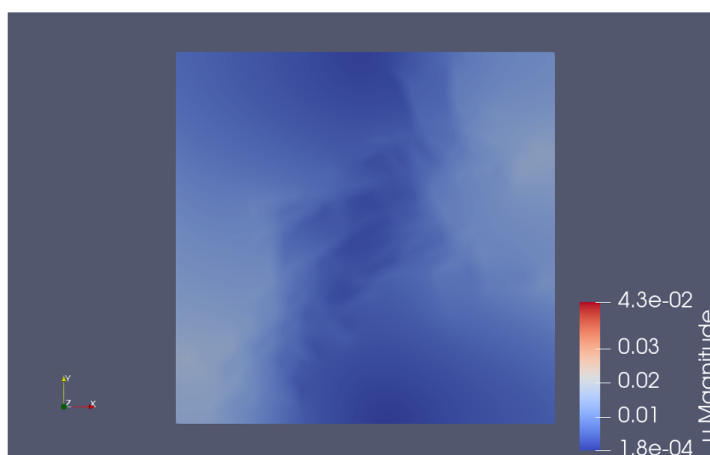
Figure 6.25: Plan view representation of the particles' displacement distribution over the area of interest of case 3.

The first three cases are characterized by the same Poisson's ratio values, but different Young's moduli. Since E is defined as the ratio between the stress and the axial strain, this parameter can be seen as a measurement of the stiffness of the material. In particular high values define a material less keen to axial deformation when subject to stress, respect to a material with low E values.
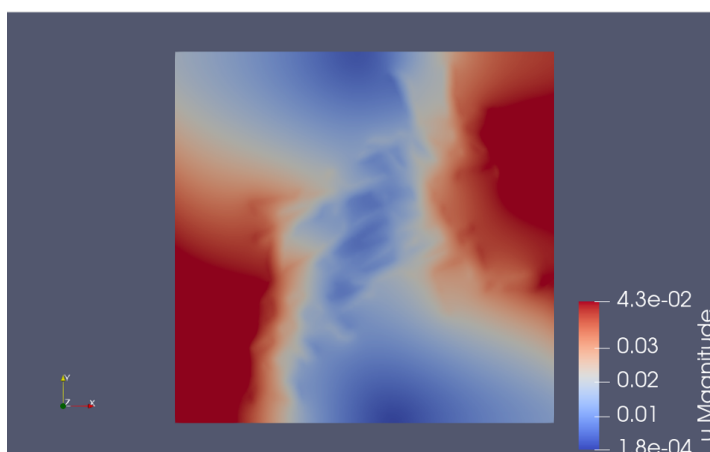
Figure 6.26: Plan view representation of the particles' displacement distribution over the area of interest of case 4.
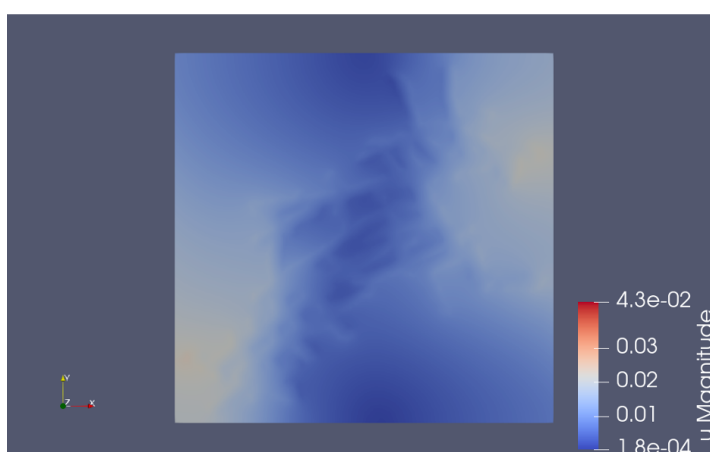
Figure 6.27: Plan view representation of the particles' displacement distribution over the area of interest of case 5.

The last two cases aim to show how materials with different Poisson's ratios react to stress, maintaining the same E value. Poisson's ratio gives a measurement of how much a material expands in the direction transversal to stress, since it is the ratio between the transversal and the axial strain caused by a stress field. With smaller Poisson's ratios therefore the axial deformation is larger.

6.6. Conduits Orientation Discussion

The orientation analysis executed highlights a general NNW-SSE WSW-ENE pattern of direction of development of the conduits (figure 6.28). The combination of this with the information obtained from the structural features present in the study areas, such as N-S and E-W oriented folds, confirms the relationship between regional deformational events and the development of fracture corridors that influence fluid flow patterns in the subsurface and therefore the formation of karst systems (Ennes-Silva et al., 2016).

Figure 6.28: Study area's map with highlighted the positions of the three caves that have been studied. Next to each cave the respective rose diagram is plotted.

The fact that the two main orientations of elongation of the conduits are perpendicular to each other, indicates that the cave systems evolved from orthogonal sets of fractures, in particular joints since in order to allow an initial fluid flow in the subsurface an aperture has to be present. Perpendicular sets of joints can only form if the minimum principal stress (σ_3) rotates of 90°, implying that one of the two fracture sets is created prior to the other one. The rotation occurs in two cases: if the first fracture set causes stress release in its vicinity or if a rotation of the regional or local stress field takes place. Both cases may cause the development of a second fracture set perpendicular to the first one, but with different characteristics. In the first case in fact, the mode I fractures start from a joint of the first set and terminates its extension as soon as another joint is encountered, while in the second case the new fracture set can crosscut the previously formed one (Li and Ji, 2020). Since joints can form in the proximity of folds, this is a further demonstration of the relationship between regional deformation and caves development (N.E et al., 1999). In addition, the recognition of terminating and crosscutting geometries on a plan view can give us information about the chronology of development of the fracture sets. Lapinha and Marota's conduits always crosscut each other, therefore it is not possible to divide the younger from the older set, however Lapa do Bode, in the northwestern part, presents NE-SW oriented conduits that abut on NW-SE oriented conduits. At least in this part of the cave system, it is possible to assume that the NW-SE oriented fracture corridors are older than the NE-SW ones.

6.7. Dimensions Analysis Discussion

The geometry analysis shows the variation of dimensions along the passages. By taking a close look at the width values plotted along the skeletons through colours, it is possible to notice differences in the dimensions between different conduits, as well as it emerges that along the conduits the dimensions often vary after encountering a node. Since the direction of development of cave passages follow the orientation of fracture corridors, higher general dimensions may be the result of the connection between different joints sets. Jourde et al. state that fluid flows are not equally distributed on all the fractures, but focus where fractures intersect each other or bedding planes, channelizing the flow. In addition, due to variations of the local stress fields, pipes can open at the intersection between fractures or fractures and bedding planes, relating the variations of conduits' dimensions to the presence of these intersections, since the more fractures intersect each other, the bigger the dimensions of the pipe are. Simplified graphs presenting conduits belonging to the same cave but with substantial differences in the dimensions are displayed at figure 6.29.

Figure 6.29: Six simplified graphs from each of the caves showing the differences in dimensions that can occur in a karst system. Width and height values are displayed respectively in the left and in the right side of the figure. The legends give the numbers of the conduits taken as examples.

As it can be noticed in these examples, dimensions differences between different conduits span from a couple of meters up to 10 meters in the most extreme cases.

The variation of dimensions within a conduit instead, for which four examples are given at figure 6.30, can be explained if we assume that local changes in the stress field, lack of continuity of fracture sets and connectivity variations within the complete network condition the hydraulical efficiency of a fracture corridor, especially in the vicinity of nodes (Jourde et al., 2002).
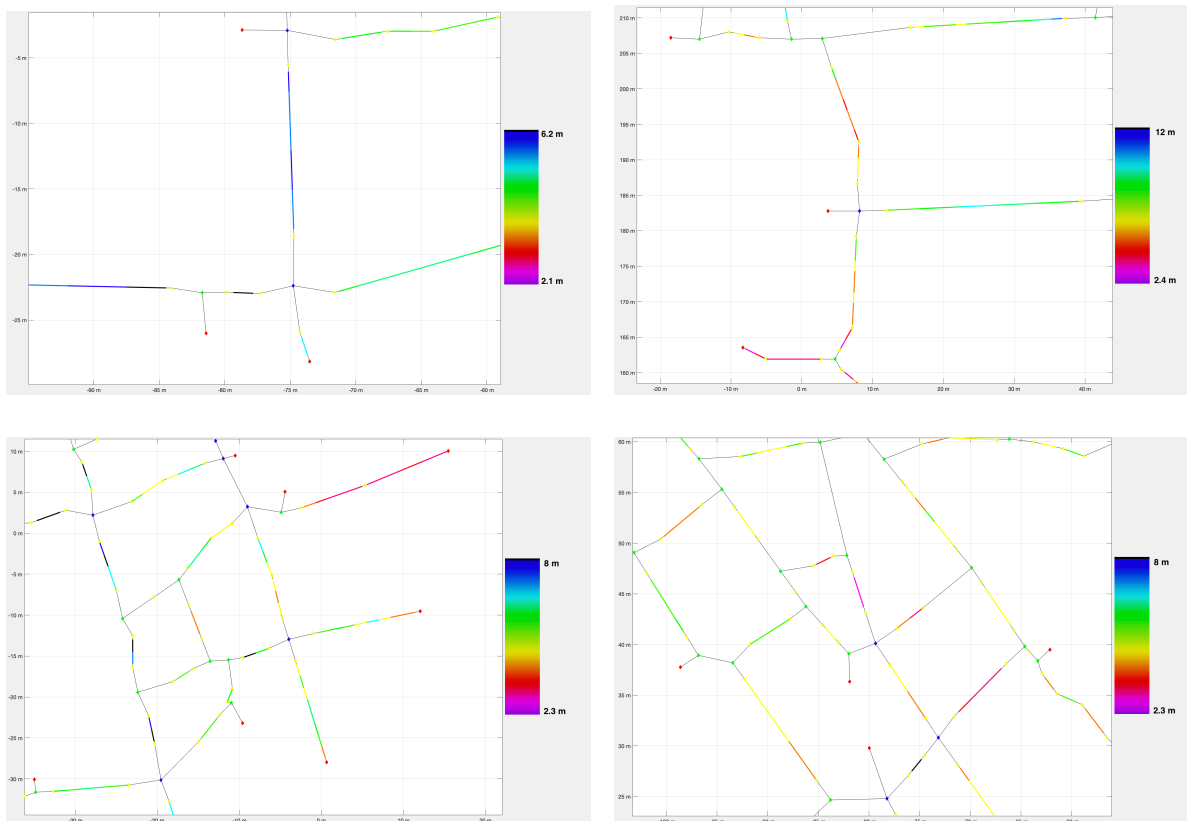
Figure 6.30: This figure shows four examples of conduits' width changes happening in the proximity of nodes. The examples come, in clockwise order from the top left image, from the southern part of Lapinha, the northern part of Marota, the south-eastern part of Lapa do Bode and the central-western part of Lapa do Bode.

In addition to variations of the dimensions within the same conduit or different ones, variations in the shapes of conduits can be observed as well. Different shapes can be found in the appendix B, as well as at figure 6.31, where cross sections along different conduits are shown. The first difference to be expected is between sub-vertical and sub-horizontal shapes, since in the early stages of dissolution when joints of different sets intersect each other a sub-vertical pipe is generated, whereas when a sub-vertical joint intersects a flat bedding plane, sub-horizontal pipes develop (Jourde et al., 2002). If the dissolution continues homogeneously with time, these two are the shapes that have to be expected.
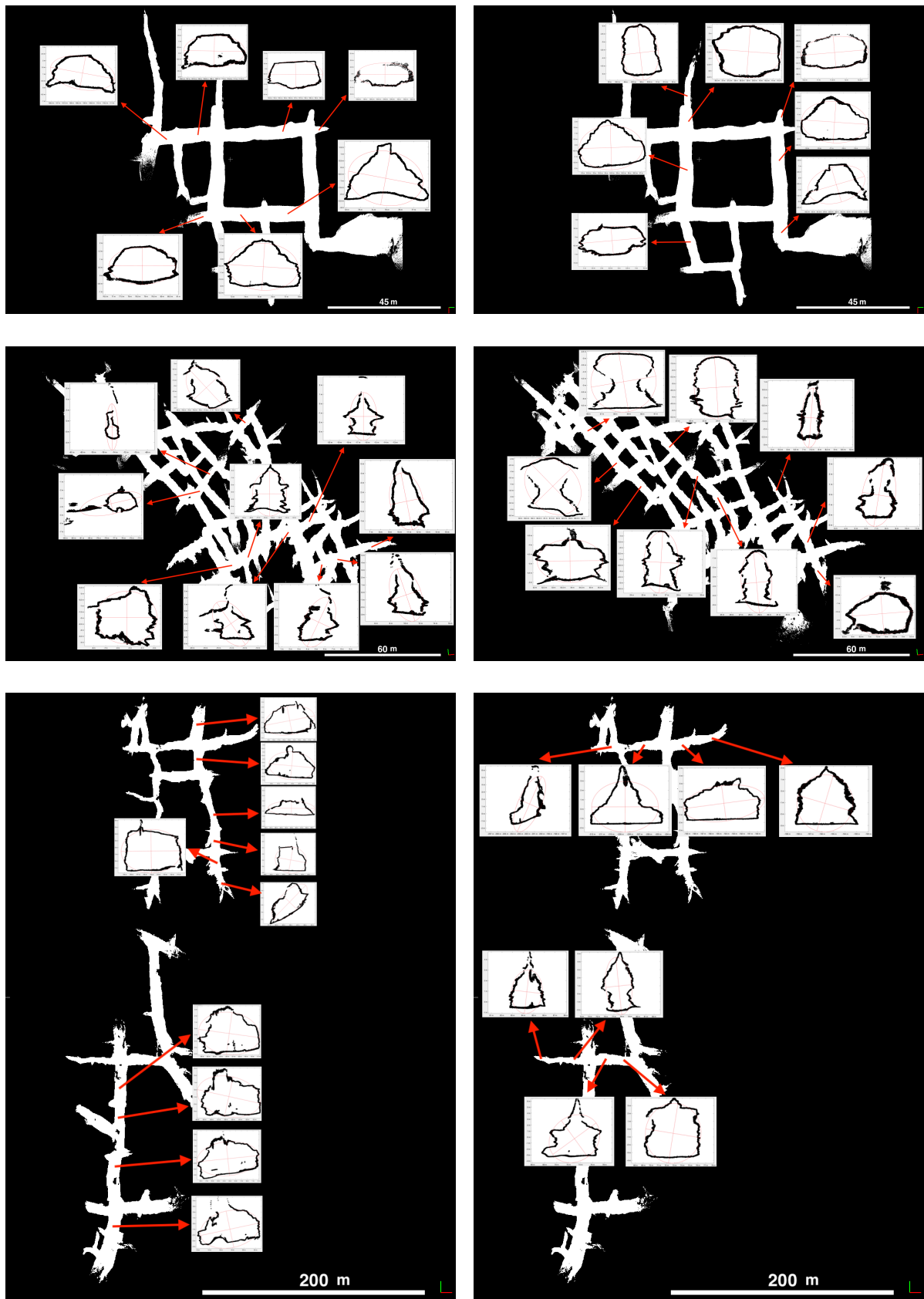
Figure 6.31: This figure shows the variation of conduit's shapes within the three caves analyzed. On the first row there are examples from Lapinha, on the second there are examples from Lapa do Bode and on the third one there are examples from Marota.

Sub-vertical and sub-horizontal elongated cross sections can be recognized in figure 6.31, especially in Lapinha, however different shapes are present as well. These derive from different factors influencing the direction of dissolution within a conduit during time. Cross sections with a flat ceiling usually indicate that the overlying layer acts as a seal for fluid flow, while when top notches can be distinguished, sediment infill in the bottom part of the section in a phreatic environment forces the flow to erode upward. The sediment accumulation in addition, causes also some concave shapes that can be observed at the floor of few cross sections. When the passages are in vadose conditions instead, they tend to enlarge laterally, sometimes creating stair-like shapes if fluctuations of the water-table level occur.

6.8. Volumetric Calculations Discussion

From the volumetric calculations the first thing that clearly emerges is that, even in areas interested by speleogenetic processes, the dissolved percentage over the entire affected bedrock volume doesn't exceed the 5%. This makes karst systems exceptional features of the subsurface, fact due to the highly selective speleogenetic processes that allow only a minimal part of joints to develop into traversable sized passages. Another consideration regarding the local patterns of fluid flow can be done, by observing that the dissolved percentage in the NNW-SSE/NW-SE direction always exceeds the one in the ENE-WSW/NE-SW. From the calculations in fact 35465 m^3 result dissolved in the first direction, while 21933 m^3 in the second direction, making the NNW-SSE/NW-SE orientated conduits to account for the 61.8% of the total caves' volume. This difference can suggest that at the time of formation of the karst systems here studied, a preferential flow path oriented in the NNW-SSE/NW-SE direction was present across the study area.

6.9. Mechanical Stability Discussion

The deformation results of the cases listed at table 6.2 are not supposed to be the final point of a mechanical stability analysis, on the contrary they should be considered as a starting point for future studies where consequences of changes in the mechanical properties of materials can be observed.

Thanks to the legend displayed in each of the figures from figure 6.23 to 6.27, it is possible to notice the mechanical behaviours of materials with different properties. As already stated, likely E and ν values for carbonates are respectively 6 GPa and 0.3, thus figure 6.23 can be considered the standard case. For softer materials, such as the ones recreated for cases 2 and 4, the displacement magnitude increases since more strain is accommodated, while for stiffer materials, such as the ones at cases 3 and 5, the displacement of the particles is less considerable since the stress applied causes less deformation.

Conclusions and Recommendations

7.1. Conclusions

The study of Lapinha, Lapa do Bode and Marota caves gives useful dimensional data from different sized cave systems. The three 3D point clouds obtained thanks to a Lidar laser scanner in the state of Bahia, Brazil, are instruments that allow quick and easy data extraction. Thanks to the Matlab code developed in this study in fact, it is possible to obtain hundreds of conduits' cross sections in few minutes and, as far as a 3D point cloud is available, the method can be applied to any cave. The parameters obtained in this way, such as width, height and area of the ellipse interpolated on a cross section, allow to study the variations of the dimensions between different conduits and study the volumes of cave systems.

The rose diagrams showing the main orientations of elongation of the conduits, that coincide with the orientation of deformation structures such as E-W and N-S oriented folds formed in the Irecê Basin during the Brasiliano Orogeny, prove the close relationship between regional structural geology and speleogenesis. The perpendicularity between the conduits suggests that the cave systems developed from orthogonal sets of joints, generated in different periods of regional or local deformation.

Variability of dimensions along the conduits and between the conduits, is justified by the characteristics that fracture corridors can assume, that highly influence the dissolutional processes in the subsurface, especially in the early stages. Fractures intersections in fact tend to channelize the fluid flow, thus higher densities of fractures intersecting each other cause the development of bigger conduits. In addition changes in the continuity and connectivity within these networks alter locally the hydraulic conductivity of the fracture corridors and consequently theur dissolution.

Intersections between joints of different sets or between joints and bedding planes initially condition the shape of a conduit's cross section, that can assume sub-horizontal or sub-vertical shapes. The presence of top and lateral notches or peculiar shapes of the floor instead develop in the mature phases of dissolution, when sediment supply and level of water table change the preferential direction of dissolution.

The mechanical stability analysis demonstrated that simulations through the software FEniCS are feasible and that the results are reliable, since for higher values of E the deformation of the material decreases, while for smaller ν values it increases, as expected.

7.2. Recommendations

Many recommendations can be made regarding the cross sections analysis. First of all the Matlab code can be improved in the noise reduction, making sure that the cross sections are isolated and don't present isolated clusters of points around it. These in fact can alter the height and width measurements, that could also be changed

if there is the necessity to calculate width and height as the maximum distance between lateral and vertical walls respectively in the horizontal and in the vertical direction. This would be especially useful in the cases of obliquely developed passages.

In the study the ellipse was considered as a good approximation from where to derive the area of a cross section, however in order to be more accurate a more complex curve can be interpolated.

In addition, since the intersections between bedding planes and fractures favour the most the cave passages development, a future combined stratigraphical and structural study may give additional information useful for the prediction of the preferential directions of conduits development during the initial stages of formation.

The simulations run in FEniCS can obviously be highly increased, since it would be interesting to apply more reliable boundary conditions and study the mechanical stability of cave systems overlain and underlain by rock layers at great depths.

A
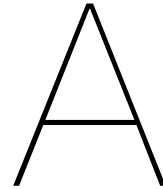
Automatic Skeletonization

In the 'Methodology' chapter, it is described how a first attempt to automatically extract cave skeletons failed. The method consisted into export, thanks to CloudCompare, the denoised point cloud as a .ply file, which can then be loaded in Matlab. Here, the point cloud is projected on the xy plane by giving a constant z-value to every point, in order to obtain a cross section. After that a .jpg of the section and a binarized image of it is created. The source image and the binarized image are used as inputs in a matlab function which gives, as outputs, the skeleton of the cave and a vectorized version of the skeleton(a polyline). The vectorized version can be used in order to convert the polyline into a shape file, that can be analyzed and studied in Matlab or QGIS. The skeleton obtained in this way, however, doesn't suit our purposes since it is not made by singular segments for each branch, but by a multitude of segments. After a careful analysis of the problem we realized that in order to fix the algorithm that generates the skeleton and makes it suitable for our purposes too much time is needed. Nevertheless this can give an idea about a possible new project to be developed.

Cross sections examples

The following figures show cross sections examples extracted from Lapinha, Lapa do Bode and Marota caves. The interpolated ellipses and the interpolation errors (R) are shown too, in order to give an idea about the accuracy of the data.

B.0.1. Lapinha's Cross Sections
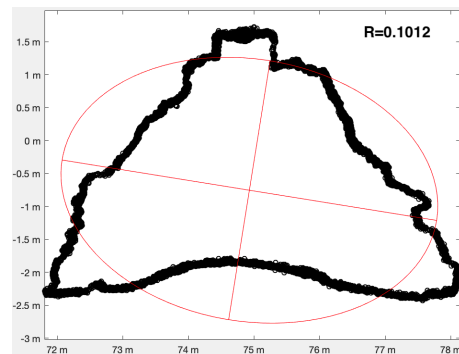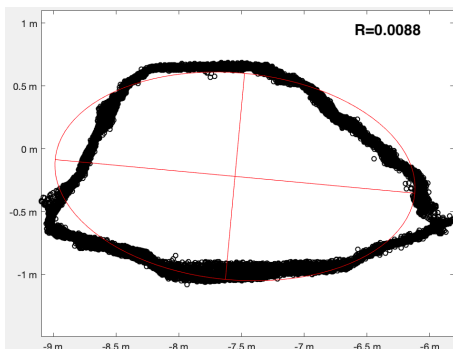
Figure B.1: Cross sections extracted from Lapinha's segments 6 (left) and 8 (right).
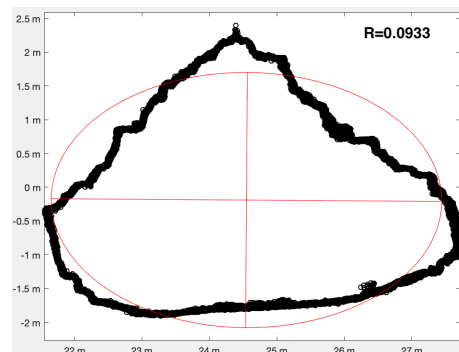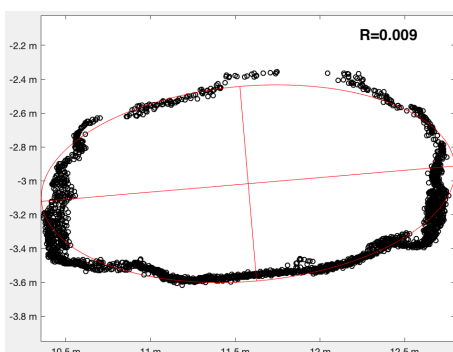
Figure B.2: Cross sections extracted from Lapinha's segments 10 (left) and 14 (right).
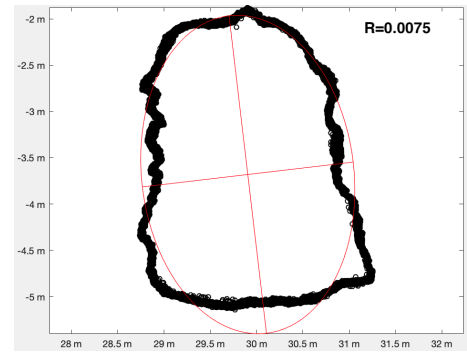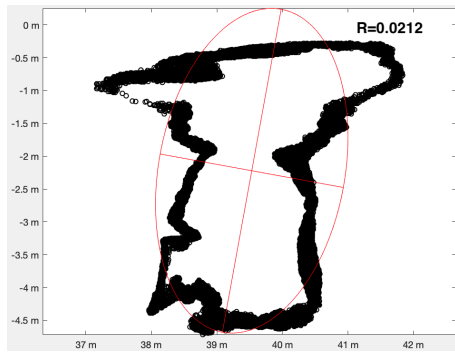
Figure B.3: Cross sections extracted from Lapinha's segments 14 (left) and 23 (right).
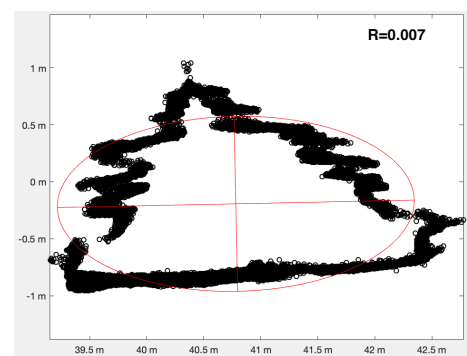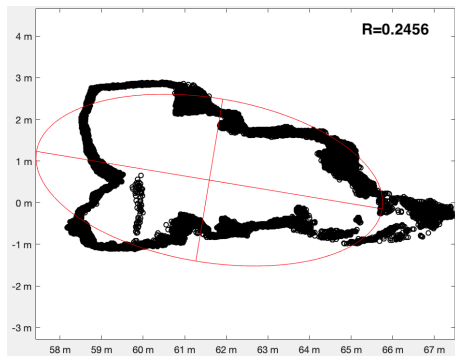
Figure B.4: Cross sections extracted from Lapinha's segments 27 (left) and 31 (right).

B.0.2. Lapa do Bode's Cross Sections
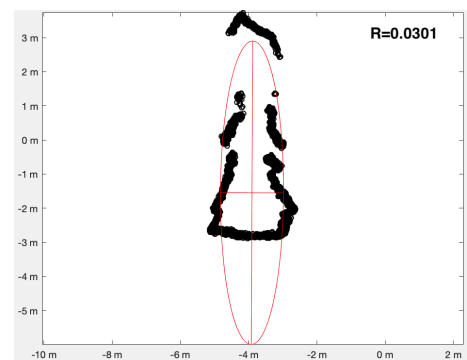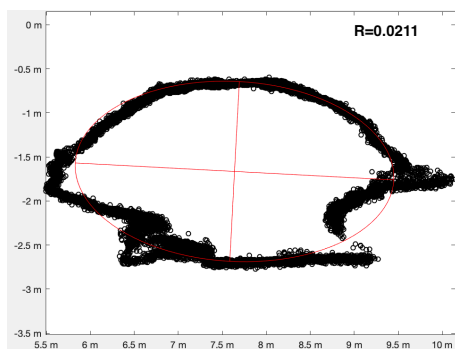
Figure B.5: Cross sections extracted from Lapa do Bodes's segments 8 (left) and 13 (right).
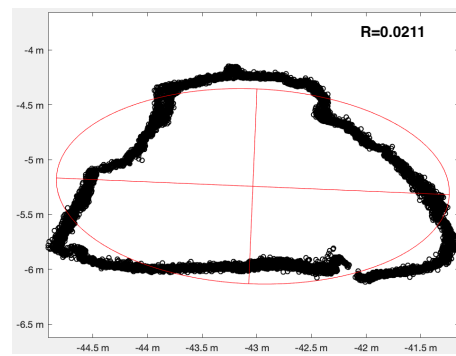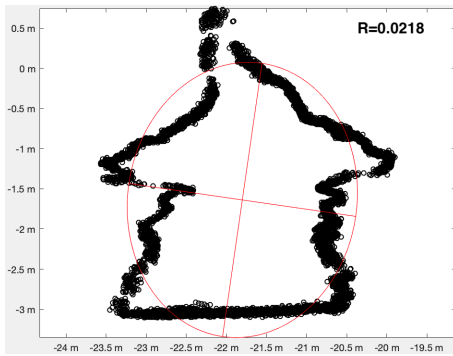
Figure B.6: Cross sections extracted from Lapa do Bodes's segments 17 (left) and 18 (right).
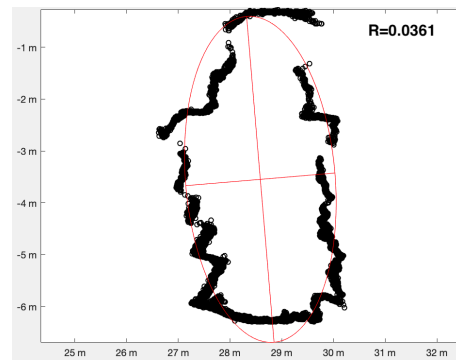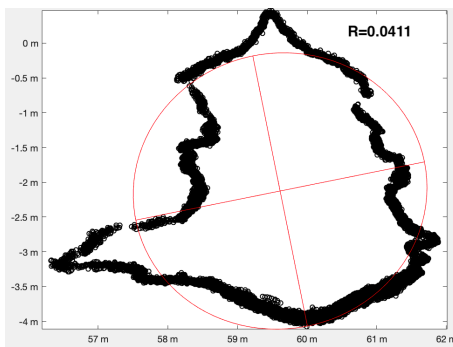
Figure B.7: Cross sections extracted from Lapa do Bodes's segments 30 (left) and 40 (right).
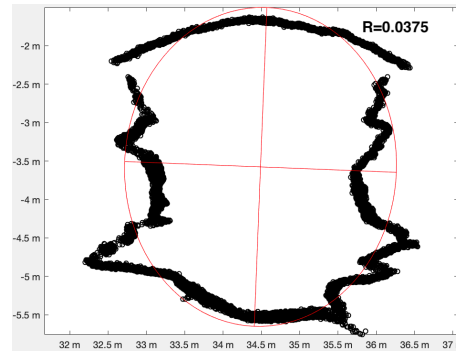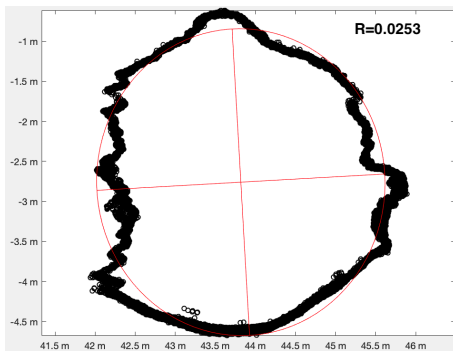
Figure B.8: Cross sections extracted from Lapa do Bodes's segments 47 (left) and 52 (right).
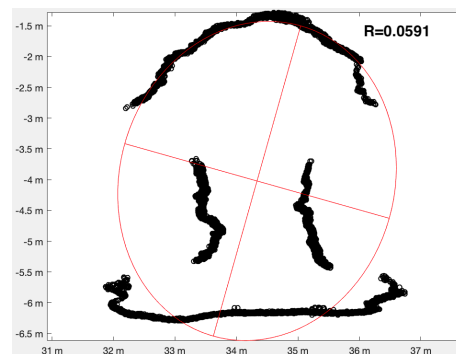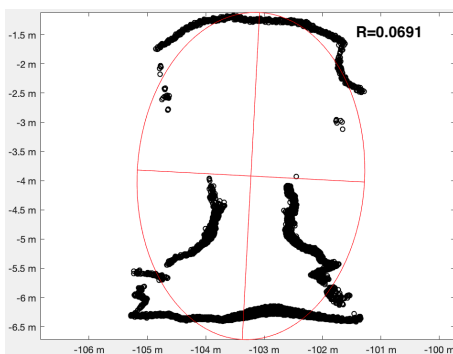
Figure B.9: Cross sections extracted from Lapa do Bodes's segments 54 (left) and 69 (right).
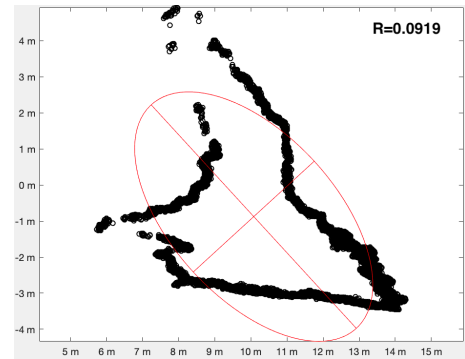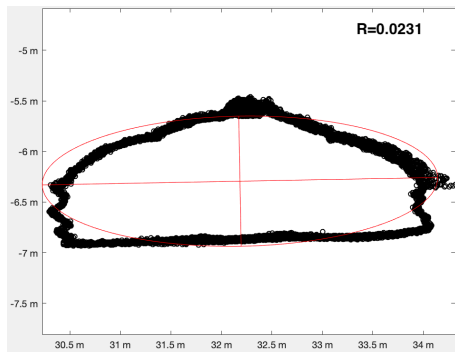
Figure B.10: Cross sections extracted from Lapa do Bodes's segments 77 (left) and 97 (right).

B.0.3. Marota's Cross Sections
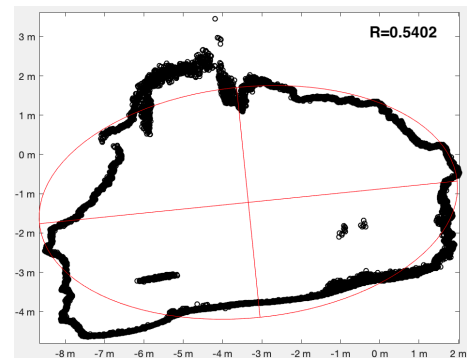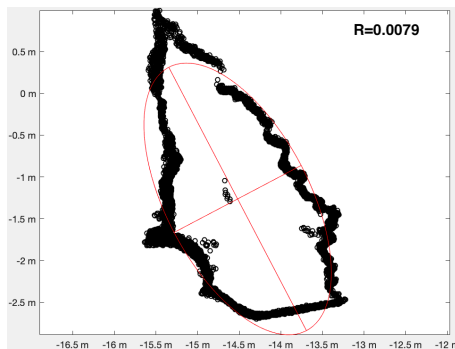
Figure B.11: Cross sections extracted from Marota's segments 9 (left) and 16 (right).
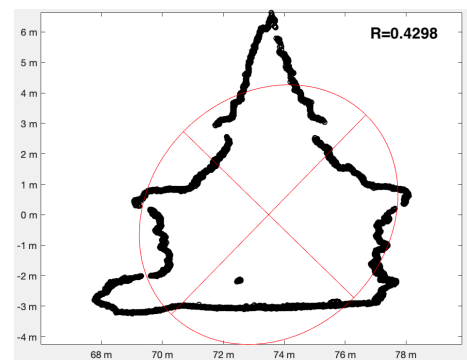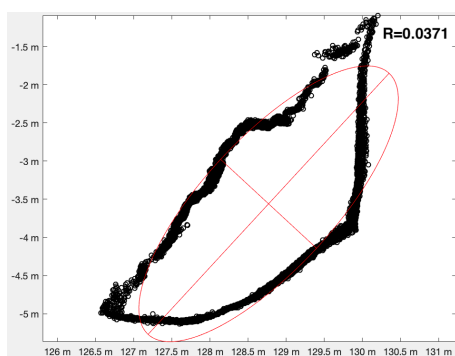
Figure B.12: Cross sections extracted from Marota's segments 18 (left) and 25 (right).
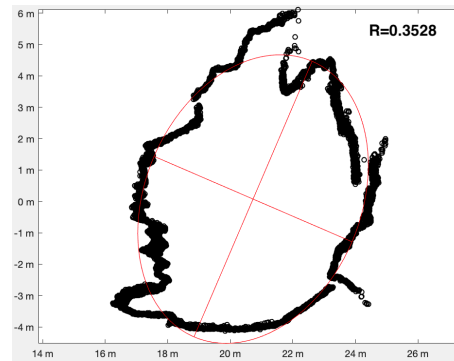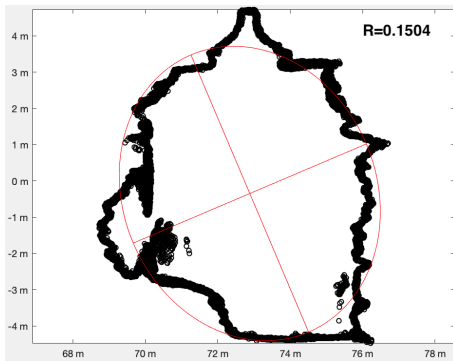
Figure B.13: Cross sections extracted from Marota's segments 25 (left) and 27 (right).
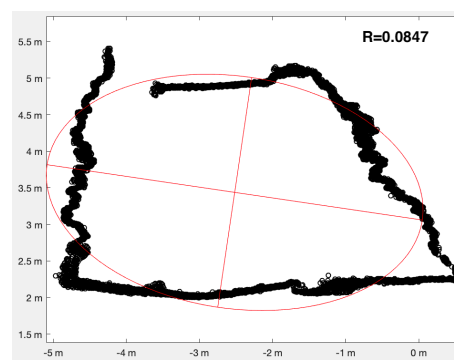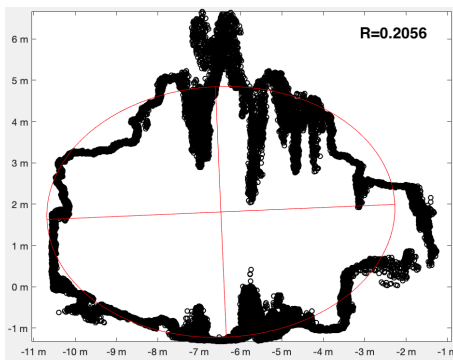
Figure B.14: Cross sections extracted from Marota's segments 31 (left) and 41 (right).
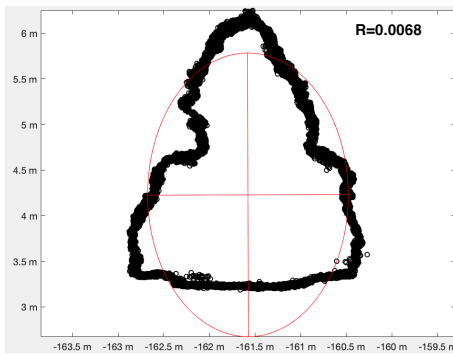
Figure B.15: Cross sections extracted from Marota's segments 45 (left) and 52 (right).
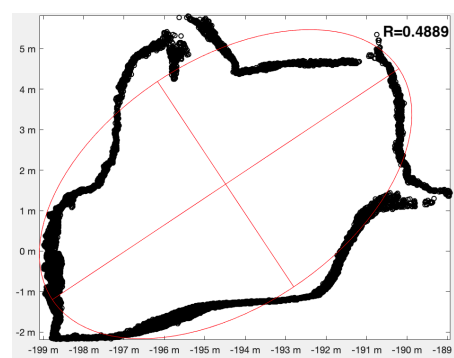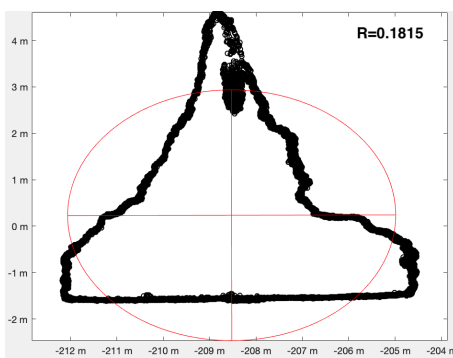
Figure B.16: Cross sections extracted from Marota's segments 56 (left) and 58 (right).
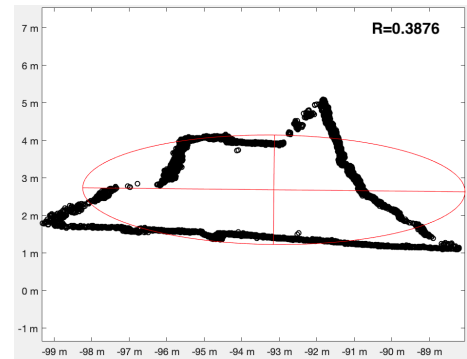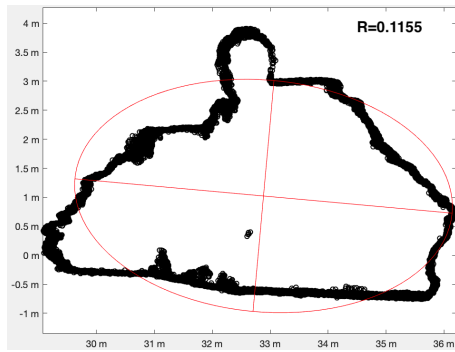
Figure B.17: Cross sections extracted from Marota's segments 62 (left) and 64 (right).
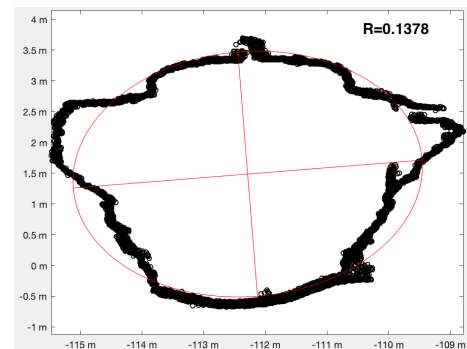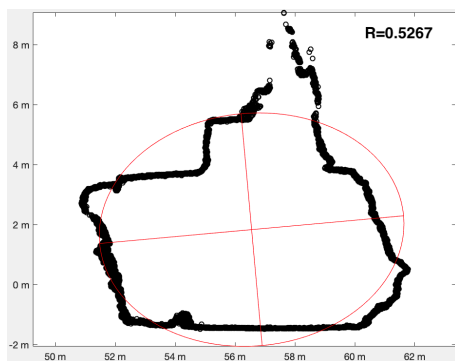
Figure B.18: Cross sections extracted from Marota's segments 65 (left) and 66 (right).

C
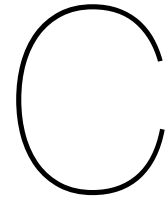
Functions Flowcharts

Three flowcharts for the three different functions present within the code are shown. Blue ovals represent starting points of a code or of a loop, green parallelograms represent inputs, yellow rectangles represent actions or calculations, red rhombuses represent decisions and turquoise rectangles represent alternatives.
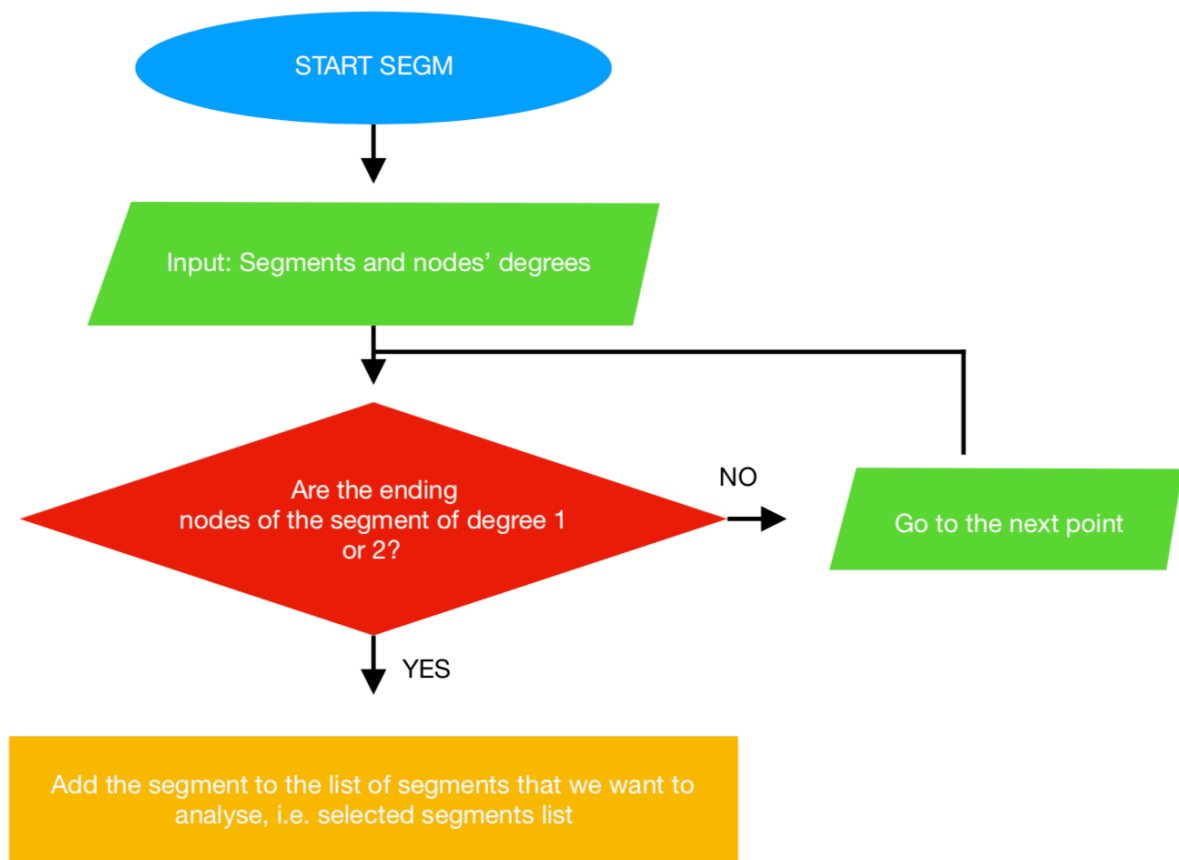
Figure C.1: Function SEGM.
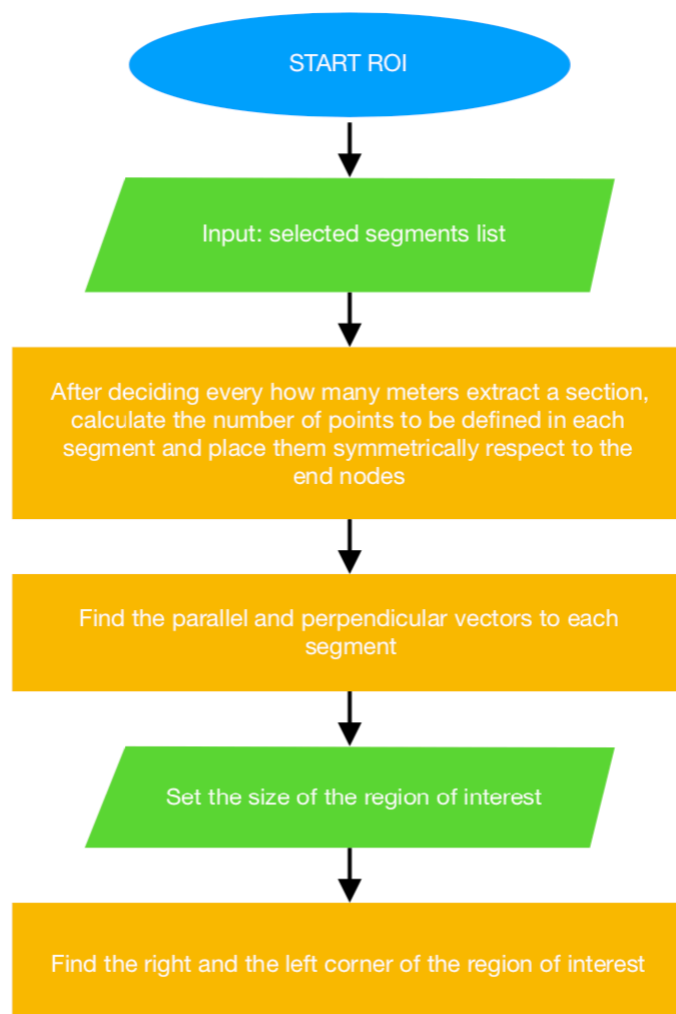
Figure C.2: Function ROI
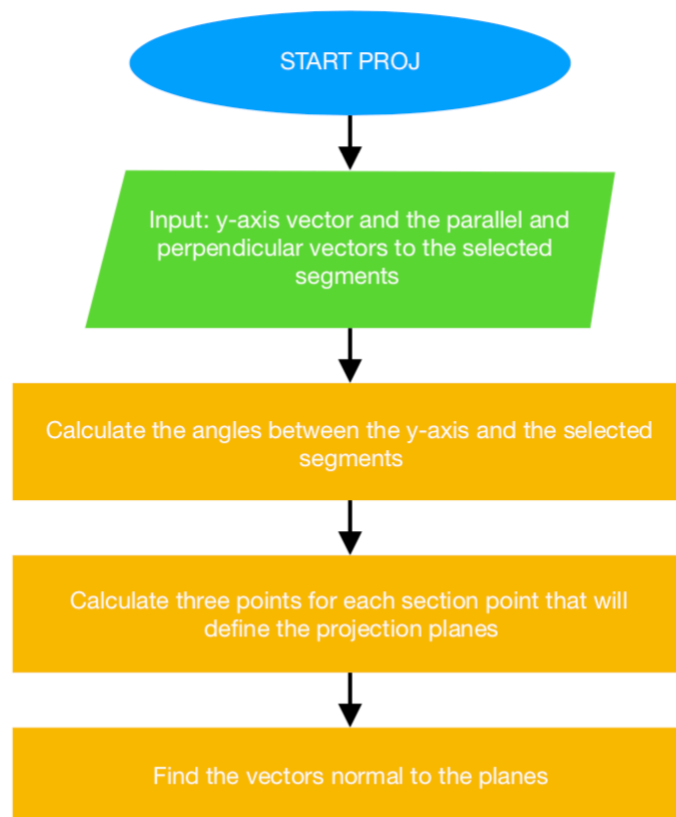
Figure C.3: Function PROJ
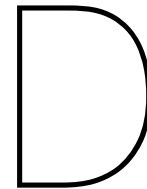
Caves' skeletons

The followings are the skeletons of Lapinha, Lapa do Bode and Marota. Red dots represent 1st degree nodes, yellow dots represent 2nd degree nodes, green dots represent 3rd degree nodes and blue dots represent 4th degree nodes.
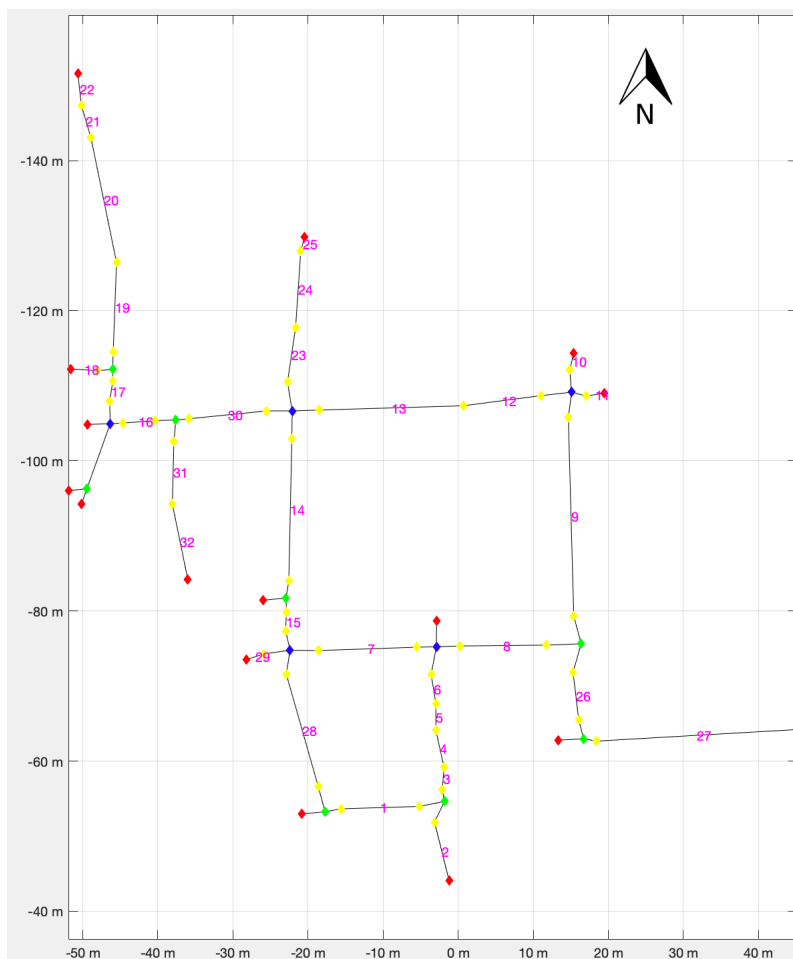
Figure D.1: This figure shows the skeleton of Lapinha cave, with its nodes' degrees.
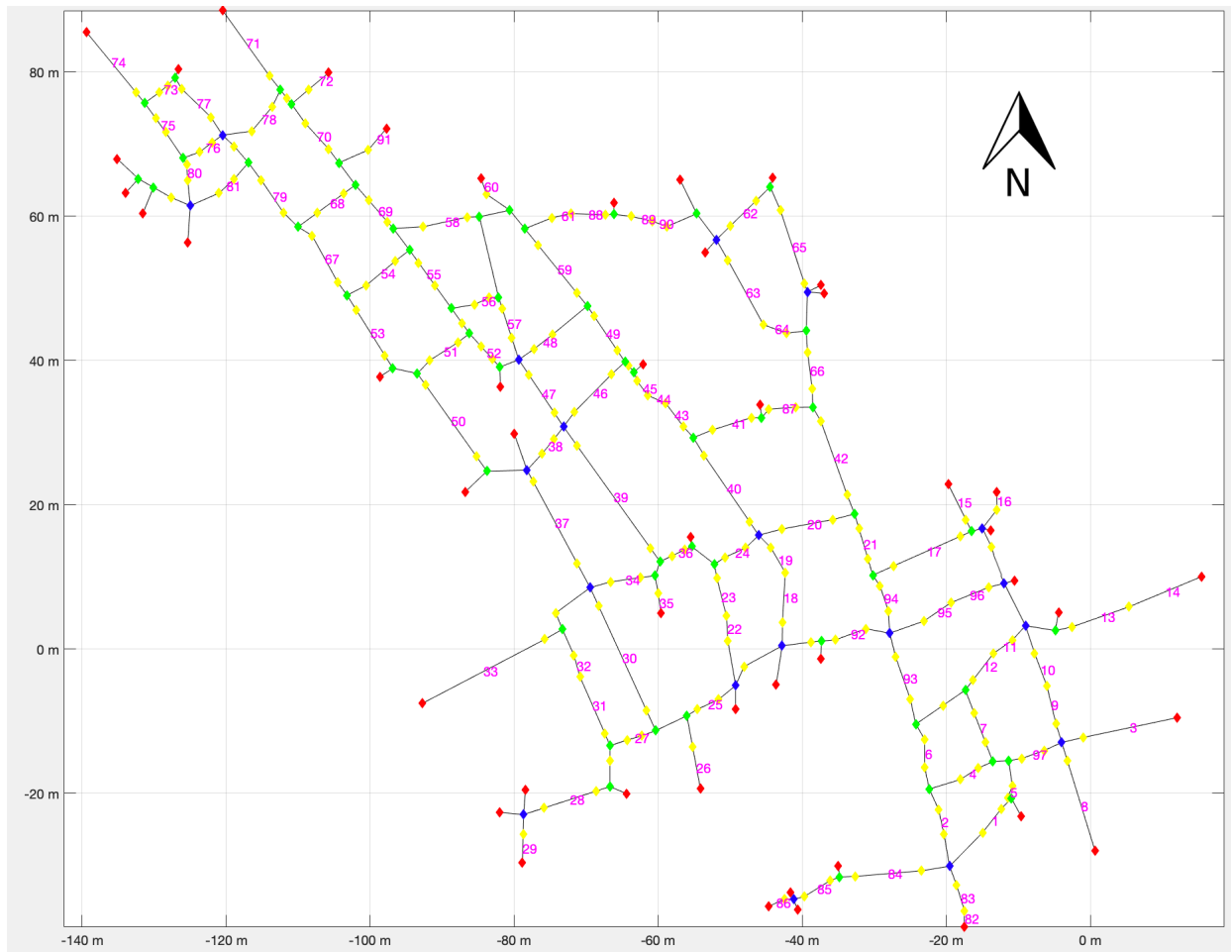
Figure D.2: This figure shows the skeleton of Lapa do bode cave, with its nodes' degrees.
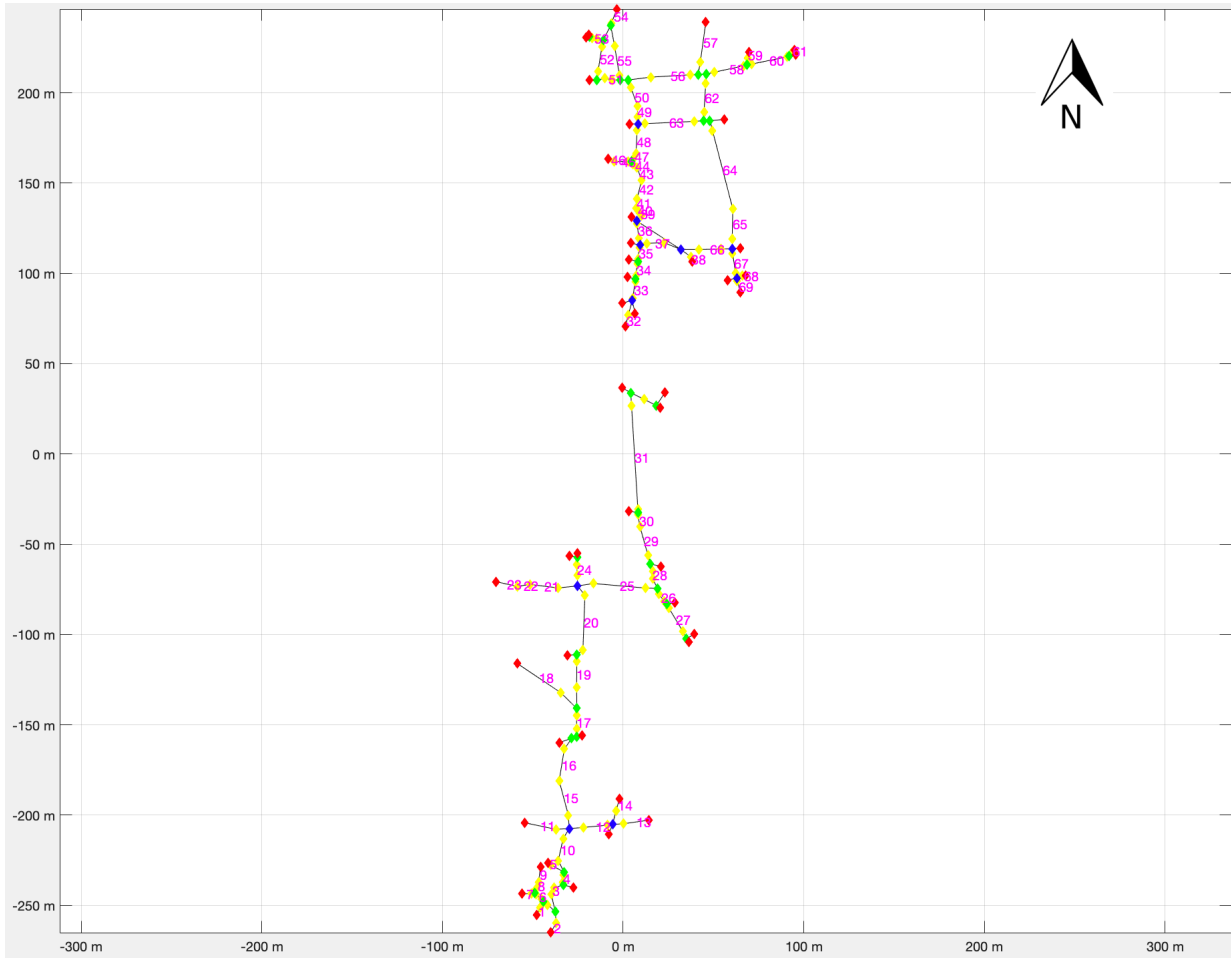
Figure D.3: This figure shows the skeleton of Marota cave, with its nodes' degrees.
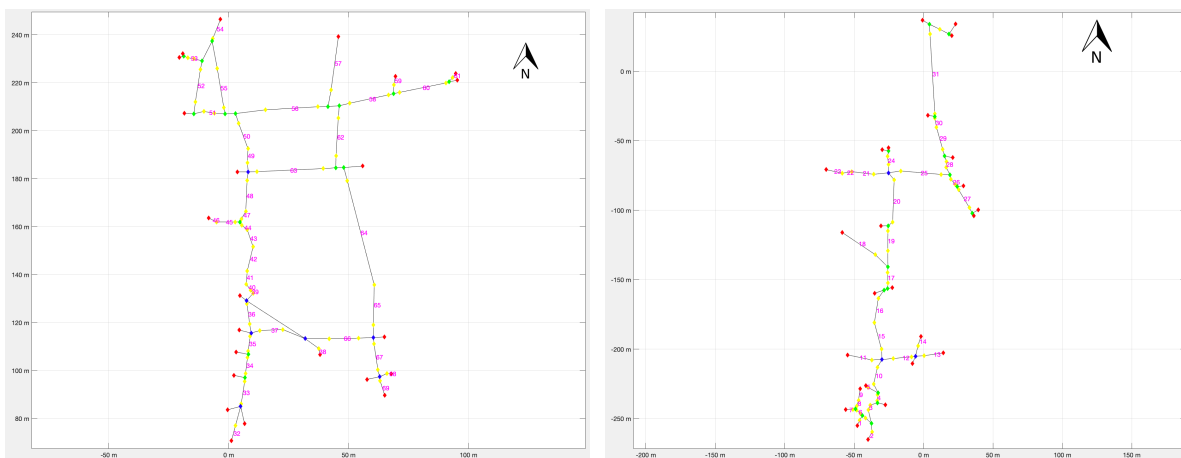
(a) Marota's northern side skeleton, with its nodes' degrees.

(b) Marota's southern side skeleton, with its nodes' degrees.

Figure D.4: Detail of northern and southern sides of marota's skeleton.
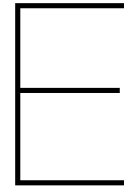
Complete Dimensional Plots
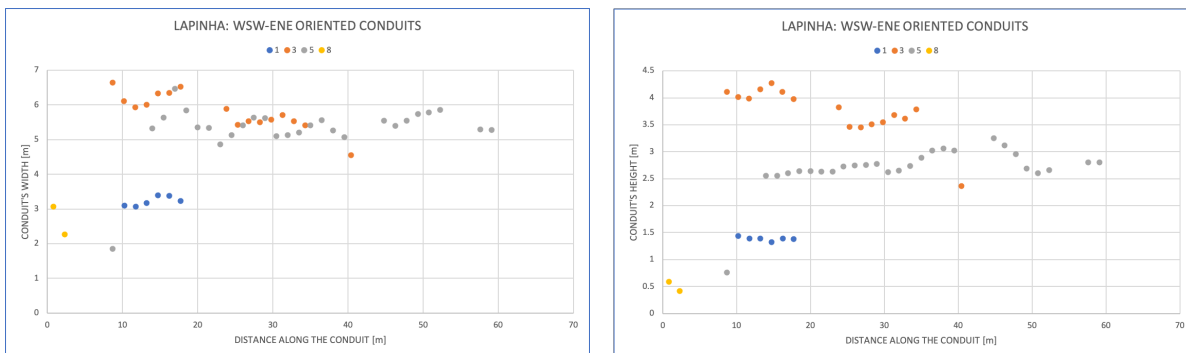
Figure E.1: Lapinha WSW-ENE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.
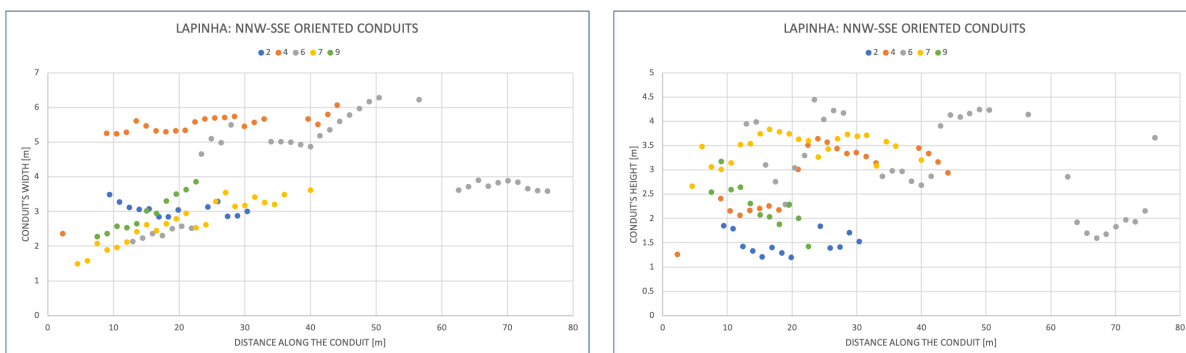
Figure E.2: Lapinha NNW-SSE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.
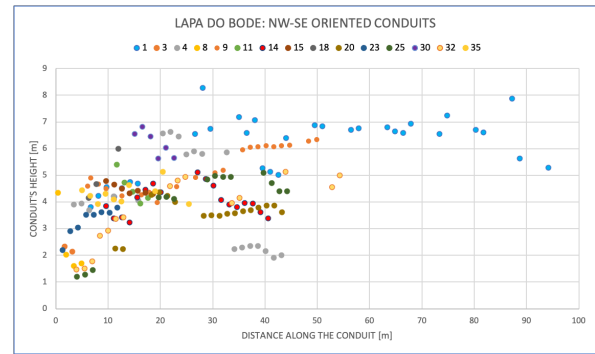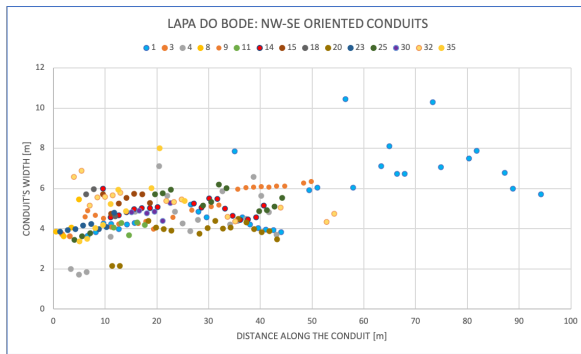
Figure E.3: Lapa do Bode NW-SE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.
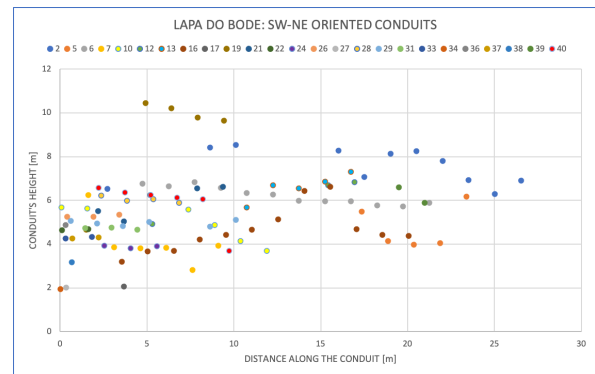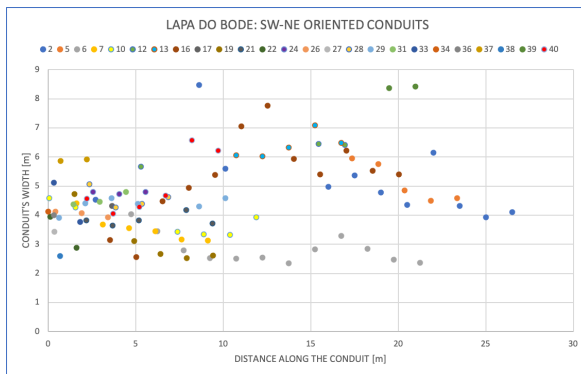
Figure E.4: Lapa do Bode SW-NE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.
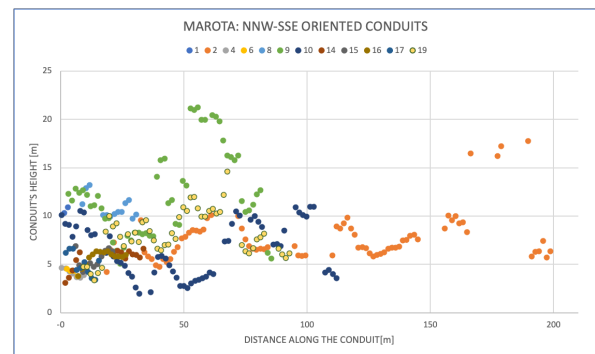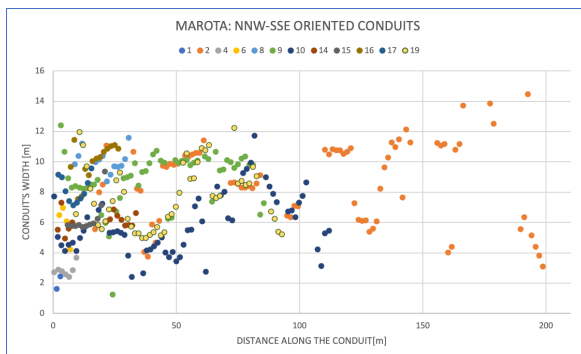
Figure E.5: Marota NNW-SSE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.
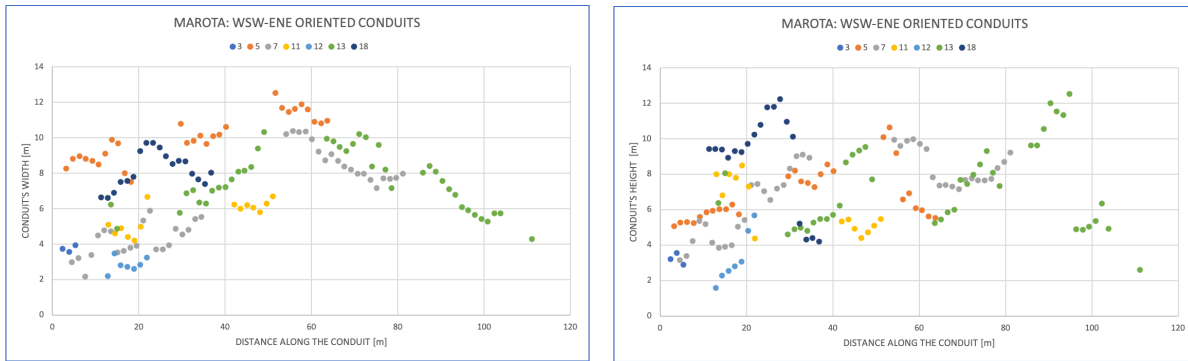
Figure E.6: Marota WSW-ENE plots of branch width vs distance along the branch(left) and branch height vs distance along the branch(right). On the legend the numbers of Lapa do Bode's conduits are listed.

F
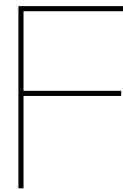
Code

F.1. Main Code

```matlab
1
2  clc
3  close all
4  clear all
5
6  tic
7
8  %% Load the pointcloud and its skeleton by converting the skeleton's shapefile into data
9  %the function Shape_to_data was provided by Rahul Prabhakaran.
10 B = shaperead('your_skeleton.shp');
11 [F, xmin, xmax, ymin, ymax] = Shape_to_Data(B);
12 E = pcread('YOUR_CAVE.ply');
13
14 %% Highlight the skeleton's nodes and identify the I-M-Y-X topological relationships ...
        between the segments
15 % from Rahul Prabhakaran (2017)
16 disp(strcat('No of Segments: ',num2str(length(F))))
17 disp(strcat('Shapefile Entries: ',num2str(length(B))))
18 [EndNodes,XY] = create_GraphEdges(F, xmin, ymin);
19
20 G1=graph();
21 for i=1:length(EndNodes)
22     G1 =  addedge(G1,EndNodes{i,1}{1,1},EndNodes{i,1}{1,2});
23 end
24
25 % calculating degree of nodes
26 D1 = degree(G1);
27 % extracting M_nodes
28 [M_nodes_idx,¬] = find(D1(:,1)==1); %red
29 M_nodes=XY(M_nodes_idx,:);
30 % extracting I_nodes
31 [I_nodes_idx,¬] = find(D1(:,1)==2); %yellow
32 I_nodes=XY(I_nodes_idx,:);
```

```matlab
33   % extracting Y_nodes
34   [Y_nodes_idx,¬] = find(D1(:,1)==3); %green
35   Y_nodes=XY(Y_nodes_idx,:);
36   % extracting X_nodes
37   [X_nodes_idx,¬] = find(D1(:,1)==4); %blue
38   X_nodes=XY(X_nodes_idx,:);
39
40   A1 = adjacency(G1);
41
42   %% Isolate the segments that we are interested in studying, which are the segments ...
         linked by I−M, I−I nodes.
43
44   [segm_check, length_segm_M, length_segm_I, length_segm_Y, length_segm_X] = SEGM (F, ...
         M_nodes, I_nodes, Y_nodes, X_nodes);
45
46   figure(1)
47   gplot(A1,XY,'k')
48   pbaspect([1 1 1])
49   axis equal
50   xtickformat('%g m')
51   ytickformat('%g m')
52   hold on
53   grid on
54   scatter(I_nodes(:,1),I_nodes(:,2),'d','filled','y');
55   hold on
56   scatter(Y_nodes(:,1),Y_nodes(:,2),'d','filled','g');
57   hold on
58   scatter(M_nodes(:,1),M_nodes(:,2),'d','filled','r');
59   hold on
60   scatter(X_nodes(:,1),X_nodes(:,2),'d','filled','b');
61   hold on
62   for ii = 1:length(segm_check(:, 1))
63       t = text(segm_check(ii, 1) + (segm_check(ii, 3) − segm_check(ii, 1))/2, ...
             segm_check(ii, 2) + (segm_check(ii, 4) − segm_check(ii, 2))/2, num2str(ii));
64       t.Color = [1 0 1];
65   end
66
67
68   %% Obtain the parameters needed to find the cross sections of the cave along
69   % the segments we are interested in every ∆_s
70   % Define a z interval in which the pointcloud is located
71   min_z = −40;
72   max_z = +40;
73
74   % Define the width and the length of the subregion
75   width_large = 80;  % width of search window _|_ to segment
76   epsilon = 1;  % additional distance
77   width_small = 0.20;  % length "slab" || to segment
78
79   [Ui, Vi, left_corner, right_corner, P_1, nslices, section_points, l_segment, P_1i, ...
         dist_points] = ROI (segm_check, width_large, epsilon, width_small, min_z, max_z);
80
81   %find the parameters useful for projecting the subpointclouds
82   [R, Q_1, normal, angle_deg] = PROJ (Ui, left_corner, P_1, Vi, width_large, epsilon, ...
         max_z);
83
84   %use the following 24 lines and keep commented the orient_idx line if you want
```

```matlab
85  %to isolate the vertical segments, otherwise comment the following 3 lines
86  %to isolate the horizontal segments
87
88  %find the vertical segments indexes
89  % orient_idx_c = find(angle_deg≤25);
90  % orient_idx_a = find(angle_deg≥155 & angle_deg≤180);
91  % orient_idx = [orient_idx_c orient_idx_a];
92  %
93  % %find the horizontal segments indexes
94  % %orient_idx = find(angle_deg≥65 & angle_deg≤115);
95  %
96  % %ridefine all the useful parameters to be used in the for loop
97  % rows_three = zeros(length(orient_idx)*3, 1);
98  % rows_two = zeros(length(orient_idx)*2, 1);
99  % for i = 1:length(orient_idx)
100 % rows_three([i*3-2 i*3-1 i*3]) = [orient_idx(i)*3-2 orient_idx(i)*3-1 orient_idx(i)*3];
101 % rows_two([i*2-1 i*2]) = [orient_idx(i)*2-1 orient_idx(i)*2];
102 % end
103 % Ui = Ui(:, orient_idx);
104 % Q_1 = Q_1(rows_three,:);
105 % normal = normal(rows_three);
106 % R = R(:, rows_two);
107 % section_points = section_points(rows_two, :);
108 % left_corner = left_corner(rows_three, :);
109 % right_corner = right_corner(rows_three, :);
110
111 toc
112 tic
113 %% Apply the subsumpling and projection. Once the projection is available, an ellipse ...
        is fitted.
114 pointcloud = E.Location;
115 %take 1 point every 3 of the pointcloud in order to decrease its size and
116 %the computational times
117 pointcloud = pointcloud(1:3:length(pointcloud), :);
118 rot_pointcloud = pointcloud;
119
120 section_points_s = [];
121 Q_1_s = [];
122 normal_s = zeros(3, 1);
123 R_loop = zeros(2, 2);
124 rot_pointcloud_s = zeros (length(pointcloud(:, 1)), 3);
125 rot_left_corner = [];
126 rot_right_corner = [];
127 a_segm = [];
128 b_segm = [];
129 error_interpolation_segm = [];
130 ellipse_segm = [];
131 ver_line_segm = [];
132 horz_line_segm = [];
133 xwidth_segm = [];
134 yheight_segm = [];
135 dimension_segm_w = [];
136
137 for i = 1:length(Ui)
138 %for i = [x] %to analyze specific segments
139 a_segm = zeros(1, nslices(1, i));
140 b_segm = zeros(1, nslices(1, i));
```

```matlab
141    xwidth_segm = zeros(1, nslices(1, i));
142    yheight_segm = zeros(1, nslices(1, i));
143    error_interpolation_segm = zeros(1, nslices(1, i));
144    ellipse_segm = zeros(nslices(1, i)*2, 100);
145    ver_line_segm = zeros(nslices(1, i)*2, 2);
146    horz_line_segm = zeros(nslices(1, i)*2, 2);
147    dimension_segm_w = zeros(nslices(1, i), 4);
148    Q_1_s = Q_1([i*3-2 i*3-1 i*3], 1:nslices(1, i));
149    normal_s = normal([i*3-2 i*3-1 i*3], 1);
150    R_loop = R(:, [i*2-1 i*2]);
151    section_points_s = section_points ([i*2-1 i*2], 1:nslices(1, i) + 1);
152    P_1i_s = P_1i(:, i);
153
154    % Only rotating x- and y-coordinates, since we assume segments
155    % are constant in z-direction!!!
156    rot_pointcloud_s(:, 1:2) = rot_pointcloud(:, 1:2) * R_loop';
157    rot_left_corner = left_corner([i*3-2 i*3-1 i*3], 1:nslices(1, i));
158    rot_left_corner(1:2, :) = R_loop * rot_left_corner([1 2], :);
159    rot_right_corner = right_corner([i*3-2 i*3-1 i*3], 1:nslices(1, i));
160    rot_right_corner(1:2, :) = R_loop * rot_right_corner([1 2], :);
161
162    for ii = 1:length(Q_1_s(1, :))
163    %for ii = x %to analyze a specific slice
164    % Extract sub-pointcloud:
165    bool_section = ((rot_pointcloud_s(:, 1) ≥ rot_left_corner(1, ii) & rot_pointcloud_s(:, ...
           1) ≤ rot_right_corner(1, ii)) & ...
166                    (rot_pointcloud_s(:, 2) ≥ rot_left_corner(2, ii) & rot_pointcloud_s(:, ...
                        2) ≤ rot_right_corner(2, ii)) & ...
167                    (rot_pointcloud_s(:, 3) ≥ rot_left_corner(3, ii) & rot_pointcloud_s(:, ...
                        3) ≤ rot_right_corner(3, ii)));
168
169    % Rotate back:
170    sub_pointcloud = pointcloud(bool_section, :);
171    %uncomment if yuo want to show the subsumpled pointcloud
172    %    figure()
173    %    plot3(sub_pointcloud(:, 1), sub_pointcloud(:, 2), sub_pointcloud(:, 3), '.', ...
174    %          'color', [0, 0, 0], 'LineWidth', 1)
175    %     axis equal
176    num_sub_pts = size(sub_pointcloud, 1);
177    % Perform projection of all sub-points on plane:
178    projected_points = sub_pointcloud - (sub_pointcloud - ones(num_sub_pts, 1) * Q_1_s(:, ...
           ii)') * normal_s .* (ones(num_sub_pts, 1) * normal_s');
179    % Rotate the subpointcloud:
180    sub_pointcloud = sub_pointcloud(:, 1:2) * R_loop';
181
182    %We want to isolate the subpointcloud related to the branch of the cave we
183    %are analyzing, and not take into account the nearby branches.
184    % Final bin is simply the x-coordinate of rotated sub point cloud:
185    distr_points_x_axis = sub_pointcloud(:, 1);
186    % Center the segment close to x-axis 0 value
187    center_segm = R_loop(1, :) * section_points_s(:, ii);
188    distr_points_x_axis = distr_points_x_axis - center_segm;
189    %figure(5) %to show the histogram
190    %histogram(distr_points_x_axis, 'NumBins', 20)
191    % Group into bins:
192    % Want to bin every X meter --> size_bin = X
193    size_bin = 1;
```

```matlab
194  nbins = round(width_large * 3 / size_bin);
195  [count, edges, bin] = histcounts(distr_points_x_axis, nbins);
196  %count the number of values in each bins
197  over_bin = 1e10;
198  count(count < 1) = over_bin;
199  bin_count = [count; 1 : nbins];
200  bin_mdpt = (edges(1:end-1) + edges(2:end))/2;
201  %find the index of the bin closer to the 0 value on the x-axis of the
202  %histogram
203  idx = min(abs(bin_mdpt));
204  index = find(abs(bin_mdpt) == idx);
205  index = index(1, 1);
206
207  %Find the right and left limits were the subpointcloud related to our
208  %segment ends. The limits are defined by a minimum number of values found
209  %in a bin.
210  split_right = count(1, index:nbins);
211  for iii = 1:(nbins-index)+1
212      if split_right(iii) == over_bin
213      split_right(iii) = 1;
214
215      else
216      split_right(iii) = 0;
217
218      end
219
220      if max(split_right) == 0
221      split_right(length(split_right) + 1) = 1;
222
223      end
224  end
225  limit_right = find(split_right, 1, 'first');
226  limit_right = index + limit_right - 2;
227
228  split_left = count(1, 1:index-1);
229  for iii = 1:index-1
230      if split_left(iii) == over_bin
231          split_left(iii) = 1;
232
233      else
234          split_left(iii) = 0;
235      end
236
237      if max(split_left) == 0
238          split_left(1, 1) = 1;
239
240      end
241  end
242  limit_left = find(split_left, 1, 'last');
243  limit_left = limit_left + 1;
244
245  count_treshold = count(:, limit_left : limit_right);
246  right_edge = edges(:, limit_right + 1);
247  left_edge = edges(:, limit_left - 1);
248  distr = (distr_points_x_axis ≥ left_edge) & (distr_points_x_axis ≤ right_edge);
249
250  %Parameters calculation
```

```matlab
251  rot_projected_points = projected_points;
252  rot_projected_points(:, 1:2) = rot_projected_points(:, 1:2) * R_loop';
253  locpoints = rot_projected_points (distr, [1, 3]);
254  locpoints_old = locpoints;
255  xwidth = max(locpoints_old(:, 1)) - min(locpoints_old(:, 1));
256  yheight = max(locpoints_old(:, 2)) - min(locpoints_old(:, 2));
257  num_pts = size(locpoints, 1);
258  mean_x = mean(locpoints(:, 1));
259  mean_y = mean(locpoints(:, 2));
260  locpoints(:, 1) = locpoints(:, 1) - mean_x;
261  locpoints(:, 2) = locpoints(:, 2) - mean_y;
262  % Fit ellipse to data:
263  % General equation:
264  % A*x^2 + B*x*y + C*y^2 + D*x + E*y + F = 0
265  % A*x^2 + B*x*y + C*y^2 + D*x + E*y = -F (assume F = -1)
266  A = [locpoints(:, 1).^2, locpoints(:, 1).*locpoints(:, 2), locpoints(:, 2).^2, ...
         locpoints(:, 1), locpoints(:, 2)];
267  coef = (A' * A) \ (A' * (ones(num_pts, 1)));
268  [a, b, c, d, e] = deal( coef(1), coef(2), coef(3), coef(4), coef(5) );
269  % Initialize
270  orientation_tolerance = 1e-3;
271  % remove the orientation from the ellipse
272  if ( min(abs(b/a),abs(b/c)) > orientation_tolerance )
273
274      orientation_rad = 1/2 * atan( b/(c-a) );
275      cos_phi = cos( orientation_rad );
276      sin_phi = sin( orientation_rad );
277      [a,b,c,d,e] = deal(...
278          a*cos_phi^2 - b*cos_phi*sin_phi + c*sin_phi^2,...
279          0,...
280          a*sin_phi^2 + b*cos_phi*sin_phi + c*cos_phi^2,...
281          d*cos_phi - e*sin_phi,...
282          d*sin_phi + e*cos_phi );
283      [mean_x,mean_y] = deal( ...
284          cos_phi*mean_x - sin_phi*mean_y, ...
285          sin_phi*mean_x + cos_phi*mean_y );
286  else
287      orientation_rad = 0;
288      cos_phi = cos( orientation_rad );
289      sin_phi = sin( orientation_rad );
290  end
291  % check if conic equation represents an ellipse
292  test = a*c;
293  switch (1)
294  case (test>0),   status = '';
295  case (test==0), status = 'Parabola found'; %warning( 'fit_ellipse: Did not locate an ...
         ellipse' );
296  case (test<0),   status = 'Hyperbola found'; %warning( 'fit_ellipse: Did not locate an ...
         ellipse' );
297  end
298  % if we found an ellipse return it's data
299  if (test > 0)
300
301      % make sure coefficients are positive as required
302      if (a<0), [a,c,d,e] = deal( -a,-c,-d,-e );end
303
304
```

```matlab
305        % final ellipse parameters
306        X0            = mean_x - d/2/a;
307        Y0            = mean_y - e/2/c;
308        F_ellipse     = 1 + (d^2)/(4*a) + (e^2)/(4*c);
309        [a,b]         = deal( sqrt( F_ellipse/a ),sqrt( F_ellipse/c ) );
310        long_axis     = 2*max(a,b);
311        short_axis    = 2*min(a,b);
312        error_interpolation = sum((a * locpoints(:, 1).^2 + b * (locpoints(:, ...
               1).*locpoints(:, 2)) + ...
313          c * locpoints(:, 2).^2 + d * locpoints(:, 1) + e * locpoints(:, 2) - 1).^2); ...
                  %calculate the ellipse interpolation error
314        error_interpolation = sqrt(error_interpolation)/num_pts;
315        % rotate the axes backwards to find the center point of the original TILTED ellipse
316        R_ellipse             = [ cos_phi sin_phi; -sin_phi cos_phi ];
317
318    else
319
320    a = NaN;
321    b = NaN;
322    X0 = NaN;
323    Y0 = NaN;
324    long_axis = NaN;
325    short_axis = NaN;
326    error_interpolation = NaN;
327    R_ellipse = NaN;
328
329    end
330
331    %store the parameters
332    a_segm(1, ii) = a;
333    a_s(sum(nslices(1:(i-1)))+1 : sum(nslices(1:i))) = a_segm;
334    b_segm(1, ii) = b;
335    b_s(sum(nslices(1:(i-1)))+1 : sum(nslices(1:i))) = b_segm;
336    error_interpolation_segm(1, ii) = error_interpolation;
337    error_interpolation_s(sum(nslices(1:(i-1)))+1 : sum(nslices(1:i))) = ...
            error_interpolation_segm;
338    xwidth_segm(1, ii) = xwidth;
339    xwidth_s(sum(nslices(1:(i-1)))+1 : sum(nslices(1:i))) = xwidth_segm;
340    yheight_segm(1, ii) = yheight;
341    yheight_s(sum(nslices(1:(i-1)))+1 : sum(nslices(1:i))) = yheight_segm;
342    % the axes
343    ver_line          = [ [X0, X0]; Y0+b*[-1, 1] ];
344    horz_line         = [ X0+a*[-1, 1]; [Y0, Y0] ];
345    new_ver_line      = R_ellipse * ver_line;
346    new_horz_line     = R_ellipse * horz_line;
347    % the ellipse
348    theta_r           = linspace(0,2*pi);
349    segm_c = [( (section_points_s(1, ii+1) + section_points_s(1, ii))/2) ...
             ((section_points_s(2, ii+1) + section_points_s(2, ii))/2)];
350    X0_loop = segm_c(1, 1);
351    Y0_loop = segm_c(1, 2);
352    rot = R_ellipse * [a*cos(theta_r); b*sin(theta_r)];
353    ellipse_xr_plan = X0_loop + rot(1, :) ;
354    ellipse_yr_plan = Y0_loop + rot(2, :) ;
355    rotated_ellipse_plan = [ellipse_xr_plan; ellipse_yr_plan];
356    ver_line_c = [( (new_ver_line(1, 1) + new_ver_line(1, 2))/2) ((new_ver_line(2, 1) + ...
             new_ver_line(2, 2))/2)];
```

```matlab
357  horz_line_c = [( (new_horz_line(1, 1) + new_horz_line(1, 2))/2) ((new_horz_line(2, 1) + ...
         new_horz_line(2, 2))/2)];
358  cent_ver_line(1, :) = (X0_loop − ver_line_c(1, 1)) + new_ver_line(1, :);
359  cent_ver_line(2, :) = (Y0_loop − ver_line_c(1, 2)) + new_ver_line(2, :);
360  cent_horz_line(1, :) = (X0_loop − horz_line_c(1, 1)) + new_horz_line(1, :);
361  cent_horz_line(2, :) = (Y0_loop − horz_line_c(1, 2)) + new_horz_line(2, :);
362  dimension_segm_w(1:nslices(1, i), [1 2]) = section_points_s(:, 1:nslices(1, i))';
363  dimension_segm_w(1:nslices(1, i), [3 4]) = section_points_s(:, 2:(nslices(1, i)+1))';
364
365  % %plot the cross sections and the ellipse interpolated
366  % figure()
367  % plot(locpoints_old(:, 1), locpoints_old(:, 2), 'o', 'color', [0, 0, 0], 'LineWidth', ...
         1, 'MarkerSize', 5)
368  % hold on
369  % plot( new_ver_line(1,:),new_ver_line(2,:),'r' );
370  % plot( new_horz_line(1,:),new_horz_line(2,:),'r' );
371  % plot( rotated_ellipse(1,:),rotated_ellipse(2,:),'r' );
372  % axis equal
373  % xtickformat('%g m')
374  % ytickformat('%g m')
375  % hold off
376
377  %store the ellipses and the axis to plot
378  ellipse_segm([ii*2−1 ii*2], :) = rotated_ellipse_plan;
379  ellipse_plan((sum(nslices(1:(i−1)))+1)*2−1 : sum(nslices(1:i))*2, :) = ellipse_segm;
380  ver_line_segm([ii*2−1 ii*2], :) = cent_ver_line;
381  ver_line_plan((sum(nslices(1:(i−1)))+1)*2−1 : sum(nslices(1:i))*2, :) = ver_line_segm;
382  horz_line_segm([ii*2−1 ii*2], :) = cent_horz_line;
383  horz_line_plan((sum(nslices(1:(i−1)))+1)*2−1 : sum(nslices(1:i))*2, :) = horz_line_segm;
384  dimension_plan((sum(nslices(1:(i−1)))+1 : sum(nslices(1:i))), :) = dimension_segm_w;
385  %to store data
386  Mwidth{i} = xwidth_segm;
387  Mheight{i} = yheight_segm;
388  Ma{i} = a_segm;
389  Mb{i} = b_segm;
390  end
391  end
392  toc
393
394  %uncomment to plot the ellipses distribution on the skeleton
395  dimension_plan(:, 5) = xwidth_s;
396  width_limit_up = sort(dimension_plan(:, 5));
397  width_limit_down = width_limit_up(8);
398  width_limit_up = width_limit_up(length(width_limit_up) − 8);
399
400  figure()
401  gplot(A1,XY,'k')
402  pbaspect([1 1 1])
403  axis equal
404  xtickformat('%g m')
405  ytickformat('%g m')
406  hold on
407  grid on
408  scatter(I_nodes(:,1),I_nodes(:,2),'d','filled','y');
409  hold on
410  scatter(Y_nodes(:,1),Y_nodes(:,2),'d','filled','g');
411  hold on
```

```matlab
412  scatter(M_nodes(:,1),M_nodes(:,2),'d','filled','r');
413  hold on
414  scatter(X_nodes(:,1),X_nodes(:,2),'d','filled','b');
415  hold on
416  for i = 1:length(nslices)
417  for ii = (sum(nslices(1:(i-1)))+1 : sum(nslices(1:i)))
418  if dimension_plan(ii, 5) ≥ width_limit_up
419  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 0 0], ...
         'LineWidth', 2)
420  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(9/10)) & dimension_plan(ii, 5) <width_limit_up
421  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 0 1], ...
         'LineWidth', 2)
422  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(8/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(9/10))
423  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 0.5 1], ...
         'LineWidth', 2)
424  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(7/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(8/10))
425  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 1 1], ...
         'LineWidth', 2)
426  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(6/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(7/10))
427  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 1 0.5], ...
         'LineWidth', 2)
428  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(5/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(6/10))
429  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0 1 0], ...
         'LineWidth', 2)
430  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(4/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(5/10))
431  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [0.5 1 0], ...
         'LineWidth', 2)
432  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(3/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(4/10))
433  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [1 1 0], ...
         'LineWidth', 2)
434  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(2/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(3/10))
435  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [1 0.5 0], ...
         'LineWidth', 2)
436  elseif dimension_plan(ii, 5) ≥(width_limit_down + ...
         (width_limit_up-width_limit_down)*(1/10)) & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(2/10))
437  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [1 0 0], ...
         'LineWidth', 2)
438  elseif dimension_plan(ii, 5) ≥width_limit_down & dimension_plan(ii, 5) ...
         <(width_limit_down + (width_limit_up-width_limit_down)*(1/10))
439  plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [1 0 0.5], ...
         'LineWidth', 2)
```

```matlab
440    elseif dimension_plan(ii, 5) ≤ width_limit_down
441    plot(dimension_plan(ii, [1 3]), dimension_plan(ii, [2 4]), 'color', [1 0 1], ...
           'LineWidth', 2)
442    end
443
444    end
445    end
446
447
448    figure()
449    gplot(A1,XY,'k')
450    pbaspect([1 1 1])
451    axis equal
452    xtickformat('%g m')
453    ytickformat('%g m')
454    hold on
455    grid on
456    scatter(I_nodes(:,1),I_nodes(:,2),'d','filled','y');
457    hold on
458    scatter(Y_nodes(:,1),Y_nodes(:,2),'d','filled','g');
459    hold on
460    scatter(M_nodes(:,1),M_nodes(:,2),'d','filled','r');
461    hold on
462    scatter(X_nodes(:,1),X_nodes(:,2),'d','filled','b');
463    hold on
464    for i = 1:sum(nslices)
465    plot(ellipse_plan(i*2 -1, :), ellipse_plan(i*2, :));
466    %uncomment to show the long and short axis of the
467    %ellipses
468    plot(ver_line_plan(i*2 -1, :), ver_line_plan(i*2, :));
469    plot(horz_line_plan(i*2 -1, :), horz_line_plan(i*2, :));
470    end
471
472    %in order to extract the ellipses parameters of a specific segment
473    %uncomment the following lines specifying the segment number. If the first
474    %segment's parameters have to be extracted, leave only 'segment_number'
475    %inside the brackets.
476
477    % segment_number = x;
478    % a_s = a_s(sum(nslices(1:(segment_number - 1)))+1:sum(nslices(1:segment_number)));
479    % b_s = b_s(sum(nslices(1:(segment_number - 1)))+1:sum(nslices(1:segment_number)));
480    % X0_s = X0_s(sum(nslices(1:(segment_number - 1)))+1:sum(nslices(1:segment_number)));
481    % Y0_s = Y0_s(sum(nslices(1:(segment_number - 1)))+1:sum(nslices(1:segment_number)));
482    % long_axis_s = long_axis_s(sum(nslices(1:(segment_number - ...
           1)))+1:sum(nslices(1:segment_number)));
483    % short_axis_s = short_axis_s(sum(nslices(1:(segment_number - ...
           1)))+1:sum(nslices(1:segment_number)));
484    % error_interpolation_s = error_interpolation_s(sum(nslices(1:(segment_number - ...
           1)))+1:sum(nslices(1:segment_number)));
```

F.2. Function ROI

```matlab
1  function [Ui, Vi, left_corner, right_corner, P_1, nslices, section_points, l_segment, ...
       P_1i, dist_points] = ROI (segm_check, width_large, epsilon, width_small, min_z, ...
       max_z)
2  %% With this function it is possible to obtain all the parameters that we
3  %need for the pointcloud subsampling, as well as it divides equally and
4  %simmetrically all the segments we are interested in
5  %%
6  for i = 1:length(segm_check)
7      segment_i(i, [1 2]) = segm_check(i, [1 2]);
8      segment_i(i, [3 4]) = segm_check(i, [3 4]);
9  % Sort from xmin to xmax:
10 if segment_i(i, 3) > segment_i(i, 1)
11     x1_i(i) = segment_i(i, 1);
12     y1_i(i) = segment_i(i, 2);
13     x2_i(i) = segment_i(i, 3);
14     y2_i(i) = segment_i(i, 4);
15 else
16     x1_i(i) = segment_i(i, 3);
17     y1_i(i) = segment_i(i, 4);
18     x2_i(i) = segment_i(i, 1);
19     y2_i(i) = segment_i(i, 2);
20 end
21
22 % For simplicity store left and right points of segment:
23 P_1i = [x1_i', y1_i']';
24 P_2i = [x2_i', y2_i']';
25 end
26 Ui = zeros(2, length(P_1i));
27 %UU = P_2i - P_1i;
28
29 % Find vector parallel to line ||
30 for ii = 1:length(P_1i)
31 UU = P_2i - P_1i;
32 Ui(:, ii) = UU(:, ii) ./ norm(UU(:, ii));  % unit vector
33 end
34
35 % If A is singular, that x and y are on the line.
36 % Therefore, find determinant of A and rewrite for a*x + b*y + c = 0
37 ai = y1_i -  y2_i;
38 bi = x2_i -  x1_i;
39 ci = x1_i.*y2_i -  x2_i.*y1_i;
40 Vi = zeros(2, length(P_1i));
41
42 %Vp = [ai', bi']';
43 % Find vector perpendicular to line _|_
44 for ii = 1:length(P_1i)
45 Vp = [ai', bi']';
46 Vi(:, ii) = Vp(:, ii) ./ norm(Vp(:, ii));  % unit vector
47 end
48
49 %% Divide each segment every ∆_s
50 %since we want the points of the slices equally and symmetrically spaced
51 %inside a segment we define t and fact
```

```matlab
52  l_segment = ((P_2i(1, :) - P_1i(1, :)).^2 + (P_2i(2, :) - P_1i(2, :)).^2) .^ (1/2);
53  Δ_s = 1.5; %[m]
54  t = (rem(l_segment, Δ_s) / 2) ./ l_segment;
55
56  nslices = floor(l_segment/Δ_s);
57  point_t = P_1i + t .* UU;
58  UU = UU(:);
59
60  for ii = 1 : length(segment_i)
61  slices(ii, 1 : nslices(:, ii)) = 1 : nslices(:, ii);
62  end
63
64  fact = zeros(length(segment_i), length(slices(1, :)));
65  eq_points = zeros(length(point_t), length(slices(1, :)));
66  for ii = 1 : length(segment_i)
67  fact(ii, :) = slices(ii, :)/l_segment(:, ii);
68  end
69
70  dist_points = ((P_1i(1, :) - point_t(1, :)).^2 + (P_1i(2, :) - point_t(2, :)).^2) .^ ...
         (1/2);
71  point_t = point_t(:);
72  % [rr, cc] = find(¬fact);
73  % z = [rr, cc];
74  ii = 1 : length(segment_i);
75  iii = linspace(1, length(point_t) - 1, length(l_segment));
76  eq_points(iii, :) = point_t(iii, 1) + Δ_s * fact(ii, :) .* UU(iii, 1);
77  eq_points(iii+1, :) = point_t(iii+1, 1) + Δ_s * fact(ii, :) .* UU(iii+1, 1);
78
79  section_points = [point_t, eq_points];
80
81  for ii = linspace(1, length(section_points(:, 1))-1, length(section_points(:, 1))/2)
82  for i = 1 : (length(section_points(1, :)) - 1)
83  if section_points([ii ii+1], 1 + i) == section_points([ii ii+1], 1)
84  section_points([ii ii+1], i+1) = NaN;
85  end
86  end
87  end
88
89  %now that we have all the points we can find the starting and the ending
90  %point of each slice
91  x1 = section_points(linspace(1, length(section_points(:, 1)) - 1, length(nslices)), :);
92  y1 = section_points(linspace(2, length(section_points(:, 1)), length(nslices)), :);
93  for i = 1:length(segm_check)
94      x1(i, nslices(1, i)+1) = NaN;
95      y1(i, nslices(1, i)+1) = NaN;
96  end
97
98  x2 = section_points(linspace(1, length(section_points(:, 1)) - 1, length(nslices)), :);
99  y2 = section_points(linspace(2, length(section_points(:, 1)), length(nslices)), :);
100 x2(:, 1) = NaN;
101 y2(:, 1) = NaN;
102
103 P_1 = zeros(length(section_points(:, 1)), length(x1(1, :)));
104 P_1(linspace(1, length(section_points(:, 1)) - 1, length(nslices)), :) = x1;
105 P_1(linspace(2, length(section_points(:, 1)), length(nslices)), :) = y1;
106 P_1 = P_1(:, 1:length(slices(1, :)));
107
```

```matlab
108  P_2 = zeros(length(section_points(:, 1)), length(x1(1, :)));
109  P_2(linspace(1, length(section_points(:, 1)) - 1, length(nslices)), :) = x2;
110  P_2(linspace(2, length(section_points(:, 1)), length(nslices)), :) = y2;
111  P_2 = P_2(:, 2:length(slices(1, :)) + 1);
112
113  x1 = x1(:, 1:length(slices(1, :)));
114  y1 = y1(:, 1:length(slices(1, :)));
115  x2 = x2(:, 2:length(slices(1, :)) + 1);
116  y2 = y2(:, 2:length(slices(1, :)) + 1);
117
118  %%
119  rows = [1 2; [1 2] + [2 2] .*(linspace(1, length(nslices) -1, length(nslices)-1)')];
120  % Find left corner of region of interest:
121  left_corner = zeros(length(P_1(:, 1)) * 1.5, length(P_1(1, :)));
122  P_VUl = zeros(length(P_1(:, 1)), length(P_1(1, :)));
123  V_Ul = zeros(2, length(nslices));
124   for ii = 1:length(nslices)
125        V_Ul(:, ii) = Vi(:, ii).* (width_large + epsilon) - Ui(:, ii) .* width_small;
126   end
127
128
129  for i = 1:length(nslices)
130
131      P_VUl(rows(i, :), :) = P_1(rows(i, :), :) + V_Ul(:, i);
132  end
133
134  left_corner([1 2; [1 2] + [3 3] .*(linspace(1, length(nslices) -1, ...
          length(nslices)-1)')]', :) = P_VUl;
135  left_corner(linspace(3, length(left_corner), length(nslices)), :) = min_z;
136
137  % Find right corner of region of interest:
138  right_corner = zeros(length(P_1(:, 1)) * 1.5, length(P_1(1, :)));
139  P_VUr = zeros(length(P_1(:, 1)), length(P_1(1, :)));
140  V_Ur = zeros(2, length(nslices));
141   for ii = 1:length(nslices)
142        V_Ur(:, ii) = -Vi(:, ii).* (width_large + epsilon) + Ui(:, ii) .* width_small;
143   end
144
145  for i = 1:length(nslices)
146
147       P_VUr(rows(i, :), :) = P_1(rows(i, :), :) + V_Ur(:, i);
148  end
149  right_corner([1 2; [1 2] + [3 3] .*(linspace(1, length(nslices) -1, ...
          length(nslices)-1)')]', :) = P_VUr;
150  right_corner(linspace(3, length(right_corner), length(nslices)), :) = max_z;
```

F.3. Function SEGM

```matlab
1  function [segm_check, length_segm_M, length_segm_I, length_segm_Y, length_segm_X] = ...
       SEGM (F, M_nodes, I_nodes, Y_nodes, X_nodes)
2  %% This function allows to isolate only the the segments we want to study, which are ...
       the segments linked by I-M, I-I nodes.
3  %%
4  F_r = round(F * 10^4)/10^4;
```

```matlab
5   M_nodes = round(M_nodes * 10^4)/10^4;
6   I_nodes = round(I_nodes * 10^4)/10^4;
7   Y_nodes = round(Y_nodes * 10^4)/10^4;
8   X_nodes = round(X_nodes * 10^4)/10^4;
9
10  %Find the indexes of M, I, Y, X nodes
11  F_M_idx12 = find(ismember(F_r(:, [1 2]), M_nodes, 'rows'));
12  F_I_idx12 = find(ismember(F_r(:, [1 2]), I_nodes, 'rows'));
13  F_Y_idx12 = find(ismember(F_r(:, [1 2]), Y_nodes, 'rows'));
14  F_X_idx12 = find(ismember(F_r(:, [1 2]), X_nodes, 'rows'));
15
16  F_M_idx34 = find(ismember(F_r(:, [3 4]), M_nodes, 'rows'));
17  F_I_idx34 = find(ismember(F_r(:, [3 4]), I_nodes, 'rows'));
18  F_Y_idx34 = find(ismember(F_r(:, [3 4]), Y_nodes, 'rows'));
19  F_X_idx34 = find(ismember(F_r(:, [3 4]), X_nodes, 'rows'));
20
21  %find the length of all the segments linked by M, I, Y or X nodes
22
23  M_col = unique(vertcat(F_M_idx12, F_M_idx34));
24  I_col = unique(vertcat(F_I_idx12, F_I_idx34));
25  Y_col = unique(vertcat(F_Y_idx12, F_Y_idx34));
26  X_col = unique(vertcat(F_X_idx12, F_X_idx34));
27
28  length_segm_M = ((F([M_col], 3) - F([M_col], 1)).^2 + (F([M_col], 4) - F([M_col], ...
        2)).^2) .^ (1/2);
29  length_segm_I = ((F([I_col], 3) - F([I_col], 1)).^2 + (F([I_col], 4) - F([I_col], ...
        2)).^2) .^ (1/2);
30  length_segm_Y = ((F([Y_col], 3) - F([Y_col], 1)).^2 + (F([Y_col], 4) - F([Y_col], ...
        2)).^2) .^ (1/2);
31  length_segm_X = ((F([X_col], 3) - F([X_col], 1)).^2 + (F([X_col], 4) - F([X_col], ...
        2)).^2) .^ (1/2);
32  %create a matrix with the degrees of the nodes
33  length_segm_M(:, 2) = 1;
34  length_segm_I(:, 2) = 2;
35  length_segm_Y(:, 2) = 3;
36  length_segm_X(:, 2) = 4;
37
38  F_r([F_M_idx12]', [1 2]) = 1;
39  F_r([F_M_idx34]', [3 4]) = 1;
40  F_r([F_I_idx12]', [1 2]) = 1;
41  F_r([F_I_idx34]', [3 4]) = 1;
42  F_r([F_Y_idx12]', [1 2]) = 0;
43  F_r([F_Y_idx34]', [3 4]) = 0;
44  F_r([F_X_idx12]', [1 2]) = 0;
45  F_r([F_X_idx34]', [3 4]) = 0;
46
47  check = all(F_r(:, [1 2 3 4]), 2);
48
49  % Create a matrix with the segments that we want to analyze
50  F_1 = F(:, 1);
51  F_2 = F(:, 2);
52  F_3 = F(:, 3);
53  F_4 = F(:, 4);
54  segm_check(:, 1) = F_1(logical(check), :);
55  segm_check(:, 2) = F_2(logical(check), :);
56  segm_check(:, 3) = F_3(logical(check), :);
57  segm_check(:, 4) = F_4(logical(check), :);
```

F.4. Function PROJ

```
1   function [R, Q_1, normal, angle_deg] = PROJ (Ui, left_corner, P_1, Vi, width_large, ...
        epsilon, max_z)
2   %% With this function we can calculate parameters useful for the subpointclouds ...
        projections
3   % First find principle coordinate to rotate to, in this case we
4   % rotate w.r.t. y_axis vector [0, 1]':
5   % REASON: We are searching wide in the x-direction (first coordinate), so
6   % we want to orient the pointcloud and search box w.r.t y-axis.
7   ref_vec = [0, 1]';
8   ref_vec = ref_vec / norm(ref_vec);  % unit vector
9
10  % Calculate angle between || and reference vector:
11  angle = zeros(1, length(Ui));
12  for i = 1:length(Ui)
13  angle(:, i) = acos( ref_vec' * Ui(:, i));
14  angle_deg = angle * 180 / pi;
15  end
16
17  R(:, [1 2; [1 2] + [2 2] .*(linspace(1, length(Ui) -1, length(Ui)-1)')]) = [cos(angle), ...
        -sin(angle);
18      sin(angle), cos(angle)];
19
20  % Find plane on which to project:
21  Q_1 = zeros(size(left_corner));
22  Q_2 = zeros(size(left_corner));
23  Q_3 = zeros(size(left_corner));
24  rows = [1 2; [1 2] + [2 2] .*(linspace(1, length(Ui) -1, length(Ui)-1)')];
25  rows_b = [1 2; [1 2] + [3 3] .*(linspace(1, length(Ui) -1, length(Ui) -1)')];
26
27  for i = 1:length(Ui)
28
29  Q_1(rows_b(i, :), :) = P_1(rows(i, :), :) + Vi(:, i) * (width_large + epsilon);
30  Q_2(rows_b(i, :), :) = P_1(rows(i, :), :);
31  Q_3(rows_b(i, :), :) = Q_1(rows_b(i, :), :);
32  Q_3(linspace(3, length(left_corner), length(left_corner)/3), :) = max_z;
33
34  end
35
36   % Get normal to find the plane
37  normal = zeros(length(left_corner), 1);
38  r_1 = linspace(1, length(left_corner) - 2, length(Ui));
39  r_2 = linspace(2, length(left_corner) - 1, length(Ui));
40  r_3 = linspace(3, length(left_corner), length(Ui));
41  normal(r_1) = (Q_2(r_2, 1) - Q_1(r_2, 1)).*(Q_3(r_3, 1) - Q_1(r_3, 1)) - (Q_2(r_3, 1) - ...
        Q_1(r_3, 1)).*(Q_3(r_2, 1) - Q_1(r_2, 1));
42  normal(r_2) = (Q_2(r_3, 1) - Q_1(r_3, 1)).*(Q_3(r_1, 1) - Q_1(r_1, 1)) - (Q_2(r_1, 1) - ...
        Q_1(r_1, 1)).*(Q_3(r_3, 1) - Q_1(r_3, 1));
43  normal(r_3) = (Q_2(r_1, 1) - Q_1(r_1, 1)).*(Q_3(r_2, 1) - Q_1(r_2, 1)) - (Q_2(r_2, 1) - ...
        Q_1(r_2, 1)).*(Q_3(r_1, 1) - Q_1(r_1, 1));
44  for i = 1:length(angle)
```

```
45   normal([i*3-2 i*3-1 i*3], 1) = normal([i*3-2 i*3-1 i*3]) ./ norm(normal([i*3-2 i*3-1 ...
         i*3])); % unit vector
46   end
```

Bibliography

[1] V. Andreychouk. Opredeleniye antropogennogo karsta (definition of the antropogenetic karst). 1991.

[2] Viacheslav Andreychouk, Yuri Dublyansky, Ezhov Y., and Lysenin G. *Karst in the Earth's Crust: its distribution and principal types*. 01 2009. ISBN 978-83-87431-93-8.

[3] Ellie Ardakani, Douglas Schmitt, and T.D. Bown. Devonian grosmont formation surface investigation using legacy high resolution seismic profiles, ne alberta, canada. 06 2013. doi: 10.3997/2214-4609.20130584.

[4] Ph Audra and Arthur Palmer. Research frontiers in speleogenesis. dominant processes, hydrogeological conditions and resulting cave patterns. *Acta Carsologica*, 44, 12 2015. doi: 10.3986/ac.v44i3.1960.

[5] Mateus Basso, Michelle Kuroda, Luis Afonso, and Alexandre Vidal. Three-dimensional seismic geomorphology of paleokarst in the cretaceous macaÉ group carbonates, campos basin, brazil. *Journal of Petroleum Geology*, 41:513–526, 10 2018. doi: 10.1111/jpg.12719.

[6] S.V.F. Borges, F. Balsamo, M.M. Vieira, P. Iacumin, N.K. Srivastava, F. Storti, and F.H.R. Bezerra. Pedogenic calcretes within fracture systems and beddings in neoproterozoic limestones of the irecê basin, northeastern brazil. *Sedimentary Geology*, 341:119 – 133, 2016. ISSN 0037-0738. doi: https://doi.org/10.1016/j.sedgeo.2016.05.012. URL http://www.sciencedirect.com/science/article/pii/S0037073816300847.

[7] Stephen Ehrenberg and P H Nadeau. Sandstone vs. carbonate petroleum reservoirs: A global perspective on porosity-depth and porosity-permeability relationships. *AAPG Bulletin*, 89:435–445, 04 2005. doi: 10.1306/11230404071.

[8] Renata A. Ennes-Silva, Francisco H.R. Bezerra, Francisco C.C. Nogueira, Fabrizio Balsamo, Alexander Klimchouk, Caroline L. Cazarin, and Augusto S. Auler. Superposed folding and associated fracturing influence hypogene karst development in neoproterozoic carbonates, são francisco craton, brazil. *Tectonophysics*, 666:244 – 259, 2016. ISSN 0040-1951. doi: https://doi.org/10.1016/j.tecto.2015.11.006. URL http://www.sciencedirect.com/science/article/pii/S0040195115006332.

[9] Andrew R. Farrant and Peter L. Smart. Role of sediment in speleogenesis; sedimentation and paragenesis. *Geomorphology*, 134(1):79 – 93, 2011. ISSN 0169-555X. doi: https://doi.org/10.1016/j.geomorph.2011.06.006. URL http://www.sciencedirect.com/science/article/pii/S0169555X11002893. Geomorphology and Natural Hazards in Karst Areas.

[10] Hervé Jourde, Fabien Cornaton, Séverin Pistre, and Pascal Bidaux. Flow behavior in a dual fracture network. *Journal of Hydrology*, 266(1):99 – 119, 2002. ISSN 0022-1694. doi: https://doi.org/10.1016/S0022-1694(02)00120-8. URL http://www.sciencedirect.com/science/article/pii/S0022169402001208.

[11] R. S. Kallweit and L. C. Wood. The limits of resolution of zero□phase wavelets. *GEOPHYSICS*, 47(7): 1035–1046, 1982. doi: 10.1190/1.1441367.

[12] Alexander Klimchouk. *Hypogene Speleogenesis: Hydrogeological and Morphogenetic Perspective, NCKRI Special Paper 1*, volume 1. 01 2007. ISBN 0-9795422-0-0.

[13] Alexander Klimchouk. Speleogenesis, hypogenic. *Encyclopedia of Caves*, pages 748–765, 12 2012. doi: 10.1016/B978-0-12-383832-2.00110-9.

[14] Alexander Klimchouk and Derek Ford. Lithological and structural controls of dissolutional cave development. pages 54 – 64, 04 2000.

[15] Alexander Klimchouk, Augusto S. Auler, Francisco H.R. Bezerra, Caroline L. Cazarin, Fabrizio Balsamo, and Yuri Dublyansky. Hypogenic origin, geologic controls and functional organization of a giant cave system in precambrian carbonates, brazil. *Geomorphology*, 253:385 – 405, 2016. ISSN 0169-555X. doi: https://doi.org/10.1016/j.geomorph.2015.11.002. URL http://www.sciencedirect.com/science/article/pii/S0169555X15301999.

[16] Le Li and Shaocheng Ji. A new interpretation for formation of orthogonal joints in quartz sandstone. *Journal of Rock Mechanics and Geotechnical Engineering*, 2020. ISSN 1674-7755. doi: https://doi.org/10.1016/j.jrmge.2020.08.003. URL http://www.sciencedirect.com/science/article/pii/S1674775520301268.

[17] N. Madhubabu, P.K. Singh, Ashutosh Kainthola, Bankim Mahanta, A. Tripathy, and T.N. Singh. Prediction of compressive strength and elastic modulus of carbonate rocks. *Measurement*, 88:202 – 213, 2016. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2016.03.050. URL http://www.sciencedirect.com/science/article/pii/S0263224116300276.

[18] Aroldo Misi and Ján Veizer. Neoproterozoic carbonate sequences of the una group, irecê basin, brazil: chemostratigraphy, age and correlations. *Precambrian Research*, 89(1):87 – 100, 1998. ISSN 0301-9268. doi: https://doi.org/10.1016/S0301-9268(97)00073-9. URL http://www.sciencedirect.com/science/article/pii/S0301926897000739.

[19] A.S.L. Montes, C.P. Gravenor, and M.L. Montes. Glacial sedimentation in the late precambrian bebedouro formation, bahia, brazil. *Sedimentary Geology*, 44(3):349 – 358, 1985. ISSN 0037-0738. doi: https://doi.org/10.1016/0037-0738(85)90019-3. URL http://www.sciencedirect.com/science/article/pii/0037073885900193.

[20] ODLING N.E, Paul Gillespie, Bernard Bourgine, CASTAING C, CHILES JP, CHRISTENSEN N.P, FILLION E, Albert Genter, OLSEN C, Lena Thrane, Robert Trice, AARSETH E.S, John J. Walsh, and WATTERSON J. Variations in fracture system geometry and their implications for fluid flow in fractured hydrocarbon reservoirs. *Petroleum Geoscience*, 5:373–384, 11 1999. doi: 10.1144/petgeo.5.4.373.

[21] Arthur Palmer. Origin and morphology of limestone caves. *Geol. Soc. Am. Bull.*, 103, 01 1991.

[22] Arthur Palmer. Distinction between epigenic and hypogenic caves. *Geomorphology*, 134:9–22, 11 2011. doi: 10.1016/j.geomorph.2011.03.014.

[23] Arthur Palmer. Sulfuric acid caves: Morphology and evolution. *Treatise on Geomorphology*, 6:241–257, 03 2013. doi: 10.1016/B978-0-12-374739-6.00133-0.

[24] Cayo Pontes, Francisco Bezerra, Giovanni Bertotti, F. Balsamo, Vincenzo La Bruna, and Stephan de Hoop. Karst conduits formed along fracture corridors in anticline hinges of carbonate units - implications for reservoir quality. 02 2020. doi: 10.13140/RG.2.2.24064.79365.

[25] Henrique V. Redivo, Ana M.P. Mizusaki, and Ana V.A. Santana. Ree patterns and trustworthiness of stable carbon isotopes of salitre formation, irecê basin (neoproterozoic), são francisco craton. *Journal of South American Earth Sciences*, 90:255 – 264, 2019. ISSN 0895-9811. doi: https://doi.org/10.1016/j.jsames.2018.11.030. URL http://www.sciencedirect.com/science/article/pii/S0895981118303122.

[26] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. Chapter 1 - skeletonization and its applications – a review. pages 3 – 42, 2017. doi: https://doi.org/10.1016/B978-0-08-101291-8.00002-X. URL http://www.sciencedirect.com/science/article/pii/B978008101291800002X.

[27] Orildo L. Silva, Francisco H.R. Bezerra, Rubson P. Maia, and Caroline L. Cazarin. Karst landforms revealed at various scales using lidar and uav in semi-arid brazil: Consideration on karstification processes and methodological constraints. *Geomorphology*, 295:611 – 630, 2017. ISSN 0169-555X. doi: https://doi.org/10.1016/j.geomorph.2017.07.025. URL http://www.sciencedirect.com/science/article/pii/S0169555X16307309.

[28] Fei Tian, Qiang Jin, Xinbian Lu, Yuhong Lei, Likuan Zhang, Songqing Zheng, Hongfang Zhang, Yuanshuai Rong, and Naigui Liu. Multi-layered ordovician paleokarst reservoir detection and spatial delineation: A case study in the tahe oilfield, tarim basin, western china. *Marine and Petroleum Geology*, 69:53 – 73, 2016. ISSN 0264-8172. doi: https://doi.org/10.1016/j.marpetgeo.2015.10.015. URL http://www.sciencedirect.com/science/article/pii/S0264817215301161.

[29] Ricardo I.F Trindade, Manoel S D'Agrella-Filho, Marly Babinski, Eric Font, and Benjamim B Brito Neves. Paleomagnetism and geochronology of the bebedouro cap carbonate: evidence for continental-scale cambrian remagnetization in the são francisco craton, brazil. *Precambrian Research*, 128(1):83 – 103, 2004. ISSN 0301-9268. doi: https://doi.org/10.1016/j.precamres.2003.08.010. URL http://www.sciencedirect.com/science/article/pii/S0301926803002638.

[30] Håvard Tveite. The QGIS line direction histogram plugin. http://plugins.qgis.org/plugins/LineDirectionHistogram/, 2015–2020.

[31] Alessandra Vasconcelos, Fernanda Souza, Joel Rodet, Cristiane Oliveira, and André Salgado. Karst developed in siliciclastic rocks at serra do espinhaço meridional (brazil). 07 2013.