A tool for soil nail wall design optimisation using Slide2 and Python

M. P. Crisp[1], O. Davies[2]

[1]Mott MacDonald, 22 King William St, Adelaide, Australia; PH (08) 7325-7325; email: michael.crisp@mottmac.com
[2]Mott MacDonald, 383 Kent St, Sydney, Australia; PH (02) 9098-6800; email: owen.davies@mottmac.com

ABSTRACT

Optimised soil nail wall designs can result in reduced material costs, reduced construction program and reduced construction risk. Optimisation through assessing a large number of models manually has, in the past, been practically unfeasible due to the user time involved in model development and interrogation. The use of interfacing tools with the python scripting language, either through an included Python API or through modifying text-based models directly, has the potential to significantly reduce the time required for rigorous optimisation. This time reduction through digital tools, will lead to realisation of all the benefits design optimisation allows.

To demonstrate the above, this paper presents a python tool for assisting in the optimisation of soil nail wall design. It is implemented through the 2D limit equilibrium method software, Slide2, although may be applied to other software. Using the tool, an engineer can determine optimal soil nail lengths for a given slope angle, soil nail inclination, nail spacing, nail location (offset), and number of nails. Combinations of these parameters may also be assessed automatically to find a globally optimal solution. An example is given where a design was optimised simultaneously across multiple scenarios with different safety factors (e.g. flooding, earthquake, etc.). The use of this tool allows the engineer to identify the influence of key variables and quickly calibrate the optimised design to the specific constraints of the project.

Keywords: limit equilibrium method, soil nail wall, optimisation, python

1 INTRODUCTION

Geotechnical engineering software is becoming increasingly capable of interfacing with programming languages such as Python. This interfacing allows for extended software capabilities beyond that implemented by the core software, such as design optimisation and sensitivity analysis. The benefit of such an approach is that bespoke solutions can be developed which greatly speed up design, or even optimise designs to a degree that may not have been previously feasible due to time limitations.

For context, a number of programs have an Application Programming Interface (API) whereby a programming language can interact directly with the program, with Python (Python Software Foundation, 2021) being used in several applications.. However, there are other methods for integrating languages with software as well. For any program that stores its model information in a text-based format, code can be created to read, modify, and write these models. In essence, the user can create models that are modified in desired ways, such as alternate soil properties, load placement or magnitude etc. With sufficient coding, a model could be created entirely from scratch according to user inputs.

Further, these model modification tools can be combined with optimisation techniques in order to optimise designs. For example, there are a collection of evolutionary algorithms, such as the genetic algorithm, which use iterative population-based techniques to minimise a value (such as cost), and that can optimise several parameters simultaneously. Here, population-based means that within a given iteration, a large number of assessments are performed using different inputs, which collectively inform the population for the next iteration.

There's at least one example of finite element method (FEM) software Plaxis (Bentley, 2022) being used to optimise parameters of a soil nailed vertical cut using and comparing various evolutionary optimisation algorithms (Benayoun, et al., 2021). The downsides of these methods are that:

1. They are largely a black box, whereby the optimisation process is opaque.
2. They are not necessarily guaranteed to find a globally-optimal solution, particularly when complex solution surfaces exist with multiple local optima. A degree of randomness is involved.
3. A large number of analyses can be required (the number of iterations multiplied by the size of the population). Essentially, these methods can be "overkill" for simpler problems.

This paper presents an example of a tool being developed through integration with established software, in the form of a soil nail configuration optimiser for the 2D Limit Equilibrium Method (LEM) (Bishop, 1955) software Slide2, a Rocscience program (Rocscience, 2022). Historically, designing soil nails for a retaining wall or cut slope could be fairly time-consuming due to:

- multiple geological sections
- construction stages
- multiple scenarios (earthquake; flooding; over-excavation etc.)
- different safety factors for each scenario

Each of these must be checked against the design to ensure the design requirements are satisfied. The tool, described within, solves this problem by determining a soil nail design by optimising across all scenarios and cases simultaneously, and on a row-by-row basis.

Here, optimisation means to minimise soil nail length for a given wall configuration. Other metrics, such as cost, are not considered. The optimisation methods discussed in this paper are well-established in literature, and as such are not the main focus of this paper. Rather, the novelty here lies in its application to multiple wall sections in a global and efficient manner, and as a proof-of-concept for enhancing established geotechnical software with scripting, regardless of the presence of a scripting API.

2 SOIL NAIL OPTIMISATION TOOL

The steps involved in the soil nail optimisation process are given in the flow chart in Figure 1. Note that this flow chart includes steps for multi-parameter optimisation, whereby variables such as vertical and horizontal nail spacing, and nail angle can be optimised. If the user wishes to simply optimise the lengths of nails that are already in the models, then the “optimise next combination” loop can be ignored.

2.1 Interaction with Slide2

Interaction with the Slide2 software is done through modification of the saved files. As is typical with Rocscience software, the models are text-based. There are various file types available, including the pure text .sli format. The alternative .slim format is simply zipped (compressed .zip) text files, with the “.zip” renamed to “.slim”. The user will most often work with multi-doc files which are not supported by this optimisation tool. However it is a straightforward process to export the multiple models into a collection of individual .slim files. Python is well-equipped for both reading and writing .zip files.

Custom functions were written for interpreting the Slide2 model information, as well as for creating and manipulating soil nails. The only modification needed to the model file is to delete the lines containing existing soil nail geometry, and save the new/updated nails in an identical format. Into-the-page spacing is modified by changing the relevant parameter value in the list of soil nail properties.

The safety factor, also known as the factor(s) of safety (FoS) is obtained by running the model files through the slide2 compute program (aslidew.exe) and reading the text-based results file. The ability to both interact with the Slide2 model using python, and to read in the resulting FoSs, is the key part of this software, regardless of the optimisation algorithm employed.

2.1 Optimisation algorithm

The bisection method is a robust single-parameter optimisation method, as it is guaranteed to converge on the solution within a consistent timeframe when the solution lies between a specified upper and lower bound (Khoury & Harder, 2016). In this case, nail length is being optimised. The method starts by the user defining an upper and lower bound nail length (i.e. a maximum nail length (L_{max}) based on property boundaries, construction constraints etc. and minimum nail length (L_{min}) which well be zero for most cases). To find this optimal length it is assumed longer nails will achieve a higher FoS and shorter nails a lower FoS. An iterative procedure is then used whereby the nail length is continually bisected until the shortest nail length which satisfies the target FoS is found.

The bisection method consistently halves the search space in each iteration. There are other methods which can eliminate more than half of the search space in each iteration, and can potentially converge faster. These speed improves either rely on assumptions (i.e. Regular Falsi assumes the function is approximated by a straight line), or on additional information (Newton-Raphson method uses the derivative of the solution space) (Khoury & Harder, 2016). However, the former type are vulnerable to choke conditions, which could cause significant slowdowns. The latter type could fail to converge entirely. For these reasons, the bisection method is deemed most suitable.

The LEM method used for calculating FoS is also is fast enough not to require advanced approaches to minimise iterations and run times. Multi-parameter optimisation methods were also not considered due to reasons discussed in the introduction. However, the reader may read more about a range of evolutionary optimisation algorithms in (Omid, et al., 2017).

The algorithm assumes that the bottom-most nail will be longest and as such, the bottom row (adjusted in length with all above rows) is optimised first. This process is elaborated in a later example. In general this is a valid assumption. Although there may be cases where soil around the base of the cut is sufficiently long that the bottom-most nail can be removed entirely. This case is currently not accounted for, and the engineer should consider this possibility when reviewing the results. In the future this can be overcome by reversing the order in which rows are optimised, such that the top ones are first.

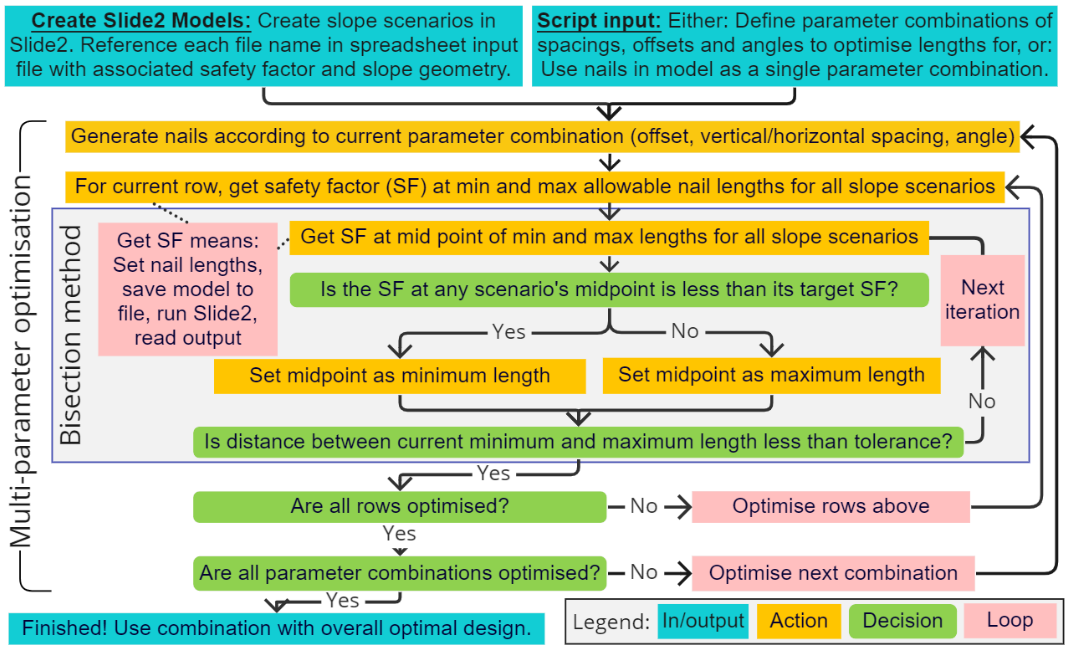

Figure 1: Flow chart describing the process of optimising soil nails.

2.2 Limitations

At present a number of limitations exist for the soil nail optimisation tool, as follows:

- All soil nails within any given model must all be of the same nail type (although different models can have different nails). Here, nail type refers to a set of material properties, such as tensile capacity, bond strength etc. Note, all Slide2 support types are supported (e.g. end-anchored, geosynthetic etc, including bond strength based on embedment material).
- A given soil nail row must have a constant head elevation, as this is how rows are identified across different models. However not all rows need to be present in all models.
- Multi-parameter optimisation involving vertical nail spacing, can not properly account for soil lift (excavation) stages. This is because the excavation levels are often dependant on the nail elevations, and the slope geometry cannot be changed in the tool.
  - This largely restricts vertical spacing optimisation to the final slope profile. The engineer should subsequently confirm that the design is valid during construction stages.
- Probabilistic analyses and transient water conditions are unsupported. In the case of transient water there is an additional file and solver program that the script is not yet hooked into.

3 EXAMPLE

An example of a slope section is considered. The geology and dimensions of the profile are shown in Figure 2. The slope is excavated from ground level at the top of the fill layer. Each lift (excavation for a nail row) is taken to 0.5m vertically below the nail head elevation. Nail lengths were fixed to a minimum and maximum of 0m and 10m respectively and with optimisation intervals at 0.5 m. Multiple scenarios for the profile are considered in the optimisation, as listed in Table 1. As such, the tool should produce a design that is not only valid and safe for the profile, but globally optimal as well.

In other words, multiple construction stages, and loading/water conditions are assessed within each optimisation iteration. Different geologies, soil or nail parameters could easily be added as additional cases in the optimisation, meaning that the results would not be sensitive to such variations, however this is not the current focus of this paper. In each iteration, the soil nail lengths are changed once across each case, based on the previous iteration. The nail lengths of each row are the same across all cases. Each case's FoS is calculated to check whether it satisfies its corresponding target value in that iteration. This means that all FoS are satisfied for all cases in the final result.

Due to space constraints within this paper, multi-parameter optimisation is not used here. Instead, the four nails, placed in the model by the user, have their length optimised. Nail location, inclination and spacing (both horizontal and vertical) are not modified within this example. As mentioned previously, this means the "optimise next combination" loop in Figure 1 can be ignored.

3.1 Conditions and material properties

The soil/rock material properties, and the soil nail properties are included in Figure 2. In addition, the soil nail bond strength is 23.3 kN/m in the "Fill" and "Residual soil" layers, and 58.3 kN/m in the remaining layers. A Mohr-coulomb strength model is used. The punching shear capacity of the shotcrete facing is not considered here.

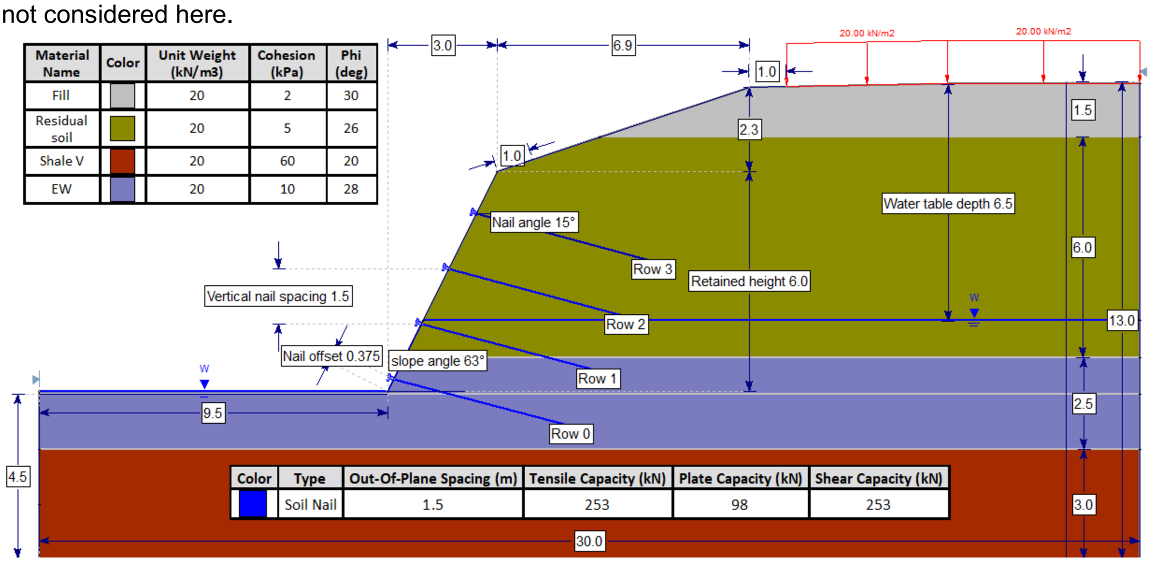

Figure 2: Dimensions and properties of the slope model.

The soil is considered undrained during the first lift, and during the earthquake analyses. For the earthquake case a horizontal seismic load coefficient of 0.09 has been adopted. In the flooding scenario, the water table within the soil is raised to 5.5 m below ground level. During all construction stages, including overcut, a surcharge of 10 kN/m^2 extended to 1 m from the crest of the retained slope.

3.2 Safety factor calculation

The FoS was determined from examining circular slip-surfaces, using the Morgenstern-Price method and a half sine interslice force function (Morgenstern & Price, 1965). The worst case slip surface was found using Slide2's "Auto Refine Search" method, which is a robust iterative search method (Rocscience, 2022). Note that all LEM and slip surface search methods, that are supported in Slide2, can be used. Multiple LEM methods can be used simultaneously, where the lowest FoS is taken.

Non-circular slip surfaces can also be used. However, care must be taken to ensure that small shallow slips (e.g. in the weak fill layer) don't define the global FoS. If such slips have a FoS that is below the target, then the slope is incorrectly considered "infeasible" and cannot be optimised.

3.3 Results

The design converged in 23 iterations, which comprised a total of 207 individual FoS calculations (23 x 9 scenarios). This is a relatively quick analysis, taking less than 30 minutes to complete. The final design lengths are given in Figure 4 Furthermore, the final FoS for each scenario is given in Table 1, along with the target FoS, and an indication of which section(s) was critical in the design of each nail row. This latter detail is useful as it informs the engineer which sections and/or slope elements to focus on in order to refine the design. These tables are taken from a CSV file produced by the tool.

The results confirm that an optimal solution has been achieved, for this number and spacing of nails, and for this slope profile. This is evident by the final minimum FoS being equal to the target value of 1.5, i.e. shorter nails would result in an invalid value. An exception to this would be if higher nails could be longer than lower ones. As mentioned previously, this can be accounted for by reversing the order that rows are optimised in. Note that the rows being optimised in a specific order, as opposed to having rows of arbitrary, independent length, serves to constrain and greatly simplify the optimisation problem.

Table 1: Final FoS for each case, along with which case(s) were critical for each nail row.

Scenario	Lift 1	Lift 2	Lift 3	Lift 4	0.5 m Overcut	Drained	Un-drained	Earth-quake	Flood
Final SF	2.28	1.36	1.37	1.34	1.41	1.50	2.29	1.84	1.43
Target SF	1.30	1.30	1.30	1.30	1.30	1.50	1.50	1.30	1.30
critical at nail(s):		3	2	0, 1		0, 1			

The "Lift 1" case notably has no soil nails present, as the critical stage occurs prior to the installation of the first nail. As such, its safety factor does not change across iterations. The case was included for the script to determine that the slope was viable, although an engineer could conceivably do this manually before hand then exclude it from optimisation. Nail 3 (the top nail) first appears in the "Lift 2" case.

Furthermore, the tool also produces a PDF showing how the design evolved across iterations for all sections. An example of this, for a different model where all nail rows have different lengths, is shown in Figure 3 for the first two rows. Row 1 optimisation begins after row 0 is optimised.

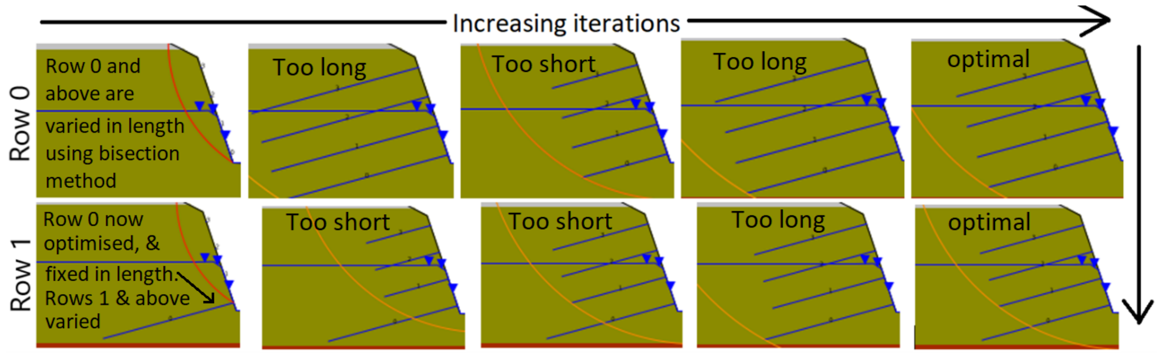

Figure 3: Selected iterations from the optimisation of the first two rows (starting from bottom row).

The final design and worst case circular slip surface (the "Drained" scenario) is shown in Figure 4, along with soil nail lengths. Upon comparison with Figure 2, which is a screenshot of the same model from the Slide2 modeller program, a lot of similarities are noted. The applied loads and water table are correctly shown, along with the geology and nails using the correct colour. This is achieved automatically as part of the tool's interpretation of the model file. Inspection of the full PDF (not shown here) can give the engineer further insight into how the design was achieved, and what the controlling factors were.

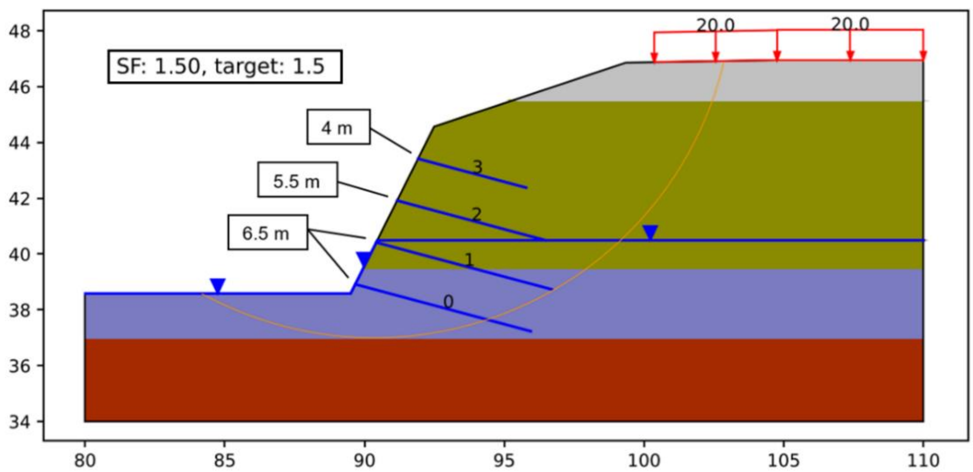

Figure 4: Tool-produced output section showing the optimised design and worst-case slip surface for the final drained scenario, along with soil nail lengths for each row.

3.1 Potential further analysis with existing tool features

Only one section was considered here, with a single set of soil properties. Soil nail designs can be sensitive to soil variability. As such, additional sections can be added with different geological profiles or alternate soil or nail material properties. There is no limit to the number of cases that can be accounted for in the optimisation of a single design. For example, each of the sections shown in the long section of Figure 5 could be incorporated, with the nail lengths being consistent within each row across all sections. Since vertical and horizontal soil nail spacing can also be optimised with the tool, using the multi-parameter optimisation mode, soil nails can effectively be designed in "3D"; i.e. optimal soil nail locations in the longitudinal and vertical directions can be identified. Alternatively, each of the sections could be optimised individually, such that nail lengths also vary longitudinally along the wall. Currently, non-horizontal rows (i.e. parallel to base cut) are not currently supported, but can be in the future. Analysis of the iteration history can provide insight into the FoS sensitivity of the final design.

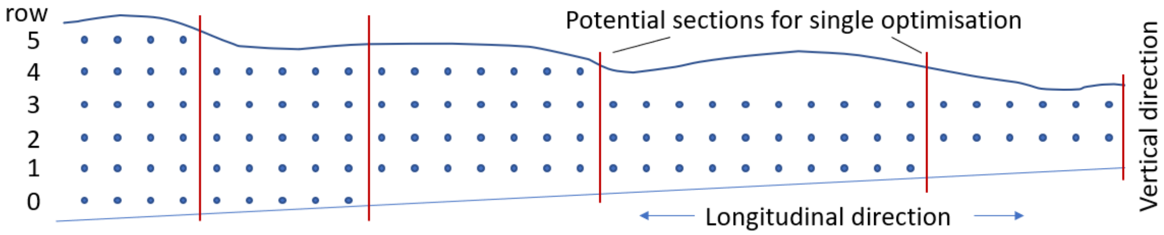

Figure 5: Example sections that can be used to optimise a wall along its entire length.

4 CONCLUSION

This paper has demonstrated a methodology and python tool to assist in optimising the design of soil nail walls using the Limit Equilibrium method. The optimisation tool accounts for a wide range of scenarios, including those related to construction, and extreme conditions such as earthquakes and flooding. This tool serves as a proof-of-concept for enhancing functionality of commercial geotechnical software with scripting, regardless of whether the software has a scripting interface.

The application of such tools is not intended to produce a finalised design. The designer remains responsible for ensuring any design meets design scope and code requirements. These tools are intended to hasten the optimisation process. A quicker process allows the designer to consider more scenarios and understand the benefits / drawbacks of design changes. This leads to the potential for more robust and economic designs with all the incumbent economic societal and sustainability benefits.

REFERENCES

Benayoun, F., Boumezerane, D. & Bekkouche, S., 2021. Techniques for Optimizing Parameters of Soil Nailed Vertical Cut. *Journal of Civil Engineering,* 16(1), pp. 131-145.

Benayoun, F., Boumezerane, D., Bekkouche, S. & Ismail, F., 2021. Optimization of geometric parameters of soil nailing using response surface methodology. *Arabian Journal of Geosciences,* Volume 14.

Bentley, 2022. *Plaxis 2D - Complete 2D Geotechnical Analysis Software.* [Online] Available at: https://www.bentley.com/en/products/product-line/geotechnical-engineering-software/plaxis-2d

Bishop, A., 1955. The use of the Slip Circle in the Stability Analysis of Slopes. *Geotechnique,* 5(1), pp. 7-17.

Khoury, R. & Harder, D., 2016. *Numerical Methods and Modelling for Engineering.* 1 ed. Berlin/Heidelberg: Springer.

Morgenstern, N. & Price, V., 1965. The analysis of the stability of general slip surfaces.. *Geotechnique,* 15(1), pp. 79-93.

Omid, B., Solgi, M. & Loiciga, H., 2017. *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization.* 1 ed. Newark: John Wiley & Sons.

Python Software Foundation, 2021. *Python Language Reference.* [Online] Available at: http://www.python.org/

Rocscience, 2022. *Auto Refine Search.* [Online] Available at: https://www.rocscience.com/help/slide2/documentation/slide-model/slip-surfaces/circular-surfaces/auto-refine-search [Accessed 28 July 2022].

Rocscience, 2022. *Slide 2 - 2D Limit Equilibrium Analysis for Slopes.* [Online] Available at: https://www.rocscience.com/software/slide2