**A New Method for Representing Inclinometer Data for Slope Stability**

Tobias Cumins1\*, P. Renforth1 & N. Farnham1

*1Earth and Ocean Sciences, Cardiff University, Cardiff, United Kingdom, CF10 3AT*

*\*Corresponding Author (email: cuminst1@cardiff.ac.uk)*

*Abbreviated title: New Method for Slope Stability Modelling*

**Abstract:** Through the study of shallow surface slope movement in Britain, and globally, a *Python* script was developed, based off Farnham’s original code. The code developed would create a contour model to represent inclinometer data collected in the field. These models provide a clearer understanding of the slope conditions, whilst allowing the user flexibility in data representation. This is achieved with ease through the graphical user interface developed with *Tkinter*, which when published as a .exe file, allows anyone simple use of this modelling software. This is the first iteration of *IncModelling* and shall continue to be developed.

**Supplementary Material:** All material worked on, with example datasets, is available to download at: *github.com/TobyJACumins/IncModelling*

Slope stability analysis is crucial to the prevention of damages that may be a direct result of shallow surface slips. Damages resulting from these slips are usually very minor, causing few casualties and injuries. However, slips are occurring at an ever-growing rate leading to increased cost in damage, with the soil movement in the UK having a financial impact of £300-500 million per annum (Pritchard *et al.* 2013). Without adequate investigation and continual surveillance, loss of life or historical buildings could occur.

Current inclinometer models represent slope movements poorly, these are hard to comprehend the extent of movement and do not spread informative information to the public. The research carried out aims to create contour models that can reflect the slight shallow surface inclination changes recorded by inclinometer probes, to allow the installation of early prevention measures to mitigate cost due to slope failure.

The Code that this project is developed from was authored by Farnham (2016), which has been manipulated to create an all-encompassing program that anyone can use and easily understand. This will produce homogeneous contour plots to easily assess and compare an entire transact of inclinometer data. Through regular investigations, slope instability will hopefully decrease as a result.

**Methodology**

Manipulation of Farnham’s code was a long progress of trial and error, made harder by the process of having to get to grips with *Python 3.5*, as *Python 3.5* was the language Farnham used to develop his code. To make development of the program easier a separate program, *Atom* (2017), was utilised for ease and to aid in presentation of what was being worked with.

To begin, I assessed Farnham’s code, re-working sections in a manner that could be built upon. This was done by creating functions that were capable of running individual components of the base code. These function incorporated *Python* packages such as *Matplotlib* (2017a) and *Numpy* (2017), to achieve the creation of the models.

An additional package used was one called *Tkinter* (Tkdocs, 2017), this package was integral to creating a graphical user interface (GUI) that was user-friendly but still ran efficiently with the parameters set by the user. Next followed a period of trial and error to firstly get the GUI to behave correctly with the code, as well as set up additional parameters, giving the user more control. These were features like titling the graph, manual input/output .csv browse selection and deciding on the style of contour model that was output. Various models that are produced from interpolation methods were researched. Four methods proved useful for the project and so the codes output potential was reworked and expanded to accommodate for these different outputs, including the incorporation of four gradient styles for more user customization.

The code was then created into an executable (.exe) file rather than a *Python* (.py) file, so anyone could easily download and run the program straight off any Windows computer, negating the need to download additional components and having prior *Python* knowledge. One error that came apparent at this point was that the code was based around a strict .csv representation of inclinometer data. To get around this there were two option: Firstly, to force all users to simplify their data collected to the style in which the code is based around, or Secondly, to design the program to have the potential to detect the information required and produce a graph straight from .csv file chosen as the input. I decided to go for the first option, this is because large portion of code would need to be re-worked to find the correct data, which could potentially categorise data into the wrong axis and produce an incorrect graph. The first option is relatively easy as data sets can be opened in excel and re-arranged accordingly.

**Program Development**

*Data Acquisition and Processing*

Inclinometer data can be saved into three different formats on most current inclinometer probes while field investigation is being conducted. This program is firstly dependent on data being saved in .csv file format, and secondly dependent on the way in which the data is represented. The layout that each inclinometer saves files differs dependent on brand, so steps must be followed to format data into the correct format, shown in supplementary material. Most inclinometers will produce two data arrays which are perpendicular from one another, this builds a greater understanding of the sense of slope movement by allowing analyse in 3D (Durham Geo-Enterprises, 2011). However, this program is designed to singularly create a model for one axis at a time. Focus then is on the A-axis as this is predominantly the downslope axis, resembling the slope of most slip. Error can arise in the current iteration of the program published with this paper, if A-axis data is not isolated from B-axis data.

Much of the development time was spent on researching and trialling various methods to interpolate the inclinometer data. *Matplotlib*, a *Matlab* addon to the basic *Python* package, was used to create the contour plot. There were three possible methods that had discrete methods of interpolation, these were: *Griddata*, *Imshow* and *Contourf* (Shepard, 1968). Due to the nature of how data was laid out and how the program read the .csv file *Griddata* and *Imshow* potentials were restricted. When these methods call data, they project the values on a one-to-one basis, like the pixels of an image, producing the highest quality image of the inputted values but this meant that interpolation had little to no effect (Matplotlib, 2017b). *Contourf* however, draw shapes of the inputted values that allow for deviation when interpolating them. By altering the intervals to which *Contourf* drew shapes a change to the “Resolution” of the model was observed. To paraphrase, the number of intervals between each change in colour saw the “Resolution” of the model can be increase (increasing interval count – decreasing interval gap) or decrease (decreasing interval count – increasing interval gap).

The accuracy of the models will change slightly dependent on the “Resolution” used, however much of the uncertainty will be from human and instrumental error whilst collecting the data in the field. Lower “Resolution” will mean for less accuracy but may be applicable over a vast area that is being assessed.

The use of colour can dramatically change a way in which a model is represented, *Matplotlib* offers a parameter called *cmap* which allows a vast selection of gradients to be used. Working through many of the colours on multiple datasets four gradients were determined the most useful. Each colour comes with its own individual advantage for different circumstances in which the data needs to be portrayed.

*Graphical User Interface Development*

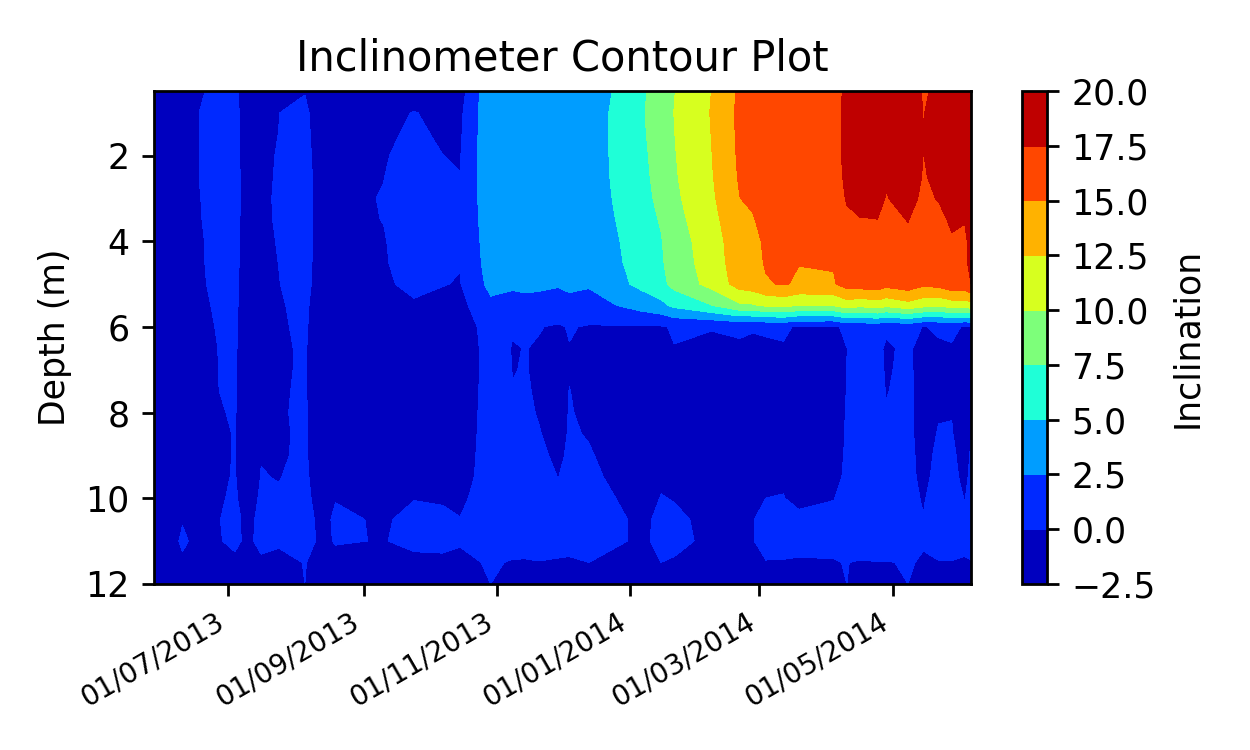
From the outset, the plan was to make a simple *Graphical User Interface* (GUI) that anyone could use quickly and effectively. This was achieved by running a couple of processes in the background without declaring them for user interaction, such as which data is assigned to each axis which I have predetermined or the interval spacing used for each axis which the program calculates and determines itself. A process which the user only see the surface of is when choosing the input and output files directory, where for the input the user is forced to use .csv data only whilst on the other end the model has to be output as a .png image. This is done to reduce error from confused users as well as reduce error from multiple formats not cooperating.

A minor issue that was run into was having the use of two sets of radio buttons, the problem being that only one button could be selected at once, the work around for this was to merely setup a new variable for each set of radio buttons that could act independent of one another.

The code processed data correctly, with adequate functionality from a created GUI, the last task was to create the .py file I had been working in to a .exe file so that any user could download the program to run without any knowledge of the work going on behind the scenes. To achieve this, a *Python* package was used called *Pyinstaller*. This “freezes” the .py into a stand-alone .exe thanks to transparent compression (Cortesi, 2017).

**Results**

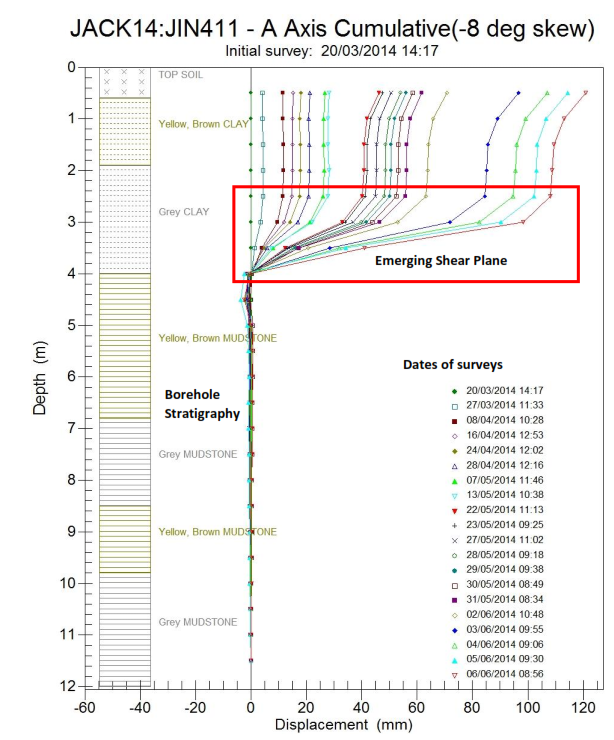
Inclinometer data was processed through this code, and Fig. 1 was created. Models can become patchy dependent on uncertainty produced by the probe data, as well as how regularly a site was visited, the more data the more accurate the model will be.

****So how can this model be read? This is predominantly up to the interpretation of the user. However, the model clearly shows the progressive change in incline that the bore hole is experiencing from slope instability. My personal interpretation changes dependent on the “Resolution” used. If Default or Halved “Resolution” is selected, then I would take the base of the first interval change to be the approximate plane of failure, in Fig. 1 this would be from the 0-2.5 to the 2.5-5 interval change. With the higher resolution models, it could be the case that small movement occur in stable units and you could have one, two or three planes that are failing at different rates in one slope section. You will need to consider all data collected from site, be it movement patterns or geology to completely understand the movement occurring on a slope. Additional models of SampleData(A-Axis) are found in Appendix.

**Discussion**

*Comparison to Existing Models*

The Model created represents slope conditions to a high standard, allowing user flexibility and continual observation of field data if its collected. However, by the very nature of making a new method to represent data, how does This model compare to current models that are used? Below, Fig. 2, is the way in which this data is commonly represented. Both of these models show the data but the contour model representation is far easier to read and understand the change in inclination that is going on. Not only does this make interpretation far easier, but it makes comparing models simpler by observing specific colour(s) variations. Data is able to be isolated to individual dates, showing the exact inclination at exact depths, although this could be done in Fig, 2, the outcome is neater in the contour model, allowing for direct comparison to inclination change in neighbouring boreholes at the same date.



*Further Development*

As the program stands it can be developed further to accommodate for additional parameters as well as dig deeper into data that is able to be collected from the field. This could firstly be done by allowing the program to read a .csv data set of both A-axis and B-axis data, so that a user could choose which axis to make a model for. This could be done by various methods the simplest being the addition of a new column in the .csv layout, produced by a double comma. The function that reads the required data for each axis could then store two sets of data for each axis, a set for A-axis and then B-axis.

The advantage to having two sets of data stored from a single bore hole is that two graphs could be made simultaneously perpendicular to one another, this will give a 3D sense of movement of the slope. This can be taken one step further by applying Pythagoras’ theorem, whilst understanding the sense of motion (loss of negative nature as squared), to produce a model that will show motion parallel to the plane of most slip, this will also be able to obtain the bearing down slope that slip in moving parallel to.

The potential to expand the .csv reading capabilities could be expanded further to allow any .csv layout to be processed through the program. This could be accomplished by re-working the functions that read the .csv file, collecting axis data, to identify various pre-set trigger words/values, that would inform the code where to start and stop selecting key data.

Another feature would be to have a user specified key, this would allow a user to opt to have a predetermined key so they could be consistent with their inclinometer models across transect(s) being investigated. Currently the code works out the max and min values for interpolation and assign a gradient to hit that range. This slight change would allow a user to follow across a boundary (contour interval change) in which they interpret slip to be occurring, whilst working quickly through a large data set.

The final addition to make this an ideal package for slope stability modelling, would be the possibility of creating a model that rather than revolving around the change in inclination, it would instead analyse the rate of movement of the slope. To achieve this, you would need to look at inclination (Z value) and see the rate in which it changes with time.

**Conclusions**

Interpolation of the inclination data collected in the field into contour plots has been an effective way to display data clearly showing planes in which slope instability originates from throughout the borehole and through the GUI designed creating models can be done very quickly, allowing for large transects of boreholes to be analysed within a short space of time. As this is the first iteration of development many features could not quite make it onto the final program, the code produced alongside this paper is constructed in a manner in which these additions, mentioned above, can be implemented with little to no reworking. In the second iteration of development, the points mentioned in the further development section are intended to be implemented into the IncModelling code, allowing for more user customisation and flexibility.

The following people ensured the completion of this paper: Thomas Cumins, Huw Fernie and Huw Fulcher for user testing the code, ensuring it is presented in a neat manner. Ardit Sulce, who authored

*Udemy* Python course that was undertaken back in February 2017.

**References**

Atom, 2017. *Atom Flight Manual*. [online] Available at: atom.io/docs [Accessed 2 Mar. 2017].

Cortesi, D. 2017., *PyInstaller Documentation*. [online] Available at*:* pyinstaller.org/documentation [Accessed 17 Mar. 2017].

Durham Geo-Enterprises, 2011. *Digitilt Inclinometer Probe 50302599*. Nova Metrix.

Gernon, O., 2015. *Identifying the Triggers and Mechanisms of Ground Instability in Jackfield.* BSc. Cardiff University.

Matplotlib, 2017a. *User’s Guide* [online] Available at: matplotlib.org/contents [Accessed 01 Mar. 2017].

Matplotlib, 2017b. *images\_contours\_and\_fields example code: interpolation\_methods.py — Matplotlib 2.0.0 documentation*. [online] Available at: matplotlib.org/contents [Accessed 11 Mar. 2017].

Numpy, 2017., *Numpy Developer Guide.* [online] Available at: docs.scipy.org/doc [Accessed 1 Mar. 2017].

Pritchard, O., Hallett, S. & Farewell, T., 2013. *Soil movement in the UK – Impacts on critical infrastructure*. ITRC Working paper series. Cranfield University, Swindon.

Shepard, D., 1968, January. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference* (pp. 517-524). ACM.

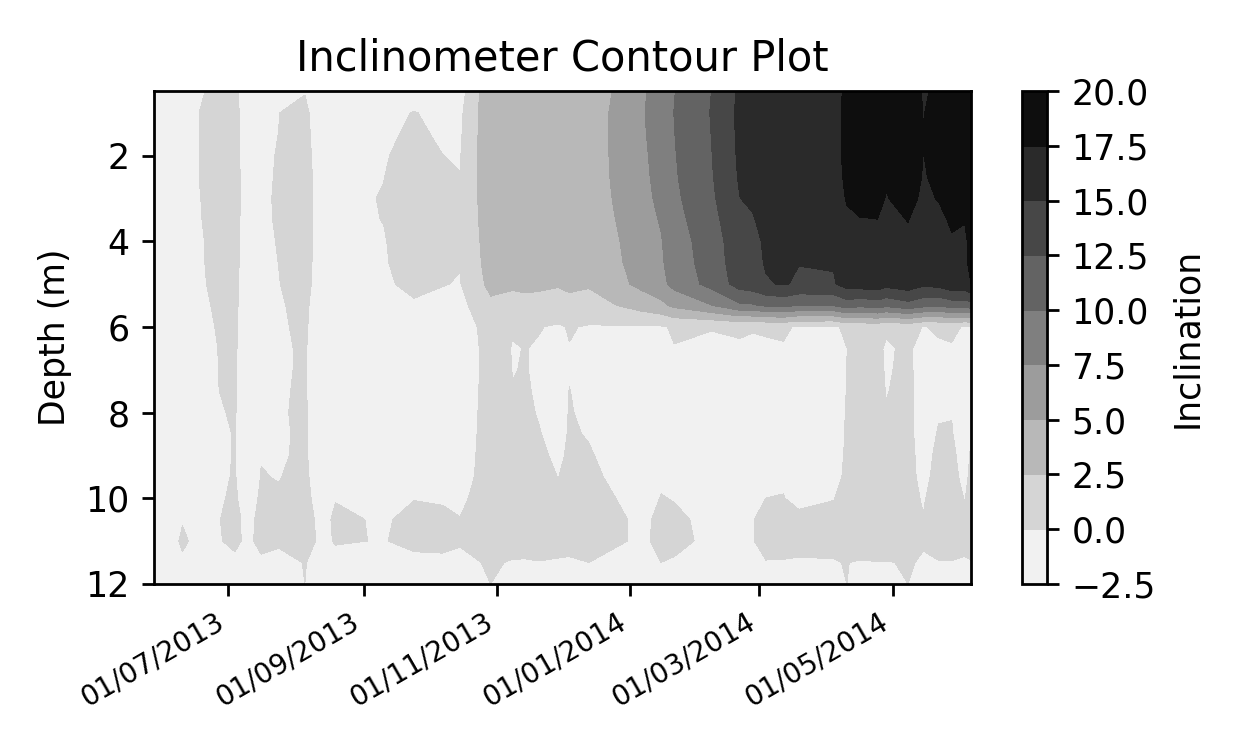
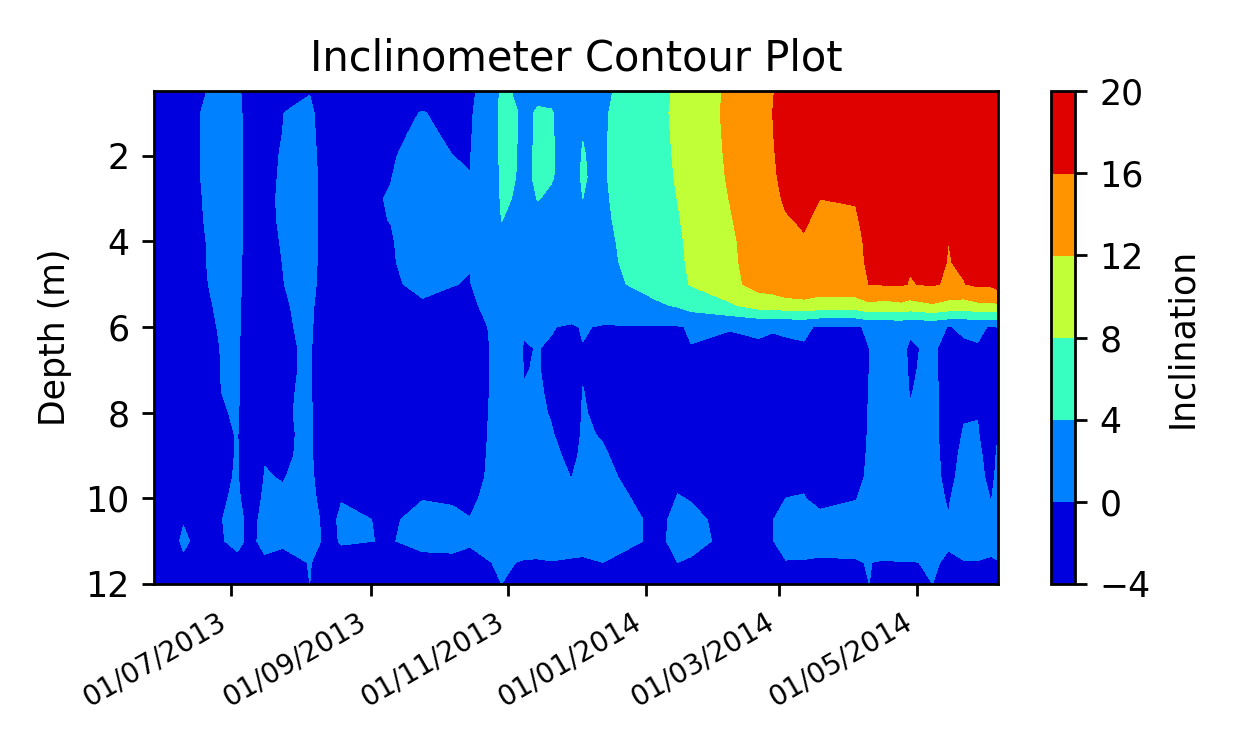
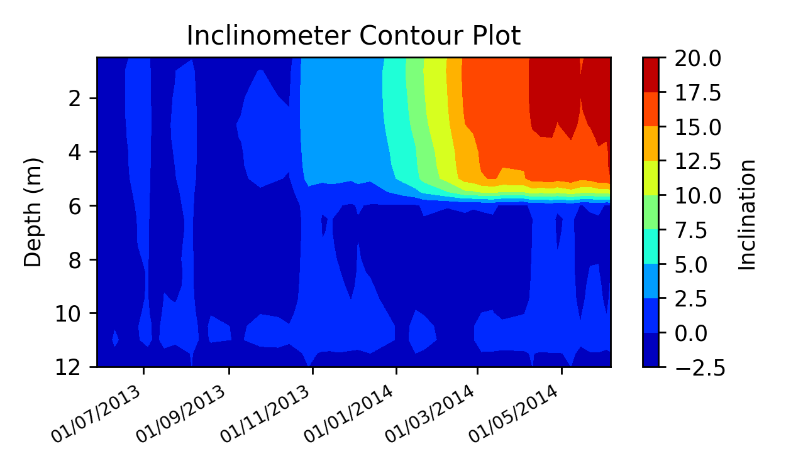
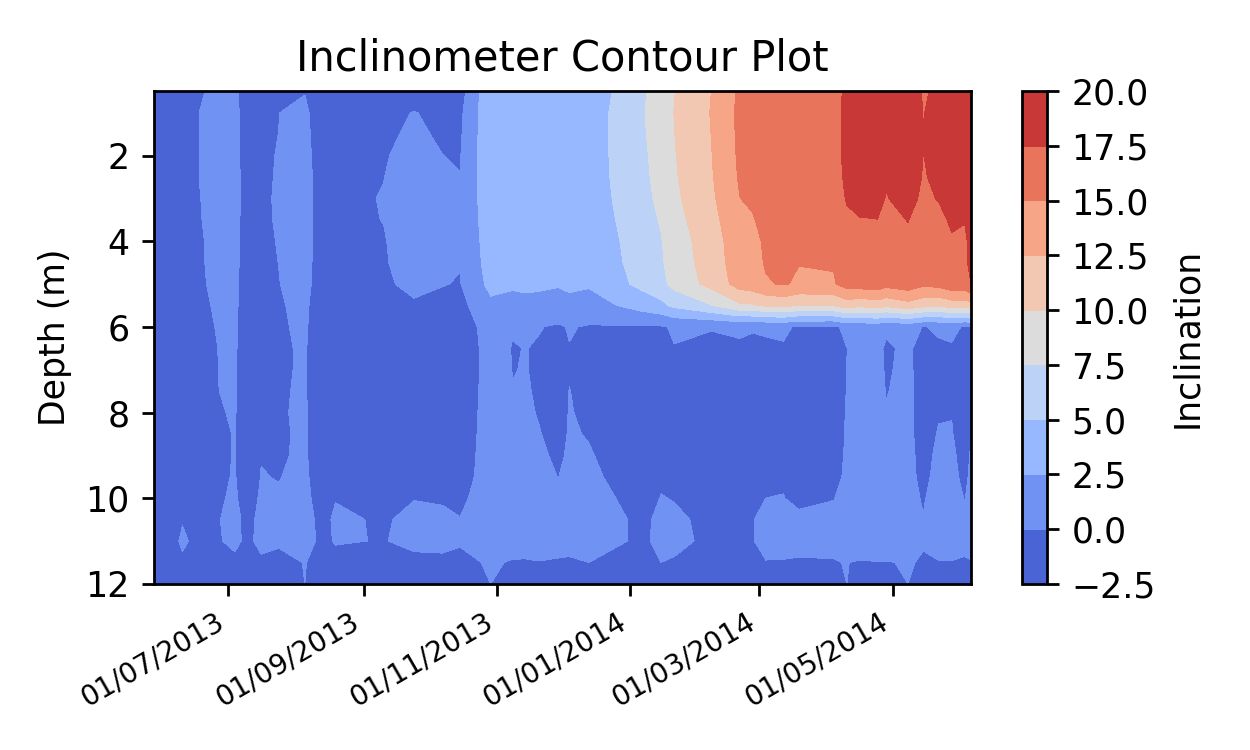
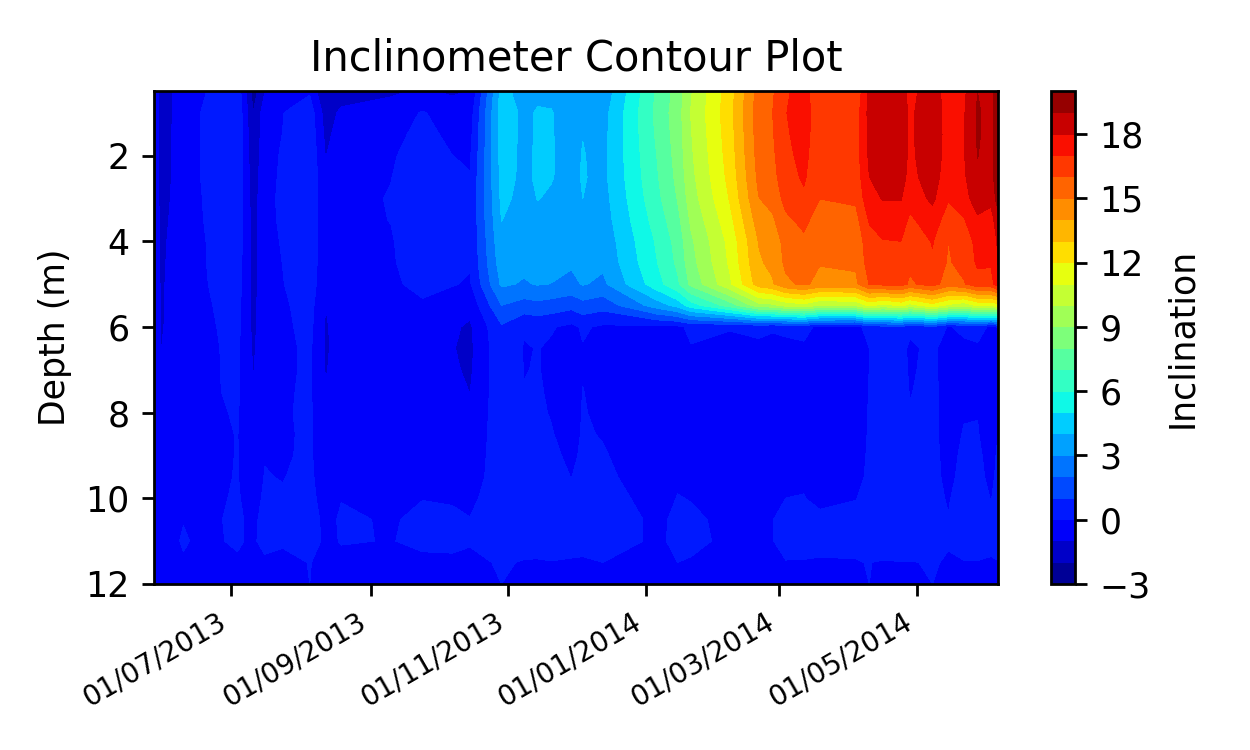
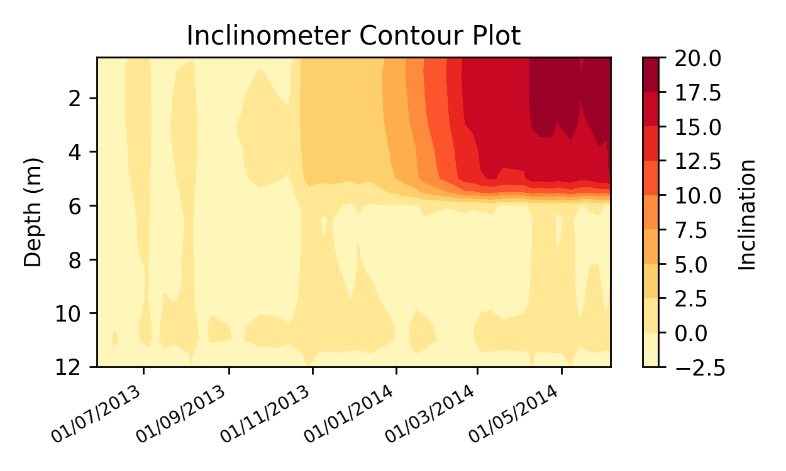
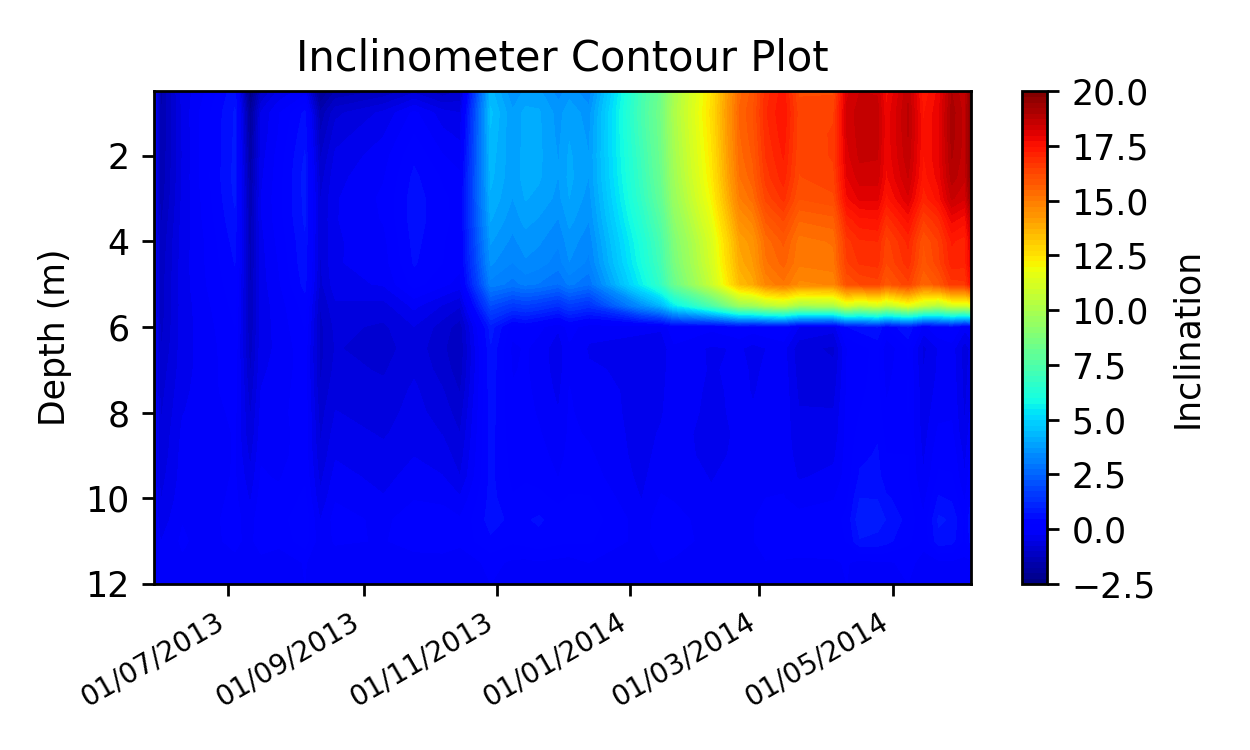
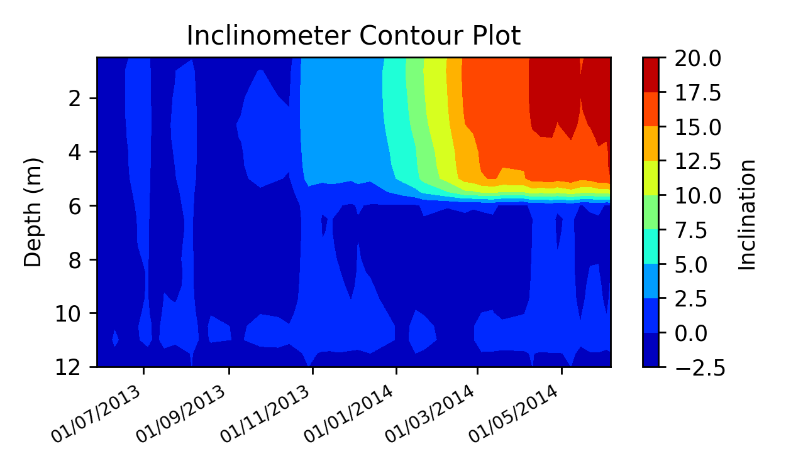
Tkdocs, 2017. *TkDocs Tutorial*. [online] Available at: http://www.tkdocs.com/index.html [Accessed 1 Mar. 2017].

**Figure Captions**

**Fig. 1.** A contour inclination model, produced through developed (IncModelling) program. Contour plot interpolates between data stored in a z axis of inclination change over time. This change is respective of starting inclination, not change in inclination between each trip.

**Fig. 2.** A Standard cumulative inclination graph commonly used. The plane of movement is indicated by the change in angle, represented by the inferred incremental displacement between survey. Software would not generate stratigraphy, so in comparison both models would require user personalisation (Gernon, 2017).

**Appendix**



Inclinometer Demo Default/Default

Inclinometer Demo Double/Default

Inclinometer Demo Squared/Default

Inclinometer Demo Halved/Default

Inclinometer Demo Default/Default

Inclinometer Demo Default/Diverge

Inclinometer Demo Default/Autumn

Inclinometer Demo Default/Greys

Appendix 1, All model variability shown, top four show “Resolution” change, bottom four show gradient change. Linked to Fig. 1, Results section.