Ingrid Suter
May 19th, 2020

Geoelectric data acquisition and modeling of dynamic hydrogeologic systems in Hawai‘i

Abstract
In this work, I study dynamic hydrogeologic systems in Hawai‘i on the islands of O‘ahu and Kaua‘i. These dynamic systems include a coastal salt making pond on the Island of Kaua‘i, a leeward stream valley on the Island of O‘ahu, and the possibility of contamination in a basalt ridge/sediment valley complex. My objective is to investigate these systems by the use of self-potential (SP) time series, 3D electrical resistivity tomography (ERT), and quasi time-lapse numerical modeling of electric resistivity in a model of contamination in a valley ridge complex. I illustrate how ERT electrode designs can be used to extract specific information from a study site, how SP can be used to track dynamic hydrogeologic systems in a ridge valley complex, and how numerical modeling can be both informative and deceiving. The combined research demonstrates how geoelectric data acquisition and modeling can be interwoven to help answer questions individual methods could not answer on their own.

1. Introduction

Hydrological systems are dynamic systems that vary in both space and time. Hawai‘i in particular, is well-known for its strong spatio-temporal gradients in for example rainfall, land-use, and hydrogeology (Oki, 2005). In this work, I focus on studying time-dependent hydrological systems on the Hawaiian Islands. I explore the use of both active and passive electrical geophysical methods to identify their capabilities and efficiency in studying dynamic hydrological systems, and how these methods can be used in conjunction with numerical modeling. Geoelectric methods such as self-potential (SP) and electrical resistivity tomography (ERT) are powerful, well-established tools for relatively non-invasive characterization and analyzation of subsurface properties (Blanchy et al., 2020). I use both the self-potential method, a passive geophysical method that measures the electrical potential difference induced via natural source current densities at depth (Sill, 1983; Corwin, 1997), and electrical resistivity tomography, an active geophysical method, which exploits human-made sources to inject electrical current into the ground and subsequently measure the resulting potential difference, with the goal to infer the electrical resistivity distribution in the subsurface (Lowrie, 2007). SP has been shown to track groundwater flow in volcanic settings (Barde-Cabusson et al., 2012) and crude oil contamination plumes (Giampaolo et al., 2014). Similarly, ERT has proven successful in finding freshwater drinking sources (Thiagarajan et al., 2018) as well as identifying contaminants that may affect drinking water supplies (Rao et al., 2014). In this work, I focus on theoretical concepts, numerical modeling, and field data acquisition and design. I have explored two study sites: a coastal study site (salt pond) on Kaua‘i (a traditional salt-making area), and a leeward stream valley-ridge complex on the island of O‘ahu. Both are time-dependent dynamic systems. The coastal study site, for example, suffers from seasonal flooding, and the degree of flooding varies depending on a variety of known and unknown controlling variables. The valley-ridge complex consists of complex geology and will respond dynamically to changes like variations in rainfall. The numerical modeling and inversion of ERT data was done to study the efficiency and sensitivity of the ERT method to study time-varying valley-ridge complexes, e.g., due to potential contamination

of the hydrologic system. Additionally, the use of synthetic data compensated the lack of sufficient field data for the valley-ridge study site due to the COVID-19 pandemic.

2. Study Sites

Hanapēpē Salt Pond, Kaua'i, Hawai'i.

The Hanapēpē salt pond is located on Kaua'i's west coast in the ahupua'a of Hanapēpē (Figure 1) (Pap et al., 2019). Salt harvesting is one of Hawai'i's oldest traditions and this salt pond is one of the last sites where this tradition has remained. In recent years this practice has been threatened by environmental impacts such as wave inundation, pollution, and flooding (Pap et al., 2019). In fact, these same issues, along with costal erosion and habitat loss, have been seen state-wide and are direct results of sea-level rise (Kane et al., 2015). The goal at this study site is to use ERT to obtain a 3D subsurface image of electrical resistivity beneath the salt pond. Information about regional differences in resistivity values and geometry of these regions can help determine the dominant flood-controlling geology, the severity of pollution and flooding and even where these sources might be coming from. I have created several field data acquisition designs for this study site. Unfortunately, the Hanapēpē salt pond remained flooded into the spring months and the actual field work could not yet be carried out.

Figure 1-(a) Location map of Kaua'i, salt pond denoted by blue pin. (b) Map view of Hanapēpē salt pond (Google Earth, 2020).

Moanalua Streamvalley - Ridge Complex, O'ahu, Hawai'i.

The Moanalua Streamvalley is located in leeward southeastern O'ahu (Figure 2). The complex is made up of porous basalt, weathered basalt clay, and infilled with both terrestrial and marine sediments that can act as barriers to fluid flow (Oki, 2005). Field work was carried out at the Moanalua Golf Club, which offered a unique experience to study dynamic hydrological changes in a ridge/valley fill complex. Identifying the origins of potentially observed changes in these dynamic systems, like groundwater variability due to rainfall and the possibility of contamination, were hoped to be located and tracked over the course of weekly SP sampling for two months. An SP profile was surveyed along the northern boundary of the golf course (Figure 2b), parallel to the axis of the valley and to the ridge/valley fill interface. This profile was chosen to have the longest profile length possible, while being as close to the ridge/valley boundary as possible. Only two weeks of the potentially eight-week survey were completed because of the global pandemic. This led to the transition of building numerical ERT models to study the sensitivity and resolvable

power of this method for what could have been representable scenarios for the Moanalua Streamvalley, and that will provide important information for future geophysics surveys of this study site. The idea would be to combine ERT and SP data for a valuable complementary analysis of this time-dependent hydrological system.

Figure 2-(a) Location map of O'ahu, Moanalua Valley denoted by blue pin. (b) Study site at Moanalua Golf Club, SP profile marked by red line from A to A' (Google Earth, 2020).

3. Methods

In this work, I explore the efficiency and practicability of two electrical geophysical methods for studying spatio-temporal hydrological systems: self-potential (SP) and electrical resistivity tomography (ERT). I also explore the usefulness and applicability of modeling a 2D ERT profile to study and analyze dynamic systems.

Self-Potential

The SP method measures the natural difference in electrical potential between two nonpolarizable electrodes (Revil, 2002). SP signals are generated through electrochemical coupling, thermoelectric coupling, and electrokinetic coupling, which is also referred to as the streaming potential (Barde-Cabusson et al., 2012). In our case, SP signals of electrokinetic origin are the main mechanism and are found to have different macroscopic and microscopic mechanisms (Grobbe and Barde-Cabusson, 2019). Macroscopic mechanisms include topographic effects and hydrothermal circulation, while microscopic effects include the electrical double layer (EDL) (Grobbe and Barde-Cabusson, 2019). The EDL exists at the surface between solid particles and the pore space and plays an important role in generating SP signals (Grobbe and Barde-Cabusson, 2019). Figure 3 illustrates the different layers found in the EDL and their role in the production of electrokinetic mechanisms. Grobbe and Barde-Cabusson (2019) state that in a porous medium containing fluid, a net charge is created on the mineral surface as a consequence of chemical reactions between the mineral surface and the pore water. The surface charge is counterbalanced by the combined effect of the Stern layer, which consists of absorbed ions on the mineral surface, and the diffuse layer, which together form the EDL (Figure 3; Revil, 2002). The presence of an EDL can cause electrokinetic phenomena like electroosmosis, where an electric field drives fluid flow, and the streaming potential, where a hydraulic gradient creates an electric field (Grobbe and Barde-Cabusson, 2019).

Revil (2002) explains how SP anomalies result from polarization mechanisms at depth. The drag of the positive charge of the EDL with the pore fluid flow generates a positive electrical potential in the flow direction which can be understood in terms of the electrical double (or triple) layer theory. Positive anomalies have also been seen where crude oil contamination is present (Giampaolo et al., 2014). Negative potential anomalies can arise through gravitational flow, i.e infiltration of fluids. Revil (2002) states that electrokinetic mechanisms best explain the patterns, polarity, and intensity of SP anomalies.

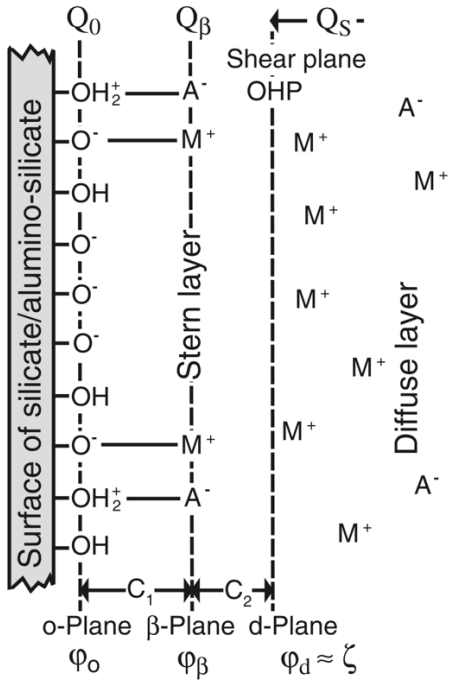


Figure 3-Diagram of EDL mechanism. The surface of a mineral interacts with the Stern layer which consists of absorbed ions on the mineral surface. The place of zero relative velocity is called the shear plane and occurs somewhere between the Stern layer and the diffuse layer. The diffuse layer occurs in the fluid where dissolved ions are present (Revil, 2002; Grobbe and Barbe-Cabusson, 2019). Figure from: Revil (2002).

SP surveys are carried out by using two identical electrodes typically consisting of a metal rod submerged in a saturated solution of its own salt. In our case, the electrodes were filled with a saturated copper sulfate solution that surrounds a copper rod. The combination is contained in a PVC plastic tube closed by a plastic plug on one side and by a porous ceramic tip on the other side, to allow electrolytes to flow through and make electrical contact with the ground. A high impedance voltmeter (i.e. 10-100 MΩ) is needed to measure the natural potential difference between the two electrodes, along with the resistance to confirm a good electrical contact, which are connected by a wire (Grobbe and Barde-Cabusson, 2019).

Lowrie (2007) describes how SP surveys are done by one of two common methods as follows: (1) The gradient method employs a fixed separation between the two electrodes and the pair is moved forward along the survey after each measurement is taken. The trailing electrode takes the place of the leading electrode, and to minimize error, a “leapfrog” technique is utilized where the leading electrode now becomes the trailing electrode and so on. Cumulative error is a big disadvantage when using a fixed electrode configuration. (2) In the total field method, the base station electrode remains fixed while the second electrode is mobile, progresses down the profile, and is used to take measurements. The total field method results in a smaller cumulative error than the gradient method and is usually the preferred method. We used the total field method in our SP

surveys with an electrode spacing of five meters and a wire length of 300 meters. Three sections were completed to produce a 900 meter profile length. The locations of the measurements were recorded with a handheld Garmin etrex 20 GPS using a Universal Transverse Mercator (UTM) coordinate system and a WGS84 Zone 4N reference frame. The data were processed using spreadsheet software. In order to get continuous potential difference values, each new base station must be normalized to the previous potential difference value. To do this, the values of a new base station must be added to the last value of the prior base station; this must occur at each new base station; the so-called reference correction (Revil and Jardani, 2013). These new normalized SP values are then plotted versus distance.

The original plan was to carry out an identical SP survey profile every week, for a total of eight weeks. We would place electrodes at the exact locations as the week prior, denoted by a marker to ensure precision. Placing the electrodes in the exact five meter spacing each week would provide a decent guarantee that any variability in the SP survey would be because of changes in the hydrology of the subsurface and not because of variability in the measurements. The COVID-19 pandemic led to the shutdown of all nonessential work which meant two weeks into this campaign I had to redirect my focus to a new way of studying dynamic systems in Hawai‘i. We decided to use numerical modeling to study a time-varying hydrological system in a valley-ridge complex similar to Moanalua Valley. We have opted for a 2D ERT profile similar to assess the sensitivity and resolvable power of ERT for such time-varying systems in Hawai‘i. Ultimately, the ERT and SP data could be used together to provide complementary information on both the dominant hydrogeology as well as flow paths and flow velocities.

Electrical Resistivity Tomography

In contrast to relying on natural currents, electrical resistivity tomography (ERT) uses two electrodes to supply an actively induced current into the ground while the resulting electrical potential difference is measured across two different electrodes (Daily et al., 2004). These lines of current flow adapt to the resistivity patterns of the subsurface geology (Lowrie, 2007). ERT allows for imaging of the subsurface geological formations and structures in both the horizontal and vertical directions along a survey line (Thiagarajan et al., 2018). A direct current can cause charges to accumulate on the potential electrodes which can result in false signals (Lowrie, 2007). Lowrie sates that because of this, it is a common practice to commutate the direction of the direct current every few seconds so that charge won’t build up. ERT has been used across the globe to study groundwater and complex hydrogeologic setups, geothermal reservoirs, and waste disposal (Thiagarajan et al., 2018). Another major use of ERT has been to locate and assess the safety of drinking water for vulnerable populations (Rao et al., 2014).

ERT surveys can be executed using multiple different configurations; Wenner, Schlumberger, dipole-dipole, and pole-dipole configurations are explained below. Lowrie (2007) describes the best configuration for different applications as summarized hereafter. The Wenner configuration is best adapted to lateral profiling where the current and potential electrode pairs have a common midpoint and the distances between adjacent electrodes are equal. This survey configuration reveals the horizontal variations in resistivity within an area at a relatively shallow depths. The Schlumberger configuration is commonly used in vertical electrical sounding (VES), where the goal is to observe the variation of resistivity at deeper depths. A Schlumberger configuration consists of the current and potential pairs of electrodes having a common midpoint, but the distances between adjacent electrodes differ. When the midpoint is kept fixed while the distance between the current electrodes is progressively increased, the current lines penetrate to greater depths. In a dipole-dipole configuration the spacing of the electrodes in each pair is a, and

the distance between their midpoints is L, which is typically much larger than a. The dipole-dipole configuration is commonly used due to its practicality and balance of imaging resolution between shallow and deep depths, which is why this configuration was chosen for the numerical models. The pole-dipole configuration places one electrode at theoretical infinity and the other pole electrodes in the area to be imaged. Its benefits are a greater depth of investigation. Furthermore, this electrode configuration will be used during the Hanapēpē campaign because of its ease to deploy in the field for a 3D ERT survey.

I used the software package FullWave Designer to build field data acquisition designs for the Hanapēpē study site (Figures 4 & 5). Parameters such as grid size, orientation, and number of electrodes all had to be adjusted through trial-and-error to obtain designs that both fit the surface of the site as well as offer optimal subsurface coverage and depth for the 3D image. Two different designs with the same number of electrodes have been created due to the environmental sensitivity of the salt making practice, where some areas may be off limits (Figures 4 & 5). These designs have not been executed at the field site yet due to prolonged spring flooding and complications due to COVID-19. However, once this project picks up in the future these designs and/or updates of the designs will be used.

Figure 4-Orange CA electrodes are pole electrodes with CB fixed at infinity, blue electrodes are V-FullWaver electrodes which measure in two directions. (a) Configuration 1-evenly spaced electrodes. (b) Electrodes confined to the perimeter of the salt pond.

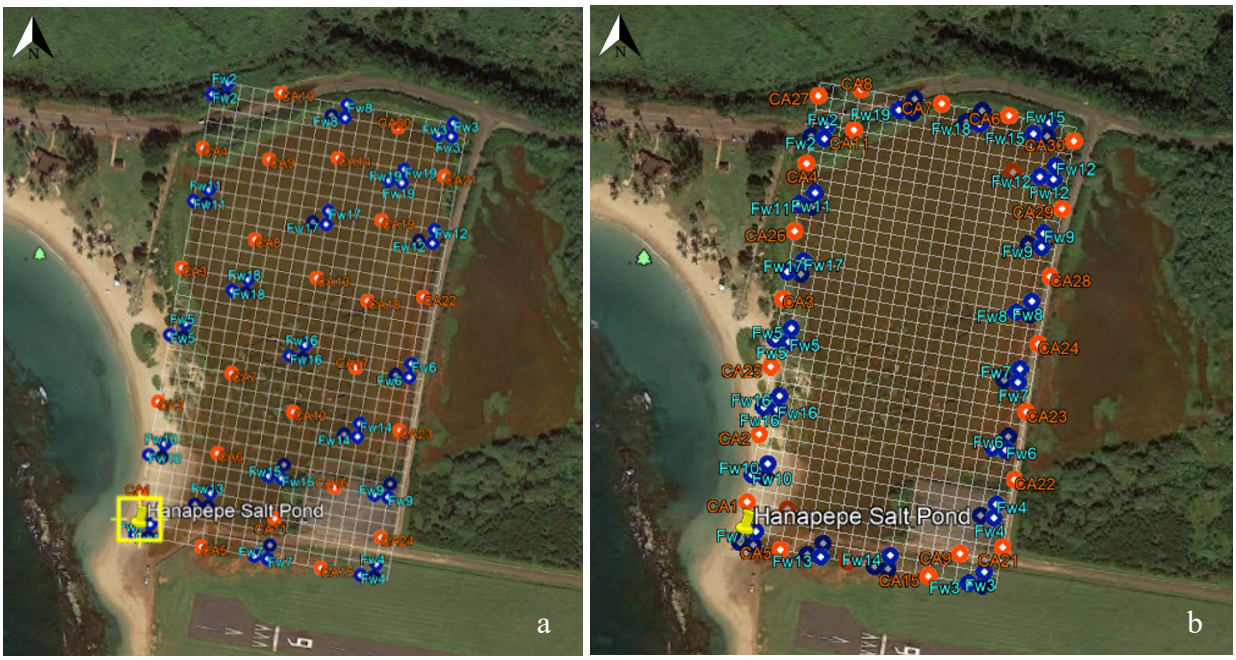


Figure 5-(a) Configuration 1 at location in map view. (b)) Configuration 2 at location in map view.

When using a four-electrode configuration for ERT surveys to assess the apparent resistivity values, the ground is assumed to be uniform and homogeneous. However, the true geologic substrate is made up of different lithologies and structures (Lowrie, 2007). Because of this, the results of the initial measurements give the apparent resistivity and do not represent the true resistivity of any part of the ground (Lowrie, 2007). Current lines behave like optical or seismic rays when they encounter boundaries of different resistivities, reflecting and refracting (Lowrie, 2007). In order to obtain the true resistivity structure along the current flow line, an inversion procedure is applied that produces a model that gives an "acceptable" fit to the data and satisfies defined constraints (Daily et al., 2004). A forward model yields transfer impedances that are compared to the measured data (Daily et al., 2004). The model is adjusted until the fitting criteria and constraints are met (Daily et al., 2004). All of these procedures were carried out in this study using the open-source numerical modeling software package ResIPy.

Numerical Modeling-ResIPy

Not only can ERT be measured in the field, but it can also be generated synthetically using modeling software. Blanchy et al. (2020) created ResIPy, an open source program that uses a Python application programming interface (API) and a graphical user interface (GUI) to produce 2D and 3D geoelectric models. ResIPy is an intuitive, user-friendly approach to inversion of geoelectric data using R2 inversion codes for ERT. Geoelectric data can be uploaded or forward modeling can be used to generate synthetic data given a synthetic geoelectric model. The Python API is object-orientated and has several classes (Figure 6). The main class is R2 (R2.py) and it manages the data processing and inversion. ResIPy is also capable of rigorous data cleaning and quality control which can be done automatically or with user control. Another class is the survey class which handles one data set for one survey. And finally, the mesh class handles tasks associated with the construction of the finite element mesh. Blanchy et al. (2020) allowed for two types of 2D finite element meshes: structured quadrilateral and unstructured triangular. The triangular mesh is recommended because it is more versatile, can account for complex topography,

and is computationally more efficient. Mesh elements tend to be finer near the electrodes and get courser at greater distances. This is to address the need for greater discretization in areas of high potential gradient. In addition, Blanchy et al. (2020) implemented a GUI, which is another import part of ResIPy and consists of a series of tabs that allows a non-linear workflow and takes the user through necessary stages of importing and filtering data (or creating synthetic data for forward modeling), generating a mesh, and inverting data. For the purpose of my work I chose to focus on using the API opposed to the GUI in order to have better control over the models and ensure reproducibility (see the Python Notebooks in the Appendices).

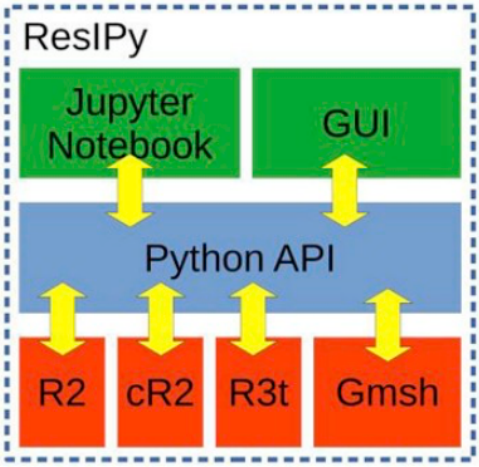


Figure 6-Flow chart of ResIPy internal structure. Visualization layer in green, Python API in charge of calling executables in blue, and compiled executables in red (Blanchy et al., 2020).

ResIPy was chosen to continue my investigation of dynamic systems at Moanalua Valley by forward modeling synthetic data. I decided to model and invert 2D ERT synthetic data for the study site, to focus on the time-varying response due to a potential contamination source originating in a ridge. The aim was to see if a contamination source could be identified in a basalt ridge/sediment valley fill complex over time and if that "contamination" could be differentiated from, e.g., infiltration of water or water diluted with some contaminant.

Several parameters were used to create these models and are identical between models unless specified otherwise. I used 36 electrodes at a spacing of 20 meters which gives a profile length of 700 meters. Using the rule of thumb that the depth of investigation is equal to one fifth of the profile length, the depth of investigation for these models is roughly 140 meters. Resistivity values used for the different geological regions are as seen in the table below.

Geology	Resistivity (Ωm)
Unsaturated dry basalt	10,000
Oil saturated basalt	4,000

Freshwater saturated basalt	2,000
Valley sediment	500
Saprolite	50

Table 1-Geology in models and corresponding resistivity values (Barde-Cabusson, 2007; Lowrie, 2007).

The goal was to build models that portrayed a leeward O'ahu stream valley while also being friendly to the model. This may result in geometry, geology, or resistivity values that are not 100% representative but is to ensure the model runs smoothly and contrasts are identifiable for the purpose of our study. A more advanced model would require integrating geological information from a particular study site. Producing simplified models is essential to identify the parameters playing the most important roles.

I built a total of seven different models with orientations perpendicular to the axis of a valley. The purpose of this was to model what contamination would look like originating from the basaltic ridge as it flows downward into an aquifer. Placed into the context of Moanalua Valley, this would correspond to a cross-section perpendicular to the SP profile acquired. Using the same orientation as the SP profile would only capture the contamination once it reached a certain distance down the ridge into the valley; this is not what I wanted to model in this particular case. Instead, I wanted to model the time sequence from the beginning of contamination at the top of the ridge until the contamination reached the freshwater aquifer.

The seven different ERT models include: one of the stream valley with no contamination, three models where the contamination begins in the ridge and ends in the aquifer, and three models of the same time series as those of the contamination, but where the contamination is simply freshwater infiltration. The main goals were to see if 2D ERT time series can identify and track contamination, what the resolvability of key features of the system is, and if this method can truly distinguish the difference between contamination and water infiltration.

4. Results

3D ERT at Hanapēpē

Figure 7 shows the subsurface measurements of both Configuration 1 and 2. The location of the measurements will give the apparent resistivity values at that specific location. The two configurations give measurements at similar depths and at with similar coverage-densities. The aim is to have a balance of coverage between shallow depths to pick up potential clay layers, while also being able to penetrate deep enough below the salt pond to see the volcanic bedrock. It is reassuring to see similar coverage between the two configurations in case certain areas of the salt pond are restricted.

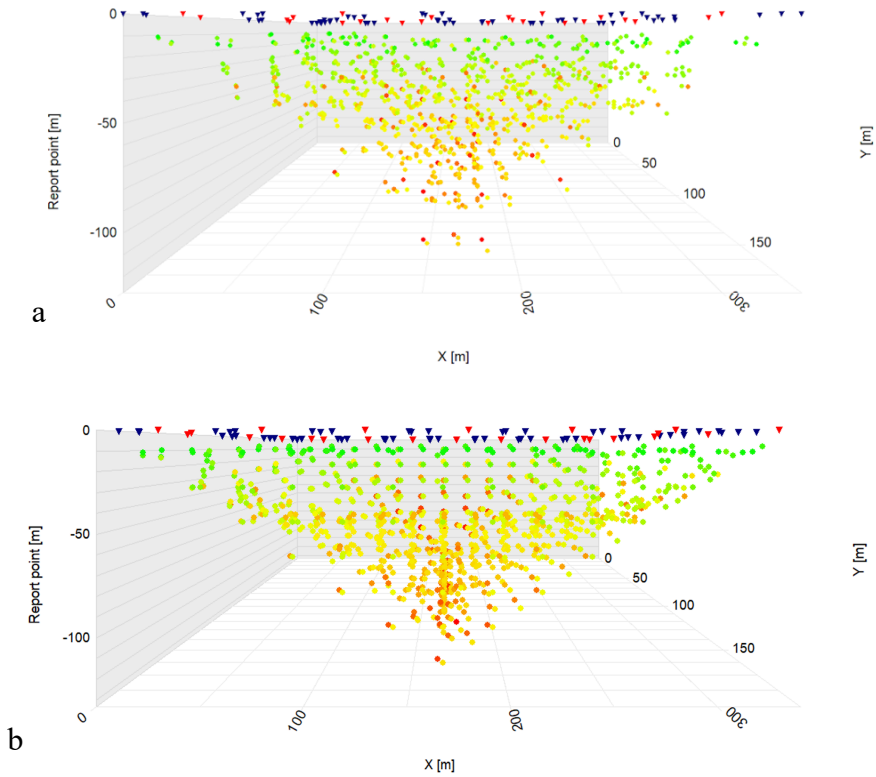


Figure 7-(a) Subsurface measurements of Configuration 1. (b) Subsurface measurements of Configuration 2.

Self-Potential at Moanalua Golf Club

Two weeks of SP data from the Moanalua Golf Club show slight changes in magnitude of potential difference values, however the periodicity of the two weeks is very similar (Figure 8). We have to consider that SP measures the difference of potential with respect to a reference and, in the case of Moanalua, the reference is inside the valley itself. This means that any change affecting the whole valley will not be visible in our data. Only contrasts within the area and preferential flows along the profile will be visible. A shift in SP value magnitudes between the two weeks begins around 250 meters into the profile length. This coincides with an increase in topography of about 20 meters on the golf course along our profile.

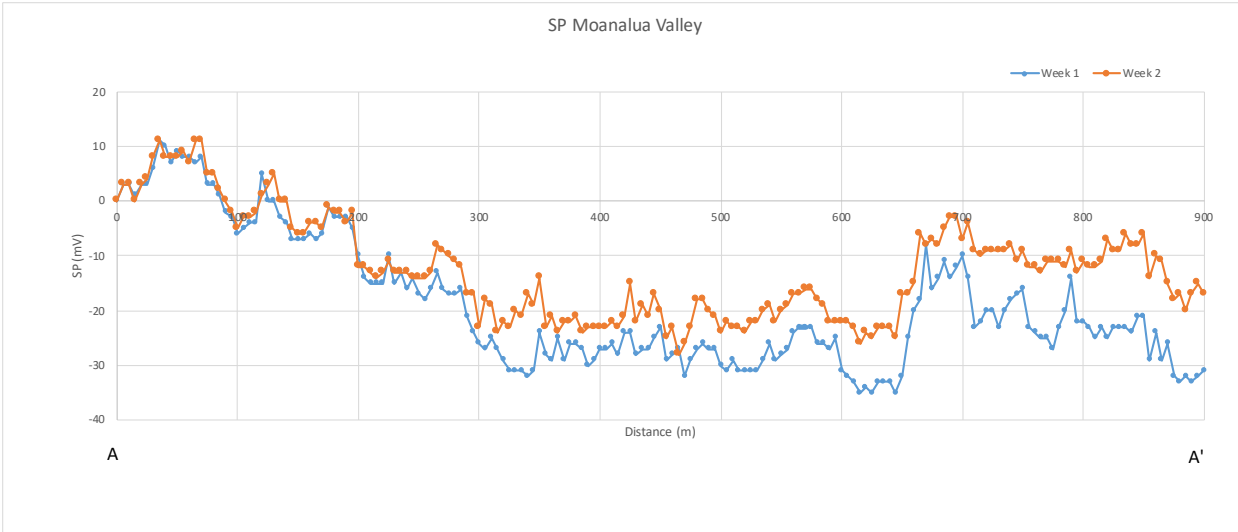


Figure 8-Two weeks of SP data gathered at Moanalua Golf Club from profile A to A' (See Figure 2).

ERT Using ResIpy

The first model created aims to represent the sloping basalt ridge, where it meets the sediment filled valley and saprolite layer, and the freshwater aquifer level. The base of the interface is composed of highly weathered basalt, also known as saprolite clay, which is buried by both weathered terrestrial and marine sediments (Oki, 2005). Figure 9 shows the creation of the subsurface geometry using the triangular mesh.

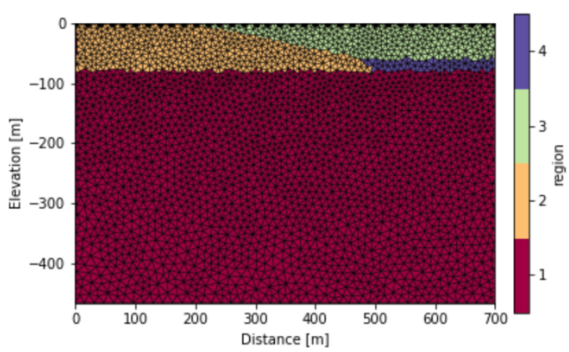


Figure 9-Defined regions using a triangular mesh. Orange is unsaturated basalt, green is valley sediments, purple is saprolite, and red is freshwater-saturated basalt representing the aquifer (see Table 1 for resistivity values).

Figure 10 compares the modeled resistivity values of the stream valley to the inverted resistivity values, where 5% noise has been added to the synthetic data, and an RMS misfit tolerance of 0.1. This model does a very nice job of showing the shape of the resistive unsaturated basalt. However, detail in the sediment and saprolite are lost but the contrast in resistivity is still clearly apparent.

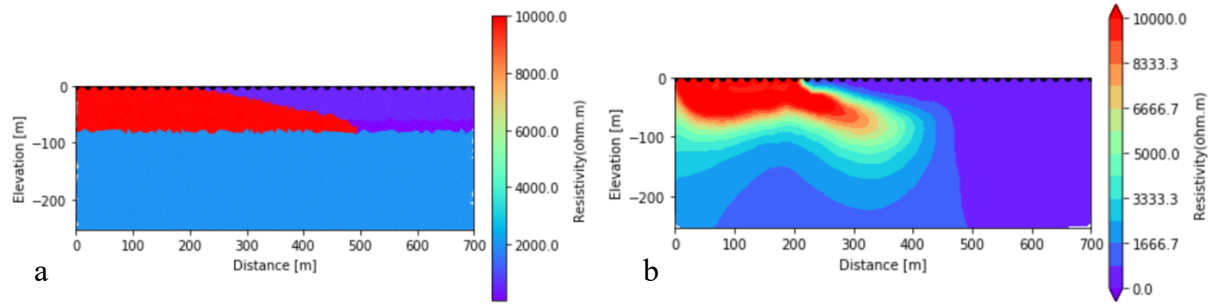


Figure 10-(a) Resistivity values of modeled regions. (b) Inverted model of stream valley.

The stream valley model was reproduced for the first stage of contamination. Figure 11 shows the model and inversion of the two different scenarios that were created. In the first scenario, the contamination was made to be oil saturated basalt and was given a resistivity value of 4,000 Ωm (Figure 11a, b). In the second scenario, the contamination was made to be freshwater saturated basalt and was given a resistivity value of 2,000 Ωm (Figure 10c, d). Let it be known that this resistivity value is the same as that given to the aquifer, which is also freshwater saturated basalt. In this stage the contamination is totally incased within the unsaturated basalt. While the geometry of the contamination is the same, the resistivity value of the contamination is noticeably different.

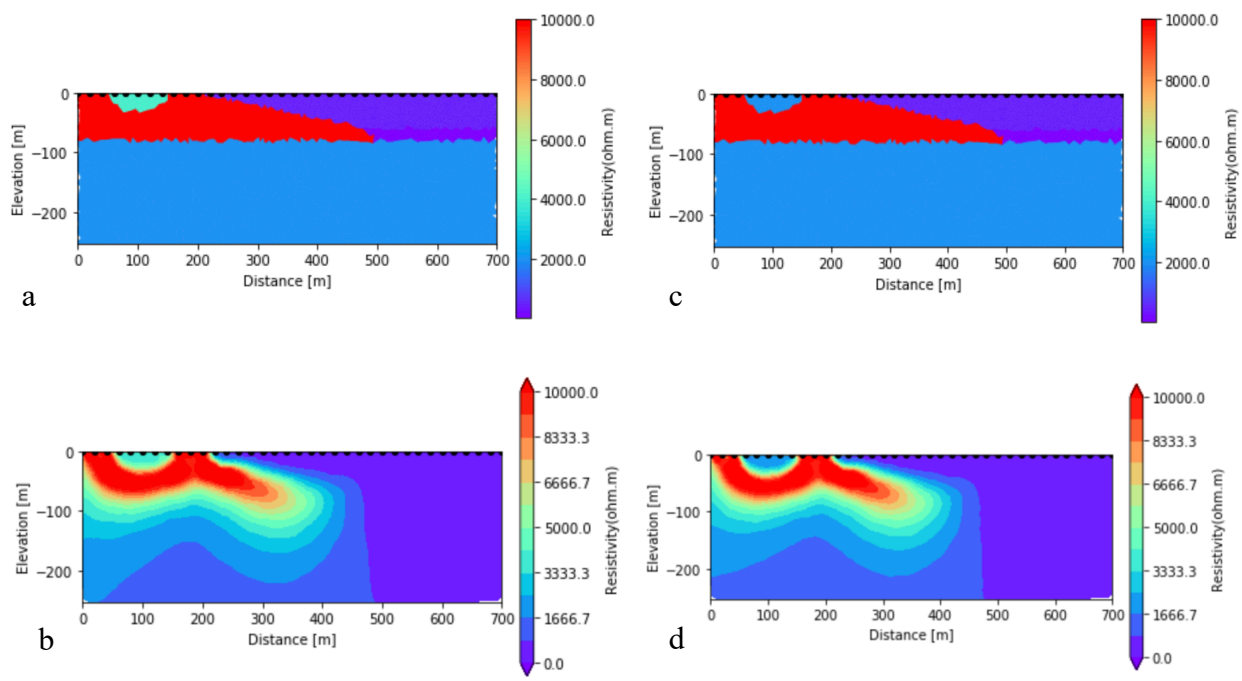


Figure 11-First stage (a) oil contamination model, (b) oil contamination inversion, (c) water infiltration model, and (d) water infiltration inversion.

A second stage of both oil contamination and contrasting water infiltration was created (Figure 12). Again, the contamination remains within the unsaturated basalt however, both the oil and the water are only 10 meters above the freshwater aquifer. At this stage the geometry of the contamination begins to differ, the water infiltration still shows a clearly defined half circle whereas the boundary of the oil contamination is not as well defined. The oil also seems to not

penetrate as deep in the inversion as it truly does in the model. Still, the two different contamination bodies do not have the same resistivity values.

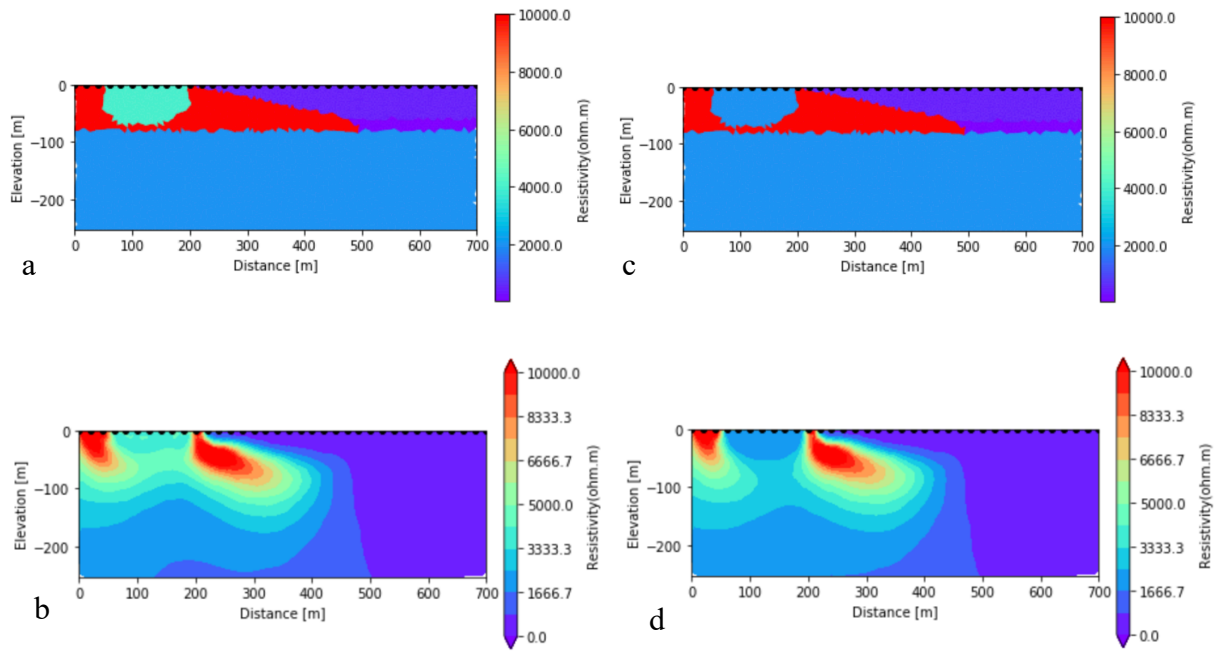
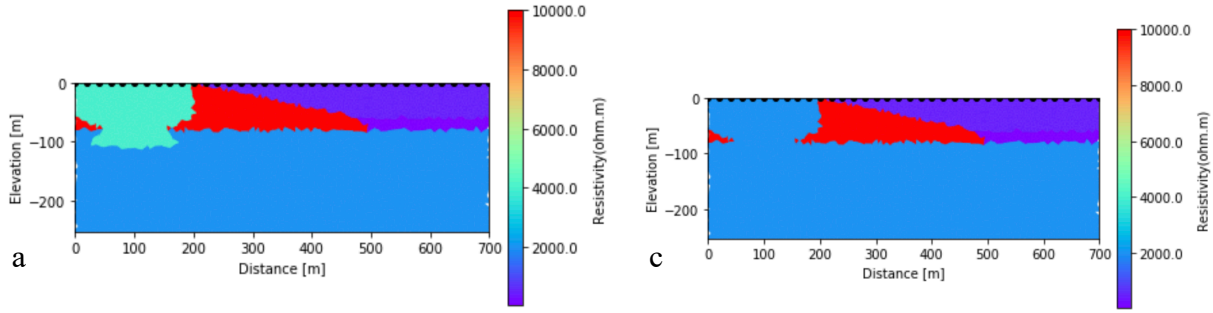


Figure 12-Second stage (a) oil contamination model, (b) oil contamination inversion, (c) water infiltration model, and (d) water infiltration inversion.

Finally, a third and final stage was created, and this time the contamination has reached the freshwater aquifer (Figure 13). In both models the left side of the basalt ridge without any oil or water is no longer visible in the inversion. It appears as though the oil and water have completely saturated the left portion of the model when this is not the case. The geometry is almost completely lost except for the boundary between the contamination and the basalt to the right of it. The contrast in resistivity values between the contamination, unsaturated basalt, and valley sediments are still all visible at the top of the model. Resistivity values between the oil contamination and water infiltration still show up in the model close to their true values. A full Jupyter notebook script of each model is available in the Appendix.



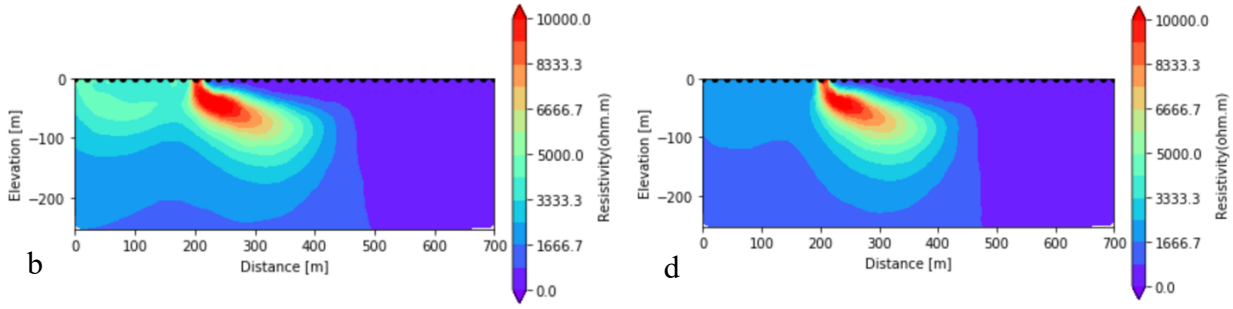


Figure 13-Third stage (a) oil contamination model, (b) oil contamination inversion, (c) water infiltration model, and (d) water infiltration inversion.

5. Hydrological Interpretation Discussion

3D ERT at Hanapēpē

The locations of the subsurface measurements are ultimately determined by the placement of the electrodes. The evenly spaced electrodes across the grid in Configuration 1 give a relatively even spacing in the subsurface measurements. In contrast, the high density of electrodes around the perimeter of Configuration 2, but absence of electrodes in the middle, give shallow subsurface measurements between close electrodes and deeper measurements between electrodes placed far away from each other. Different electrode spacings can be optimized for certain targeted images. If thin, shallow clay layers are the target, then a tightly spaced electrode configuration should be implemented to optimize imaging detail at shallow depths. Alternatively, if the volcanic bedrock is the target, the further the electrodes are spaced (and the larger the total length of the array), the deeper the imaging capabilities. Furthermore, if only one configuration can be deployed then a combination of electrodes spaced both closely and further away will give the best range of imaging resolution.

Self-Potential at Moanalua Golf Club

The similar periodicity between week one and two is due to the fact that the geology and structure of the substrate did not change between these two weeks. The difference in magnitudes between the plot of week one and week two can be interpreted as different amounts of groundwater infiltration and/or flow. According to the National Weather Service (2019), during the week leading up to the first sample date of March 4th, 2020, a total of 2.16 inches of rain fell in Moanalua Valley. NWS station 13003 was used to collect this data, which is only about one kilometer away from the SP profile ("Hawaii Rainfall Summary 90 Day Archive", 2019). In the week leading up to the second sample date of March 11th, 2020, a total of 1.70 inches of rain fell in Moanalua Valley ("Hawaii Rainfall Summary 90 Day Archive", 2019). The relatively lower values of week one can indicate that there was more groundwater infiltration than week two. The rainfall data supports this interpretation. This would also mean that rainfall variations are rapidly reflected by the infiltration rates in the valley. However, it is important to consider that it is hard to draw conclusions from only two sample dates. There could be a number of other factors that either support or dispute this hypothesis, and future measurements to increase our data volume are needed to guide this interpretation.

ERT Using ResIPy

The models I have created can be interpreted in a variety of ways. First off, forward modeling using ResIPy has proven to be successful. The models accurately portray subsurface geology; however, small contrasts between resistivity are not as detailed as they could be. Difficulties have

been seen to arise when trying to distinguish fluid saturated zones from chemical alteration regions like clay (Grobbe and Barde-Cabusson, 2019). Both areas display relatively high electrical conductivity which could account for why the valley sediment, saprolite, and freshwater aquifer are not visible in the inverted models. I would say modeling the difference between oil contamination and freshwater infiltration has also proven to be successful. The models accurately display the difference in resistivity values between the oil and water. Modeling ERT, however, is a lot different than acquiring data through field work. If this situation arose in the field, there would be no side by side comparison to see if a region is truly a contaminant or not. If the resistivity values of oil and water saturated rock are not known, it may be impossible to tell what an inverted image is displaying. Moreover, the context a survey is completed in is highly important. What is expected out of an ERT survey? Is there a lot of rainfall or groundwater recharge going on in the area? Is there a possible source for contamination nearby? My water infiltration models look just as likely to be contamination without the known resistivity value for water saturated basalt, or the context that the Moanalua Valley receives quite a bit of rainfall. This can lead to misidentification in the field which can lead to devastating consequences. On the other hand, a harmless model showing water infiltration could be used to inaccurately illustrate a contamination hazard. It is extremely important to look for relative variations, as well as have knowledge of substrate hydrogeology, resistivity values for saturated and unsaturated materials, and to know the purpose of an ERT campaign. Models can also be used to misconstrued information by colors used in images, values and units used in scales, and even the way the model is built can lead to very different results.

6. Discussion

The models created have demonstrated the strengths and weaknesses of numerical modeling. Modeling parameters can be used to shape the outcomes of inverted models. Using less noise and a lower RMS misfit tolerance can prove to give a very good fit of synthetic data. Opposingly, increasing the level of noise and RMS misfit can blur the results of an inverted model making it harder to distinguish from synthetic data. Whenever numerical modeling is incorporated it is important to remember that the creator of the models have control over the images produced.

If contamination is present near a water source, that contamination may be quite dilute making the resistivity value resemble that of water saturated rock. This is where introducing different methods, like SP, can really make a difference. SP anomalies are dependent on the types of ions dissolved in a fluid. A dilute contaminant will have different ions than uncontaminated freshwater. Methods used together in this sense will help answer questions that individual methods cannot answer on their own.

As more resources are available, more data types should be used when investigating the substrate. Research should be backed by multiple methods when conducting hydrogeologic studies. We have seen how using one method may be misleading, but as soon as another method is introduced, we are guided closer to the truth. This reasoning should be considered when looking at my work and other individuals work to ask the question, did they do enough?

7. Conclusions

Geoelectric methods can be very beneficial in the field of geophysics. Not only can SP and ERT be used together in the field to accurately identify phenomena in the geologic substrate, but models can be made of these methods to experiment with techniques or locations that may not be possible in the field. Uncertainty when using one method can be clarified using another.

My results from the SP campaign agree with the findings of Revil (2002) where gravitational flow due to topography has created negative SP anomalies. Of course, the short duration of this campaign has left many unanswered questions and have made conclusions very hard to draw.

It would be highly beneficial to my research to see the SP profile at Moanalua Valley and ERT acquisition at Hanapēpē be completed in the shortest time possible. A difference in the SP values after only two weeks at Moanalua Valley shows to be very promising in the study of dynamic hydrological systems on O‘ahu. Likewise, understanding the hydrogeology of Hanapēpē, one of the last traditional salt making ponds in the Hawaiian Islands, could not only benefit the geologic knowledge of coastal systems on Kaua‘i and inundation of other coastal areas across the state, but could empower the Hawaiian community to continue and adapt cultural traditions in a rapidly changing environment.

If I could repeat or improve on this research I would model the same scenario in a 3D inversion. This would give insight on how contamination would spread in a 3-dimensional space. More importantly, I would like to conduct an ERT acquisition in Moanalua Valley to compare with the completed SP survey. Having these two methods be completed in the field would mean being able to distinguish signals of geological/structural origin from signals due to groundwater circulation and its variation. The more research that is done with electrical methods in the field could furthermore assist in numerical modeling software to become more precise and accurate.

8. Acknowledgements

I would like to thank my advisors, Niels Grobbe and Stéphanie Barde-Cabusson, for guiding me through geophysical methods both in and out of the field. Your patience with me has allowed me to do research I never thought I was capable of. I would like to thank the Moanalua Golf Club for opening their doors to us and dealing with our many challenges. I would like to thank Guillaume Blanchy and Andrew Binley, the creators of ResIPy, for being extremely timely and helpful when I was stuck on code; I give the same gratitude to the creators of R2 who have made modeling doable, even for an undergraduate like myself.

9. Appendix

Stream Valley Model-No Contamination

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

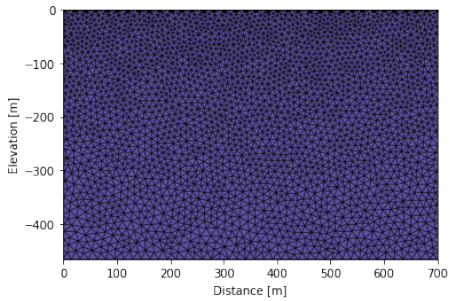
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

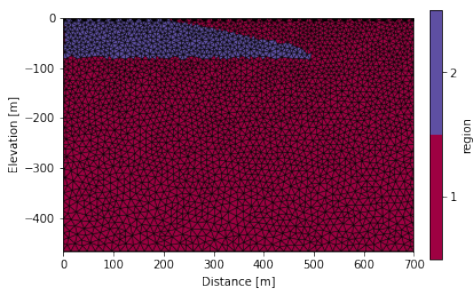
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

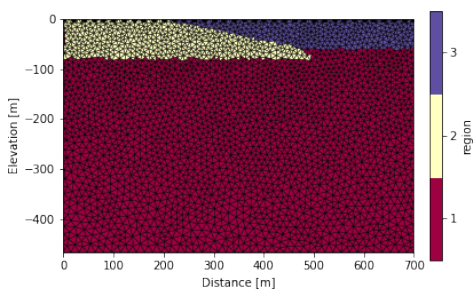
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



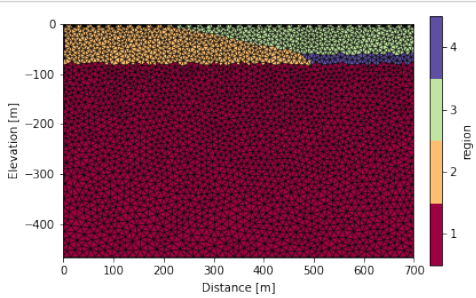
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```

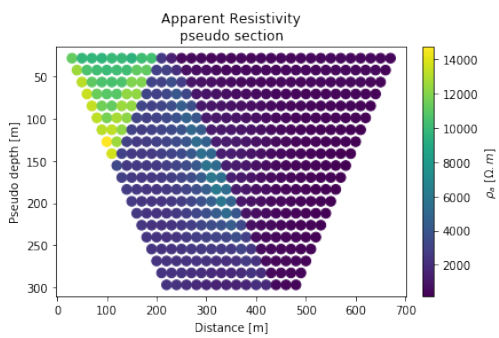


In [8]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [9]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



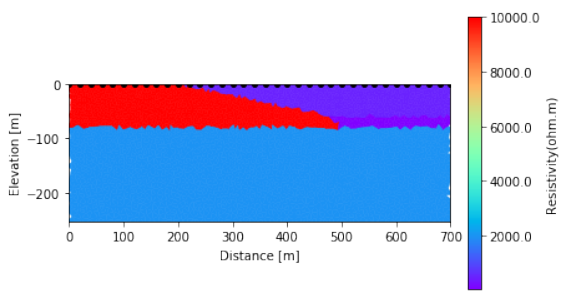
In [10]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.12780E+04

 >> Total Memory required is:          0.346 Gb

   Iteration   1
```

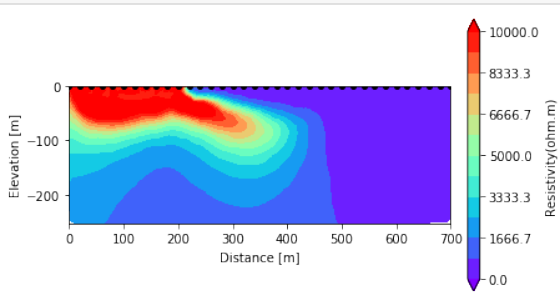
```
Iteration   1
  Initial RMS Misfit:        75.18       Number of data ignored:     0
  Alpha:        1377.764   RMS Misfit:        4.05  Roughness:        7.966
  Alpha:         639.501   RMS Misfit:        3.89  Roughness:        9.330
  Alpha:         296.830   RMS Misfit:        3.86  Roughness:       10.785
  Alpha:         137.776   RMS Misfit:        3.89  Roughness:       12.303
  Step length set to   1.00000
  Final RMS Misfit:        3.86
  Updated data weights

Iteration   2
  Initial RMS Misfit:         2.29       Number of data ignored:     0
  Alpha:         164.321   RMS Misfit:        1.22  Roughness:       12.133
  Alpha:          76.271   RMS Misfit:        0.90  Roughness:       15.179
  Alpha:          35.402   RMS Misfit:        0.70  Roughness:       18.385
  Alpha:          16.432   RMS Misfit:        0.59  Roughness:       21.626
  Alpha:           7.627   RMS Misfit:        0.54  Roughness:       25.087
  Alpha:           3.540   RMS Misfit:        0.53  Roughness:       28.755
  Alpha:           1.643   RMS Misfit:        0.54  Roughness:       32.395
  Step length set to   1.00000
  Final RMS Misfit:        0.53
  Updated data weights

Iteration   3
  Initial RMS Misfit:         0.34       Number of data ignored:     0
  Alpha:           1.551   RMS Misfit:        0.15  Roughness:       34.876
  Alpha:           0.720   RMS Misfit:        0.10  Roughness:       40.149
  Alpha:           0.334   RMS Misfit:        0.08  Roughness:       44.781
  Step length set to   1.00000
  Final RMS Misfit:        0.08
  Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [11]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [12]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 1-Oil Contamination

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

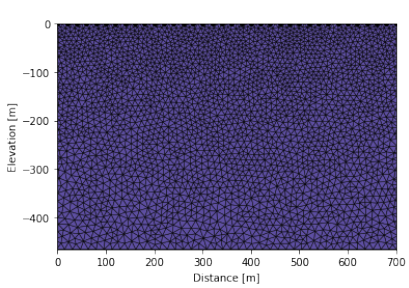
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

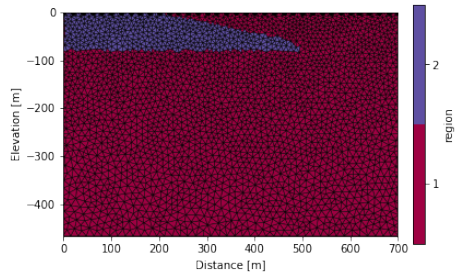
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

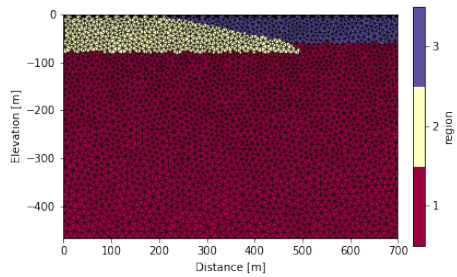
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



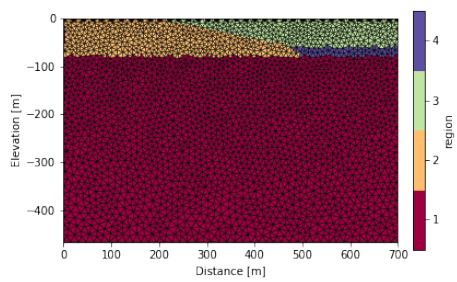
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



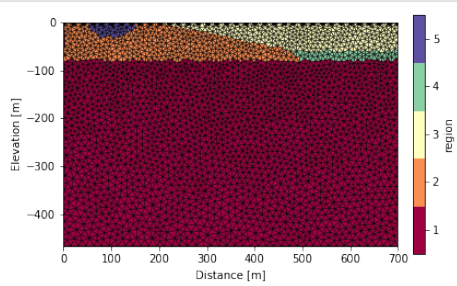
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[50,0],[150,0],[150,-10],[125,-30],[75,-30],[50,-10],[50,0]]), 4000, ipl
ot=True) #oil contamination
```

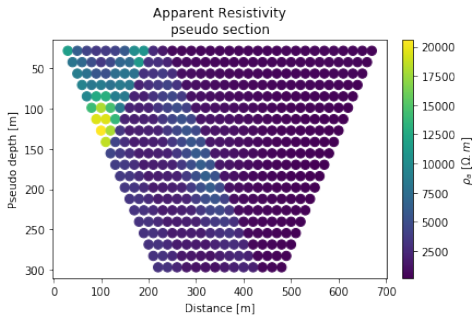


In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



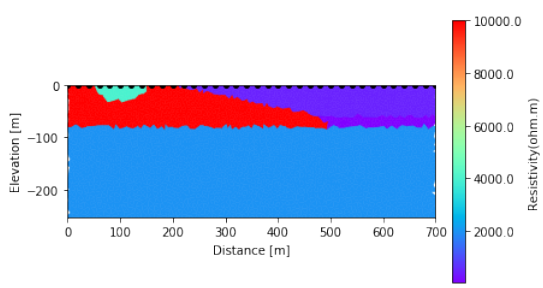
In [11]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION -----------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.12123E+04

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        70.96       Number of data ignored:     0
     Alpha:        1394.520   RMS Misfit:        4.36  Roughness:        8.583
     Alpha:         647.279   RMS Misfit:        4.11  Roughness:       10.247
     Alpha:         300.440   RMS Misfit:        4.04  Roughness:       11.905
     Alpha:         139.452   RMS Misfit:        4.05  Roughness:       13.593
     Step length set to   1.00000
     Final RMS Misfit:        4.04
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         2.49       Number of data ignored:     0
     Alpha:         168.299   RMS Misfit:        1.28  Roughness:       13.959
     Alpha:          78.117   RMS Misfit:        0.94  Roughness:       17.194
     Alpha:          36.259   RMS Misfit:        0.74  Roughness:       20.580
     Alpha:          16.830   RMS Misfit:        0.62  Roughness:       24.149
     Alpha:           7.812   RMS Misfit:        0.57  Roughness:       28.073
```

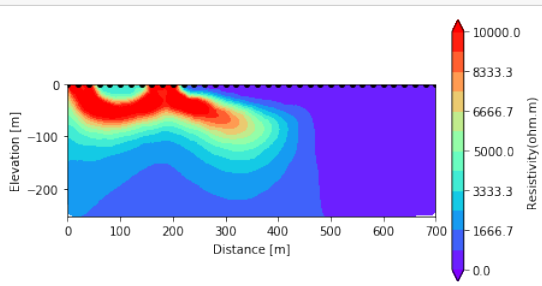
```
>> Total Memory required is:          0.346 Gb

  Iteration   1
    Initial RMS Misfit:        70.96       Number of data ignored:     0
    Alpha:        1394.520   RMS Misfit:        4.36  Roughness:        8.583
    Alpha:         647.279   RMS Misfit:        4.11  Roughness:       10.247
    Alpha:         300.440   RMS Misfit:        4.04  Roughness:       11.905
    Alpha:         139.452   RMS Misfit:        4.05  Roughness:       13.593
    Step length set to   1.00000
    Final RMS Misfit:        4.04
    Updated data weights

  Iteration   2
    Initial RMS Misfit:         2.49       Number of data ignored:     0
    Alpha:         168.299   RMS Misfit:        1.28  Roughness:       13.959
    Alpha:          78.117   RMS Misfit:        0.94  Roughness:       17.194
    Alpha:          36.259   RMS Misfit:        0.74  Roughness:       20.580
    Alpha:          16.830   RMS Misfit:        0.62  Roughness:       24.149
    Alpha:           7.812   RMS Misfit:        0.57  Roughness:       28.073
    Alpha:           3.626   RMS Misfit:        0.55  Roughness:       32.198
    Alpha:           1.683   RMS Misfit:        0.56  Roughness:       36.149
    Step length set to   1.00000
    Final RMS Misfit:        0.55
    Updated data weights

  Iteration   3
    Initial RMS Misfit:         0.36       Number of data ignored:     0
    Alpha:           1.595   RMS Misfit:        0.15  Roughness:       38.724
    Alpha:           0.740   RMS Misfit:        0.10  Roughness:       44.156
    Alpha:           0.344   RMS Misfit:        0.08  Roughness:       48.802
    Step length set to   1.00000
    Final RMS Misfit:        0.08
    Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 1-Water Infiltration

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

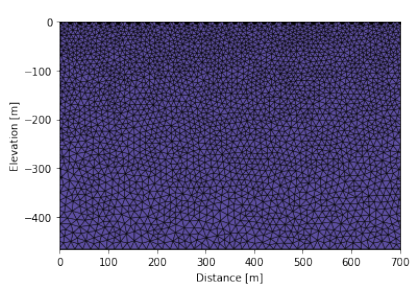
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

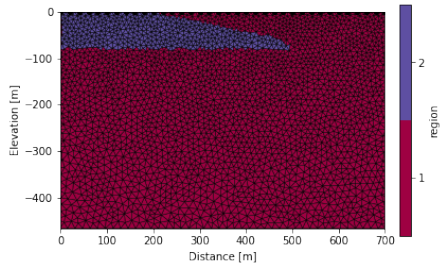
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

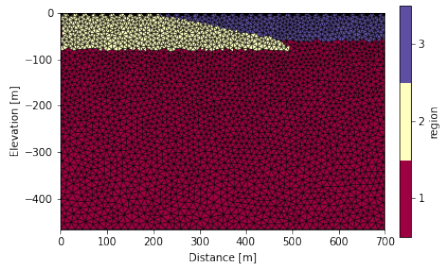
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



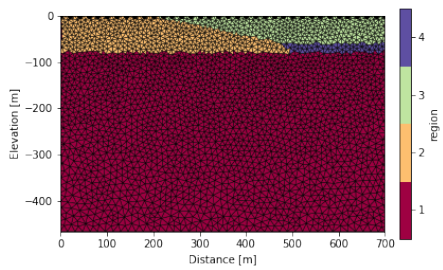
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



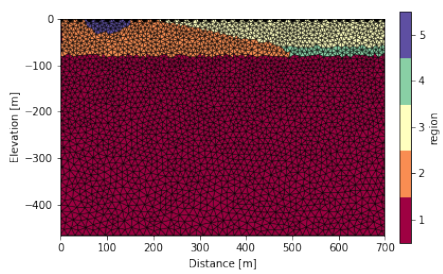
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[50,0],[150,0],[150,-10],[125,-30],[75,-30],[50,-10],[50,0]]), 2000, ipl
ot=True) #water "contamination"
```

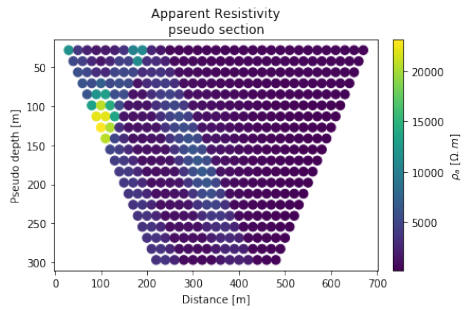


In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



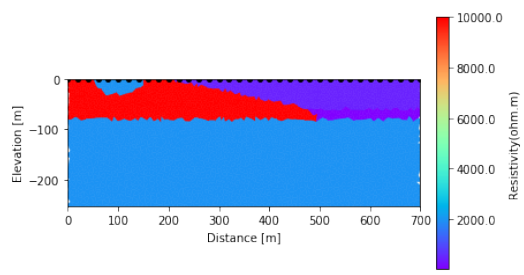
In [11]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.11258E+04

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        65.95       Number of data ignored:     0
     Alpha:        1385.050   RMS Misfit:        5.08  Roughness:       10.056
     Alpha:         642.883   RMS Misfit:        4.83  Roughness:       12.238
     Alpha:         298.400   RMS Misfit:        4.78  Roughness:       14.358
     Alpha:         138.505   RMS Misfit:        4.82  Roughness:       16.510
     Step length set to   1.00000
     Final RMS Misfit:        4.78
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         3.05       Number of data ignored:     0
     Alpha:         171.809   RMS Misfit:        1.40  Roughness:       17.483
     Alpha:          79.747   RMS Misfit:        1.03  Roughness:       21.331
     Alpha:          37.015   RMS Misfit:        0.81  Roughness:       25.105
     Alpha:          17.181   RMS Misfit:        0.68  Roughness:       29.116
     Alpha:           7.975   RMS Misfit:        0.63  Roughness:       33.644
     Alpha:           3.702   RMS Misfit:        0.62  Roughness:       38.343
```

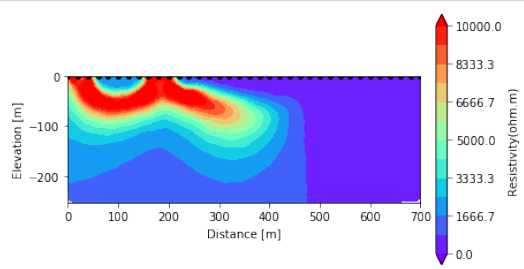
```
>> Total Memory required is:          0.346 Gb

  Iteration   1
    Initial RMS Misfit:        65.95       Number of data ignored:     0
    Alpha:        1385.050   RMS Misfit:        5.08  Roughness:       10.056
    Alpha:         642.883   RMS Misfit:        4.83  Roughness:       12.238
    Alpha:         298.400   RMS Misfit:        4.78  Roughness:       14.358
    Alpha:         138.505   RMS Misfit:        4.82  Roughness:       16.510
    Step length set to   1.00000
    Final RMS Misfit:        4.78
    Updated data weights

  Iteration   2
    Initial RMS Misfit:         3.05       Number of data ignored:     0
    Alpha:         171.809   RMS Misfit:        1.40  Roughness:       17.483
    Alpha:          79.747   RMS Misfit:        1.03  Roughness:       21.331
    Alpha:          37.015   RMS Misfit:        0.81  Roughness:       25.105
    Alpha:          17.181   RMS Misfit:        0.68  Roughness:       29.116
    Alpha:           7.975   RMS Misfit:        0.63  Roughness:       33.644
    Alpha:           3.702   RMS Misfit:        0.62  Roughness:       38.343
    Alpha:           1.718   RMS Misfit:        0.63  Roughness:       42.771
    Step length set to   1.00000
    Final RMS Misfit:        0.62
    Updated data weights

  Iteration   3
    Initial RMS Misfit:         0.40       Number of data ignored:     0
    Alpha:           1.646   RMS Misfit:        0.16  Roughness:       45.668
    Alpha:           0.764   RMS Misfit:        0.11  Roughness:       51.465
    Alpha:           0.355   RMS Misfit:        0.09  Roughness:       56.275
    Step length set to   1.00000
    Final RMS Misfit:        0.09
    Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 2-Oil Contamination

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

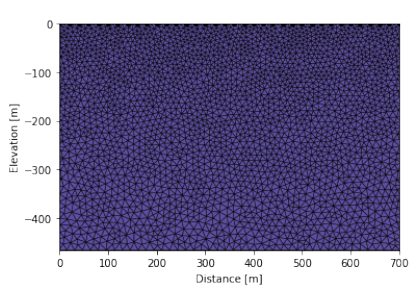
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

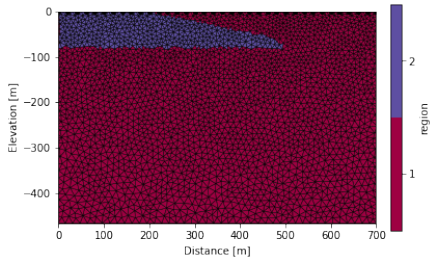
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

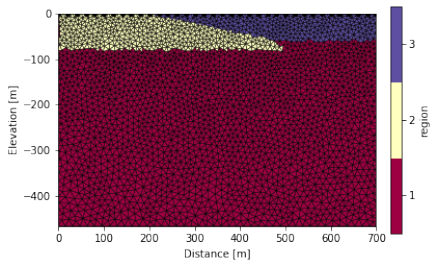
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



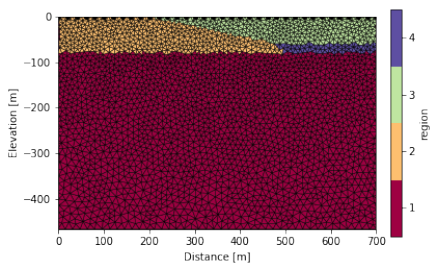
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



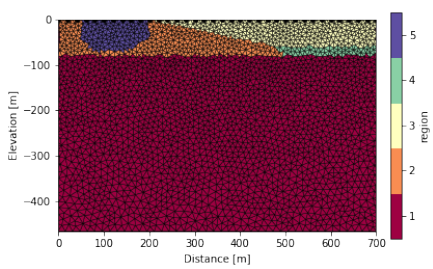
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[50,0],[200,0],[200,-40],[150,-70],[75,-70],[50,-40],[50,0]]), 4000, ipl
ot=True) #oil contamination
```



In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

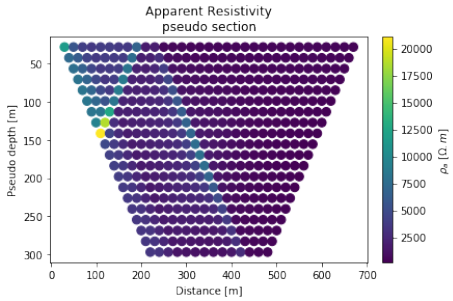
```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
```

```
In [10]: k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



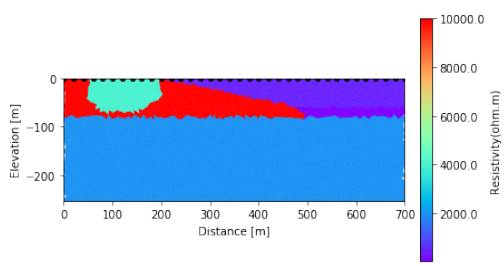

```
In [11]: k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.11444E+04

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        62.09       Number of data ignored:     0
     Alpha:        1171.266   RMS Misfit:        3.84  Roughness:        7.704
     Alpha:         543.654   RMS Misfit:        3.55  Roughness:        9.594
     Alpha:         252.342   RMS Misfit:        3.46  Roughness:       11.532
     Alpha:         117.127   RMS Misfit:        3.46  Roughness:       13.385
     Alpha:          54.365   RMS Misfit:        3.49  Roughness:       15.177
     Step length set to   1.00000
     Final RMS Misfit:        3.46
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         1.94       Number of data ignored:     0
     Alpha:          63.784   RMS Misfit:        0.84  Roughness:       15.792
     Alpha:          29.606   RMS Misfit:        0.61  Roughness:       19.581
     Alpha:          13.742   RMS Misfit:        0.48  Roughness:       23.254
     Alpha:           6.378   RMS Misfit:        0.41  Roughness:       26.965
     Alpha:           2.961   RMS Misfit:        0.38  Roughness:       30.773
     Alpha:           1.374   RMS Misfit:        0.38  Roughness:       34.482
     Step length set to   1.00000
     Final RMS Misfit:        0.38
     Updated data weights
```

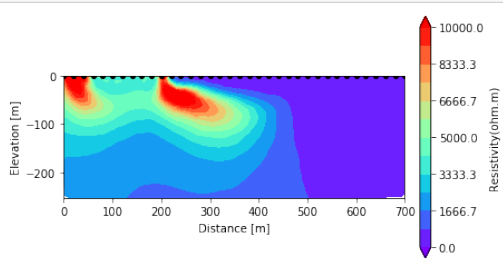
```
Processing dataset   1


Measurements read:   470     Measurements rejected:     0
  Geometric mean of apparent resistivities:  0.11444E+04

>> Total Memory required is:          0.346 Gb

  Iteration   1
    Initial RMS Misfit:        62.09       Number of data ignored:     0
    Alpha:        1171.266   RMS Misfit:        3.84  Roughness:        7.704
    Alpha:         543.654   RMS Misfit:        3.55  Roughness:        9.594
    Alpha:         252.342   RMS Misfit:        3.46  Roughness:       11.532
    Alpha:         117.127   RMS Misfit:        3.46  Roughness:       13.385
    Alpha:          54.365   RMS Misfit:        3.49  Roughness:       15.177
    Step length set to   1.00000
    Final RMS Misfit:        3.46
    Updated data weights

  Iteration   2
    Initial RMS Misfit:         1.94       Number of data ignored:     0
    Alpha:          63.784   RMS Misfit:        0.84  Roughness:       15.792
    Alpha:          29.606   RMS Misfit:        0.61  Roughness:       19.581
    Alpha:          13.742   RMS Misfit:        0.48  Roughness:       23.254
    Alpha:           6.378   RMS Misfit:        0.41  Roughness:       26.965
    Alpha:           2.961   RMS Misfit:        0.38  Roughness:       30.773
    Alpha:           1.374   RMS Misfit:        0.38  Roughness:       34.482
    Step length set to   1.00000
    Final RMS Misfit:        0.38
    Updated data weights

  Iteration   3
    Initial RMS Misfit:         0.25       Number of data ignored:     0
    Alpha:           1.240   RMS Misfit:        0.13  Roughness:       36.021
    Alpha:           0.576   RMS Misfit:        0.09  Roughness:       41.318
    Step length set to   1.00000
    Final RMS Misfit:        0.09
    Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 2-Water Infiltration

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

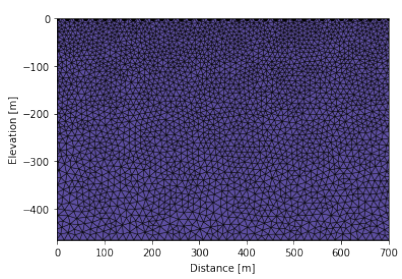
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

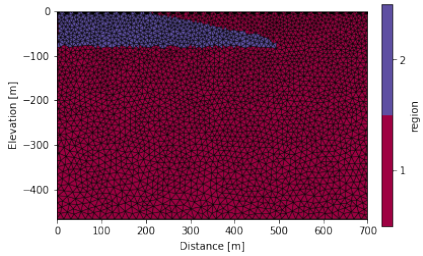
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

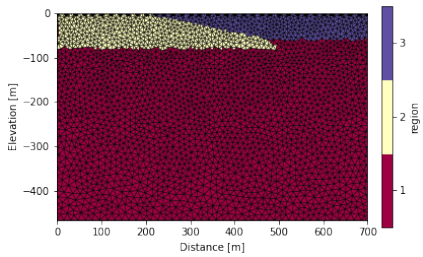
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



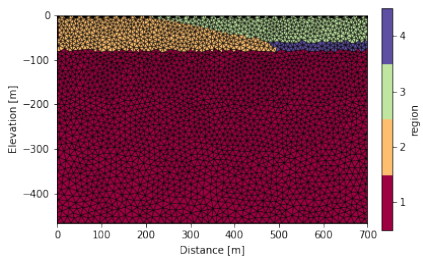
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



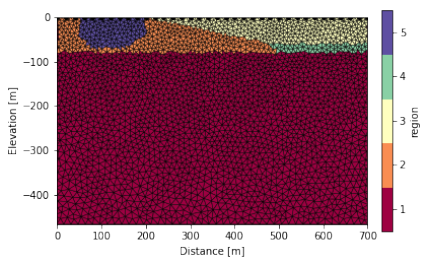
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[50,0],[200,0],[200,-40],[150,-70],[75,-70],[50,-40],[50,0]]), 2000, ipl
ot=True) #water "contamination"
```



In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

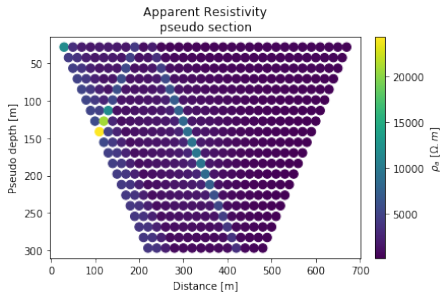
```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



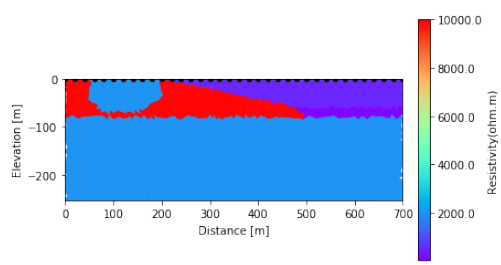
In [11]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.99962E+03

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        52.09       Number of data ignored:     0
     Alpha:        1004.803   RMS Misfit:        4.24  Roughness:        9.286
     Alpha:         466.388   RMS Misfit:        3.92  Roughness:       11.972
     Alpha:         216.478   RMS Misfit:        3.82  Roughness:       14.772
     Alpha:         100.480   RMS Misfit:        3.82  Roughness:       17.525
     Step length set to   1.00000
     Final RMS Misfit:        3.82
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         2.11       Number of data ignored:     0
     Alpha:         119.641   RMS Misfit:        1.23  Roughness:       14.812
     Alpha:          55.532   RMS Misfit:        0.85  Roughness:       19.427
     Alpha:          25.776   RMS Misfit:        0.62  Roughness:       23.864
     Alpha:          11.964   RMS Misfit:        0.50  Roughness:       27.871
     Alpha:           5.553   RMS Misfit:        0.44  Roughness:       31.720
     Alpha:           2.578   RMS Misfit:        0.43  Roughness:       35.600
     Alpha:           1.196   RMS Misfit:        0.45  Roughness:       39.429
     Step length set to   1.00000
     Final RMS Misfit:        0.43
     Updated data weights
```

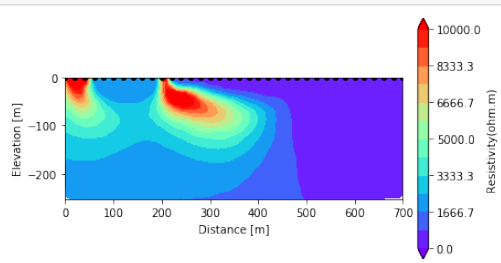
```
Processing dataset   1


Measurements read:   470     Measurements rejected:     0
  Geometric mean of apparent resistivities:  0.99962E+03

>> Total Memory required is:          0.346 Gb

  Iteration   1
    Initial RMS Misfit:        52.09       Number of data ignored:     0
    Alpha:        1004.803   RMS Misfit:        4.24  Roughness:        9.286
    Alpha:         466.388   RMS Misfit:        3.92  Roughness:       11.972
    Alpha:         216.478   RMS Misfit:        3.82  Roughness:       14.772
    Alpha:         100.480   RMS Misfit:        3.82  Roughness:       17.525
    Step length set to   1.00000
    Final RMS Misfit:        3.82
    Updated data weights

  Iteration   2
    Initial RMS Misfit:         2.11       Number of data ignored:     0
    Alpha:         119.641   RMS Misfit:        1.23  Roughness:       14.812
    Alpha:          55.532   RMS Misfit:        0.85  Roughness:       19.427
    Alpha:          25.776   RMS Misfit:        0.62  Roughness:       23.864
    Alpha:          11.964   RMS Misfit:        0.50  Roughness:       27.871
    Alpha:           5.553   RMS Misfit:        0.44  Roughness:       31.720
    Alpha:           2.578   RMS Misfit:        0.43  Roughness:       35.600
    Alpha:           1.196   RMS Misfit:        0.45  Roughness:       39.429
    Step length set to   1.00000
    Final RMS Misfit:        0.43
    Updated data weights

  Iteration   3
    Initial RMS Misfit:         0.28       Number of data ignored:     0
    Alpha:           1.086   RMS Misfit:        0.13  Roughness:       41.586
    Alpha:           0.504   RMS Misfit:        0.09  Roughness:       47.117
    Step length set to   1.00000
    Final RMS Misfit:        0.09
    Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 3-Oil Contamination

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

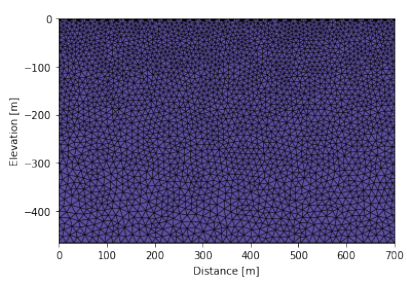
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

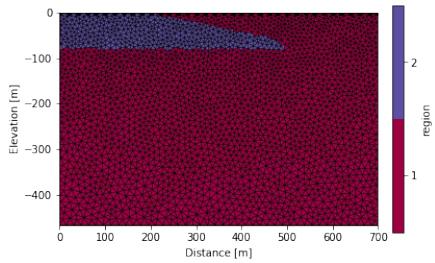
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

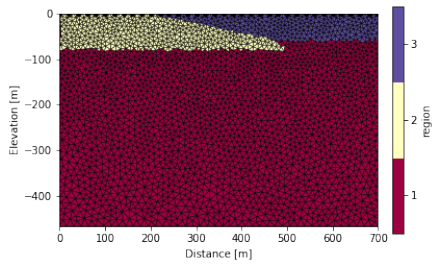
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



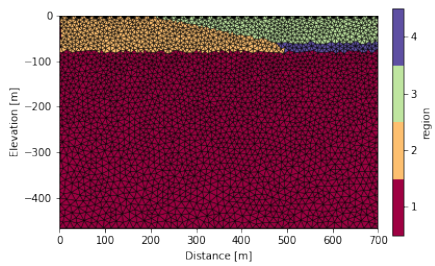
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



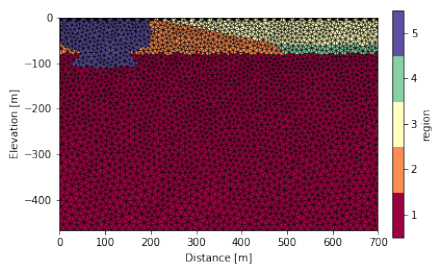
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[0,0],[200,0],[200,-60],[150,-80],[175,-100],[150,-110],[50,-110],[25,-1
00],[50,-80],[0,-60],[0,0]]), 4000, iplot=True) #oil contamination
```



In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

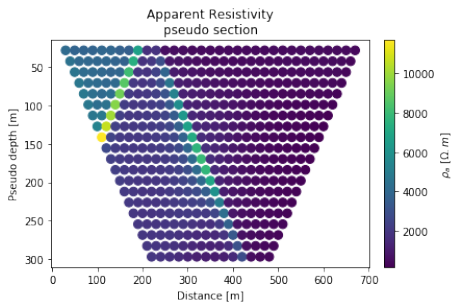
```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



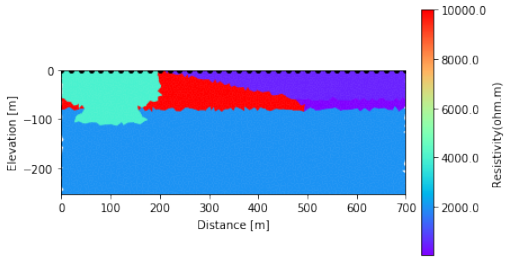
In [11]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
-------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.10982E+04

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        58.32       Number of data ignored:     0
     Alpha:        1130.675   RMS Misfit:        3.66  Roughness:        6.515
     Alpha:         524.813   RMS Misfit:        3.41  Roughness:        8.292
     Alpha:         243.597   RMS Misfit:        3.33  Roughness:       10.167
     Alpha:         113.067   RMS Misfit:        3.34  Roughness:       11.978
     Step length set to   1.00000
     Final RMS Misfit:        3.33
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         1.80       Number of data ignored:     0
     Alpha:         131.539   RMS Misfit:        1.14  Roughness:       10.542
     Alpha:          61.055   RMS Misfit:        0.81  Roughness:       14.022
     Alpha:          28.339   RMS Misfit:        0.60  Roughness:       17.595
     Alpha:          13.154   RMS Misfit:        0.49  Roughness:       21.044
     Alpha:           6.106   RMS Misfit:        0.44  Roughness:       24.524
     Alpha:           2.834   RMS Misfit:        0.42  Roughness:       28.016
     Alpha:           1.315   RMS Misfit:        0.43  Roughness:       31.326
     Step length set to   1.00000
     Final RMS Misfit:        0.42
     Updated data weights
```

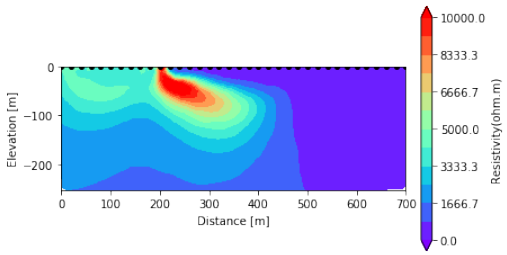
```
Processing dataset   1


Measurements read:   470     Measurements rejected:     0
  Geometric mean of apparent resistivities:  0.10982E+04

>> Total Memory required is:          0.346 Gb

  Iteration   1
    Initial RMS Misfit:        58.32       Number of data ignored:     0
    Alpha:        1130.675   RMS Misfit:        3.66  Roughness:        6.515
    Alpha:         524.813   RMS Misfit:        3.41  Roughness:        8.292
    Alpha:         243.597   RMS Misfit:        3.33  Roughness:       10.167
    Alpha:         113.067   RMS Misfit:        3.34  Roughness:       11.978
    Step length set to   1.00000
    Final RMS Misfit:        3.33
    Updated data weights

  Iteration   2
    Initial RMS Misfit:         1.80       Number of data ignored:     0
    Alpha:         131.539   RMS Misfit:        1.14  Roughness:       10.542
    Alpha:          61.055   RMS Misfit:        0.81  Roughness:       14.022
    Alpha:          28.339   RMS Misfit:        0.60  Roughness:       17.595
    Alpha:          13.154   RMS Misfit:        0.49  Roughness:       21.044
    Alpha:           6.106   RMS Misfit:        0.44  Roughness:       24.524
    Alpha:           2.834   RMS Misfit:        0.42  Roughness:       28.016
    Alpha:           1.315   RMS Misfit:        0.43  Roughness:       31.326
    Step length set to   1.00000
    Final RMS Misfit:        0.42
    Updated data weights

  Iteration   3
    Initial RMS Misfit:         0.27       Number of data ignored:     0
    Alpha:           1.196   RMS Misfit:        0.13  Roughness:       33.528
    Alpha:           0.555   RMS Misfit:        0.09  Roughness:       38.556
    Step length set to   1.00000
    Final RMS Misfit:        0.09
    Final RMS Misfit:        0.10

Solution converged - Outputing results to file

Calculating sensitivity map


Processing dataset   2


End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



Stage 3-Water Infiltration

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
import os
import sys
sys.path.append((os.path.relpath('./src')))
import numpy as np
from resipy import R2
```

```
API path =  C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\resipy-2.1
.0\resipy
ResIPy version =  2.1.0
```

In [2]:
```
k = R2()
```

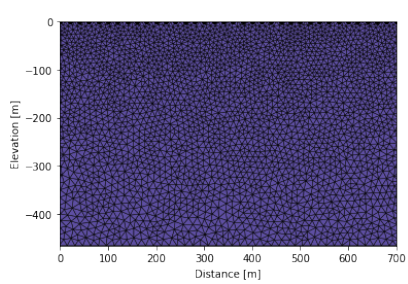
```
Working directory is: C:\Users\Geophysics\Documents\ResIPy\resipy-2.1.0-20200504T223131Z-001\
resipy-2.1.0\resipy\invdir
clearing the dirname
```

In [3]:
```
nx=36;
dx=20;
elec = np.zeros((nx,3))
elec[:,0] = np.arange(0,nx*dx,dx)
k.setElec(elec)
print(k.elec)
```

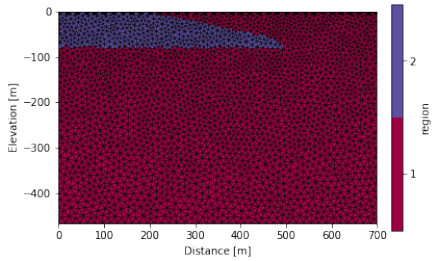
```
[[  0.   0.   0.]
 [ 20.   0.   0.]
 [ 40.   0.   0.]
 [ 60.   0.   0.]
 [ 80.   0.   0.]
 [100.   0.   0.]
 [120.   0.   0.]
 [140.   0.   0.]
 [160.   0.   0.]
 [180.   0.   0.]
 [200.   0.   0.]
 [220.   0.   0.]
 [240.   0.   0.]
 [260.   0.   0.]
 [280.   0.   0.]
 [300.   0.   0.]
 [320.   0.   0.]
 [340.   0.   0.]
 [360.   0.   0.]
 [380.   0.   0.]
 [400.   0.   0.]
 [420.   0.   0.]
 [440.   0.   0.]
 [460.   0.   0.]
 [480.   0.   0.]
 [500.   0.   0.]
 [520.   0.   0.]
 [540.   0.   0.]
 [560.   0.   0.]
 [580.   0.   0.]
 [600.   0.   0.]
 [620.   0.   0.]
 [640.   0.   0.]
 [660.   0.   0.]
 [680.   0.   0.]
 [700.   0.   0.]]
```

In [4]:
```
k.createMesh(typ='trian', show_output=False, res0=2000) #let background resistivity be freshwa
ter saturated basalt
k.showMesh()
```

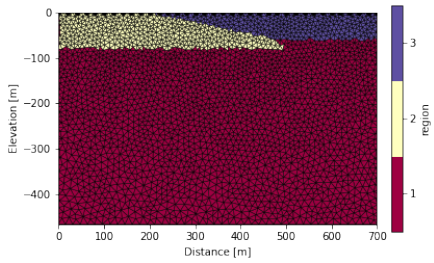
```
Creating triangular mesh...Reading mesh.msh
Gmsh version == 3.x
reading node coordinates...
Determining element type...Triangle
Reading connection matrix...
ignoring 0 elements in the mesh file, as they are not required for R2/R3t
Finished reading .msh file
done
```



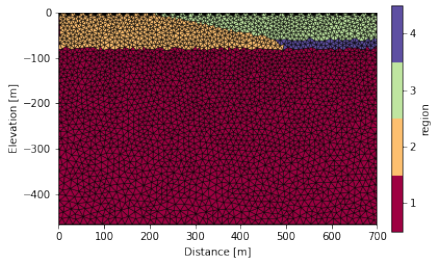
In [5]:
```
k.addRegion(np.array([[0,0],[200,0],[475,-60],[500,-80],[0,-80],[0,0]]), 10000, iplot=True) #d
ry basalt
```



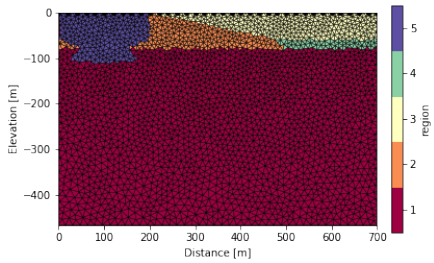
In [6]:
```
k.addRegion(np.array([[200,0],[700,0],[700,-60],[475,-60],[200,0]]), 500, iplot=True) #stream
valley fill
```



In [7]:
```
k.addRegion(np.array([[475,-60],[700,-60],[700,-80],[500,-80],[475,-60]]), 50, iplot=True) #sa
prolite
```



In [8]:
```
k.addRegion(np.array([[0,0],[200,0],[200,-60],[150,-80],[175,-100],[150,-110],[50,-110],[25,-1
00],[50,-80],[0,-60],[0,0]]), 2000, iplot=True) #water "contamination"
```



In [9]:
```
k.createSequence([('dpdp1', 1, 20)]) # create a dipole-dipole of diple spacing of 1 (=skip 0)
with 20 levels
print(k.sequence) # the sequence is stored inside the R2 object
```

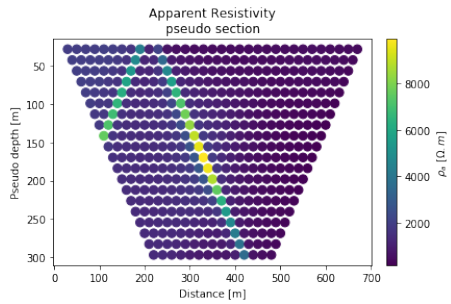
```
470 quadrupoles generated.
[[ 1  2  3  4]
 [ 2  3  4  5]
 [ 3  4  5  6]
 ...
 [12 13 33 34]
 [13 14 34 35]
 [14 15 35 36]]
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
```

In [10]:
```
k.forward(noise=0.05, iplot=True)
```

```
Writing .in file and mesh.dat... done!
Writing protocol.dat... done!
Running forward model...

 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> F o r w a r d   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from resistivity.dat

 Measurements read:   470     Measurements rejected:     0

 >> Total Memory required is:          0.313 Gb
470/470 reciprocal measurements NOT found.
0 measurements error > 20 %
Forward modelling done.
```



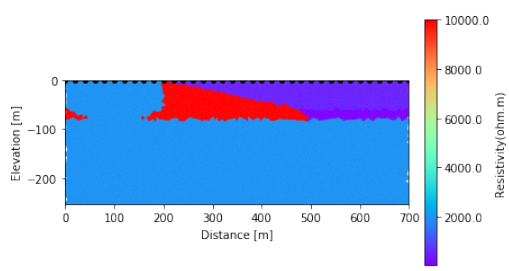
In [11]:
```
k.invert(param={'tolerance':0.1})
```

```
Writing .in file and protocol.dat... All non fixed parameters reset to 100 Ohm.m and 0 mrad,
as the survey to be inverted is from a forward model.
done!
--------------------- MAIN INVERSION ------------------


 >> R  2    R e s i s t i v i t y   I n v e r s i o n   v4.0 <<

 >> D a t e : 09 - 05 - 2020
 >> My beautiful survey
 >> I n v e r s e   S o l u t i o n   S e l e c t e d <<
 >> Determining storage needed for finite element conductance matrix
 >> Generating index array for finite element conductance matrix
 >> Reading start resistivity from res0.dat
 >> R e g u l a r i s e d   T y p e <<
 >>   L i n e a r   F i l t e r   <<
 >> L o g - D a t a   I n v e r s i o n <<
 >> N o r m a l   R e g u l a r i s a t i o n <<
 >> D a t a   w e i g h t s   w i l l   b e   m o d i f i e d <<
 ** WARNING: it is recommended that the tolerance is set to 1


 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.91672E+03

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        46.75       Number of data ignored:     0
     Alpha:         913.515   RMS Misfit:        3.78  Roughness:        7.090
     Alpha:         424.016   RMS Misfit:        3.54  Roughness:        9.381
     Alpha:         196.811   RMS Misfit:        3.47  Roughness:       11.720
     Alpha:          91.352   RMS Misfit:        3.49  Roughness:       13.918
     Step length set to   1.00000
     Final RMS Misfit:        3.47
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         1.79       Number of data ignored:     0
     Alpha:         106.624   RMS Misfit:        1.12  Roughness:       11.375
     Alpha:          49.490   RMS Misfit:        0.77  Roughness:       15.575
     Alpha:          22.971   RMS Misfit:        0.56  Roughness:       19.748
     Alpha:          10.662   RMS Misfit:        0.45  Roughness:       23.590
     Alpha:           4.949   RMS Misfit:        0.40  Roughness:       27.234
     Alpha:           2.297   RMS Misfit:        0.38  Roughness:       30.718
     Alpha:           1.066   RMS Misfit:        0.40  Roughness:       33.947
     Step length set to   1.00000
     Final RMS Misfit:        0.38
     Updated data weights
```

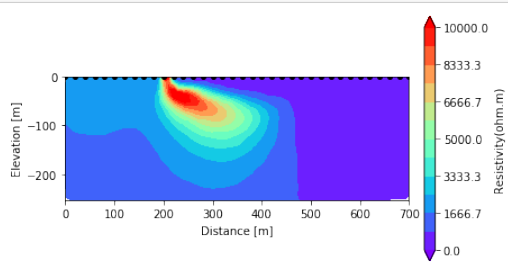
```
 Processing dataset   1


 Measurements read:   470     Measurements rejected:     0
   Geometric mean of apparent resistivities:  0.91672E+03

 >> Total Memory required is:          0.346 Gb

   Iteration   1
     Initial RMS Misfit:        46.75       Number of data ignored:     0
     Alpha:         913.515   RMS Misfit:        3.78  Roughness:        7.090
     Alpha:         424.016   RMS Misfit:        3.54  Roughness:        9.381
     Alpha:         196.811   RMS Misfit:        3.47  Roughness:       11.720
     Alpha:          91.352   RMS Misfit:        3.49  Roughness:       13.918
     Step length set to   1.00000
     Final RMS Misfit:        3.47
     Updated data weights

   Iteration   2
     Initial RMS Misfit:         1.79       Number of data ignored:     0
     Alpha:         106.624   RMS Misfit:        1.12  Roughness:       11.375
     Alpha:          49.490   RMS Misfit:        0.77  Roughness:       15.575
     Alpha:          22.971   RMS Misfit:        0.56  Roughness:       19.748
     Alpha:          10.662   RMS Misfit:        0.45  Roughness:       23.590
     Alpha:           4.949   RMS Misfit:        0.40  Roughness:       27.234
     Alpha:           2.297   RMS Misfit:        0.38  Roughness:       30.718
     Alpha:           1.066   RMS Misfit:        0.40  Roughness:       33.947
     Step length set to   1.00000
     Final RMS Misfit:        0.38
     Updated data weights

   Iteration   3
     Initial RMS Misfit:         0.24       Number of data ignored:     0
     Alpha:           0.944   RMS Misfit:        0.11  Roughness:       36.265
     Alpha:           0.438   RMS Misfit:        0.08  Roughness:       41.254
     Step length set to   1.00000
     Final RMS Misfit:        0.08
     Final RMS Misfit:        0.10

 Solution converged - Outputing results to file

 Calculating sensitivity map


 Processing dataset   2


 End of data:  Terminating
1/1 results parsed (1 ok; 0 failed)
```

In [12]:
```
k.showResults(index=0, attr='Resistivity(ohm.m)', color_map='rainbow') #show the initial model
```

```
ERROR: No sensitivity attribute found
```



In [13]:
```
k.showResults(index=1, attr='Resistivity(ohm.m)', vmin=0, vmax=10000, sens=False, color_map='r
ainbow', contour=True)
```



7. References

Barde-Cabusson, Stéphanie. Formation de caldera par fluage d'un syst`eme hydrothermal volcanique. Volcanologie. Université Blaise Pascal - Clermont-Ferrand II, 2007. Fran ̧cais. <NNT: 2007CLF21752>. <tel-00718055>

Barde-Cabusson, S et al. "Structural Control of Collapse Events Inferred by Self-Potential Mapping on the Piton de La Fournaise Volcano (La Réunion Island)." Journal of Volcanology and Geothermal Research 209-210.C (2012): 9–18. Web.

Blanchy G., Saneiyan S., Boyd J., McLachlan P. and Binley A. 2020. "ResIPy, an Intuitive Open Source Software for Complex Geoelectrical Inversion/Modeling." Computers & Geosciences, February, 104423. https://doi.org/10.1016/j.cageo.2020.104423.

Corwin, R.F., 1997. The self-potential method for environmental and engineering applications: geotechnical and environmental geophysics. In: Ward, H. (Ed.), Investigations in Geophysics: Soc. Expl. Geophys, vol. 5, p. 1.

Daily, William et al. "Electrical Resistance Tomography." The Leading Edge 23.5 (2004): 438–442. Web.

Giampaolo, V. et al. "Self-Potential Monitoring of a Crude Oil-Contaminated Site (Trecate, Italy)." Environmental Science and Pollution Research 21.15 (2014): 8932–8947. Web.

"Google Earth." *Google Earth*, Google, 2020, www.google.com/earth/.

Grobbe, N. and Barde-Cabusson, S., "Self-Potential Studies in Volcanic Environments: A Cheap and Efficient Method for Multiscale Fluid-Flow Investigations." International Journal of Geophysics 2019 (2019): 1–19. Web.

Kane, Haunani et al. "Modeling Sea-Level Rise Vulnerability of Coastal Environments Using Ranked Management Concerns." Climatic Change 131.2 (2015): 349–361. Web.

Lowrie, William. *Fundamentals of Geophysics*. 2nd ed., Cambridge University Press, 2007.

Oki, Delwyn S. Numerical Simulation of the Effects of Low-Permeability Valley-Fill Barriers and the Redistribution of Ground-Water Withdrawals in the Pearl Harbor Area, Oahu, Hawaii . Reston, Va: U.S. Geological Survey, 2005. Print.

Pap, R., Lerner, D., and Nobrega-Olivera, G. "Hydrological Investigation at Salt Pond Work Plan." Funded Proposal, March 2019.

Rao, Gtamma et al. "Application of Electrical Resistivity Tomography Methods for Delineation of Groundwater Contamination and Potential Zones." Arabian Journal of Geosciences 7.4 (2014): 1373–1384. Web.

Revil, A. “Comment on ‘Rapid Fluid Disruption: A Source for Self-potential Anomalies on Volcanoes’ by M. J. S. Johnston, J. D. Byerlee, and D. Lockner.” Journal of Geophysical Research: Solid Earth 107.B8 (2002): ETG 2–1–ETG 2–5. Web.

Revil, André, and Abderrahim. Jardani. The Self-Potential Method Theory and Applications in Environmental Geosciences . Cambridge ;: Cambridge University Press, 2013. Print.

Sill, W.R., 1983. Self potential modeling from primary flows. Geophysics 48, 76–86.

Thiagarajan, S. et al. “Delineation of Groundwater Resources Using Electrical Resistivity Tomography.” Arabian Journal of Geosciences 11.9 (2018): 1–16. Web.

US Department of Commerce, and Noaa. “Hawaii Rainfall Summary 90 Day Archive.” *National Weather Service*, NOAA's National Weather Service, 6 Feb. 2019, www.weather.gov/hfo/HRS_archive.