Proceedings of the XVIII ECSMGE 2024
GEOTECHNICAL ENGINEERING CHALLENGES
TO MEET CURRENT AND EMERGING NEEDS OF SOCIETY

ISBN 978-1-032-54816-6
DOI 10.1201/9781003431749-57

ECSMGE 24
XVIII EUROPEAN CONFERENCE
ON SOIL MECHANICS AND
GEOTECHNICAL ENGINEERING

Slog: an open geotechnical software

Slog: un logiciel libre pour le calcul géotechnique

J. de Sauvage[*]
Université Gustave Eiffel, Bron, France

[*]*jean.de-sauvage@univ-eiffel.fr*

ABSTRACT: In the era of complex numerical modelling, the geotechnical engineering practice always resort to simpler numerical tools, exhibiting the key phenomena. Such software is important for teaching since it allows to better understand the principles underlying the design of structures. They are also important for expertise or research on the one hand because they are easy to use and on the other hand because they establish a clear link between the assumptions made and the results. Most of this software is distributed under proprietary licenses, generating a “black box” effect and offering little flexibility to take a particular situation into consideration. From a teaching point of view, it is often useful to consider aberrant hypotheses in order to show their consequences. The Gustave Eiffel University is therefore developing geotechnical calculation software, entitled Slog. Open access and developed with Python, a widely used language, this software aims to federate a community of user-contributors, particularly among geotechnical teachers and students. In order to be able to easily compare different potential structures on the same site or, conversely, to be able to assess the relevance of a structure in different sites, the user is led to independently define the ground model and the anthropogenic actions. The intersection between both is carried out automatically, allowing the user to easily modify the properties of one or the other automatically. This flexible and scalable process is useful for teaching. The detailed example of the study of the stability of reinforced soils is presented here.

RÉSUMÉ: À l’ère de la modélisation numérique complexe, la pratique de l’ingénierie géotechnique recourt toujours à des outils numériques plus simples, mettant en évidence les phénomènes clés. De tels logiciels sont importants pour l’enseignement puisqu’ils permettent de mieux comprendre les principes sous-tendant le dimensionnement des ouvrages géotechniques. Ils sont également importants pour l’expertise ou la recherche d’une part parce qu’ils sont faciles à utiliser et d’autre part parce qu’ils établissent un lien clair entre les hypothèses formulées et les résultats. La plupart de ces logiciels sont distribués sous licences propriétaires, générant un effet de « boîte noire » et offrant peu de flexibilité pour prendre en compte une situation particulière. D’un point de vue pédagogique, il est souvent utile de considérer des hypothèses aberrantes afin d’en montrer les conséquences. L'Université Gustave Eiffel développe ainsi un logiciel de calcul géotechnique, intitulé Slog. En libre accès et développé avec Python, langage très utilisé, ce logiciel a pour vocation de fédérer une communauté d'utilisateurs-contributeurs, notamment parmi les enseignants et étudiants géotechniques. Afin de pouvoir comparer facilement différentes structures potentielles sur un même site ou, à l'inverse, pouvoir évaluer la pertinence d'un ouvrage dans différents sites, l'utilisateur est amené à définir indépendamment le modèle de sol et les actions anthropiques. L'intersection entre les deux s'effectue automatiquement, permettant à l'utilisateur de modifier facilement et automatiquement les propriétés de l'un ou de l'autre. Ce processus flexible et évolutif est utile pour l’enseignement. L’exemple détaillé de l’étude de la stabilité des sols renforcés est présenté ici.

Keywords: Geotechnical software; open access; teaching modelling.

1 INTRODUCTION

Complex numerical modeling, specifically by finite elements, is today widely developed in geotechnical engineering, integrating the most complex constitutive laws and, when used correctly, produce predictions in accordance with field observations (El Arja, 2020). However, in engineering practice, geotechnical calculation still uses simplified tools exhibiting the key phenomena. As an example, the c-phi reduction method makes it possible to determine the safety factor of a slope but, in practice, most studies are carried out using traditional tools considering a bundle of potential failure surfaces and evaluating, for each of them, the ratio between motor and resistant efforts (Pilot, 1966).

Such software is important for teaching since it allows a better understanding of the principles underlying the design of structures. With some commercial finite element software students can have the impression that “you just have to press a button”. However, it would be simplistic to locate the sole interest of simplified digital tools there. They are also important for expertise or research on the one hand because they are easy to use and on the other hand because they establish a clear link between the calculation hypotheses retained and the results produced. The goal here is not to condemn this or that calculation method but to emphasize the usefulness for the geotechnical engineer of having access to all the calculation hypotheses and being able to modify them as he wishes, avoiding the "black box" effect. Open software provides the flexibility to take into consideration a particular situation. From an educational point of view, it also makes it possible to integrate aberrant assumptions into the calculation in order to show their consequences. Open software dedicated to geotechnical engineering can be found but they are generally devoted to one specific application. One can think for example about geotecha (https://github.com/rtrwalker/geotecha) for the consolidation of soft soils or PyGeo (https://github.com/CalvinNeo/PyGeo) for the recognition of structural surfaces.

Université Gustave Eiffel is therefore developing geotechnical calculation software, entitled Slog, freely distributed through its GitHub platform (https://github.com/Ifsttar/Slog). Developed using Python, a widely used programming language, this software aims to bring together a large community of user-contributors, particularly among geotechnical teachers and students. Accessible to the greatest number of people through the use of Python, Slog aims to respond to the difficulties of geotechnical engineers by taking into account the specificities of this domain.

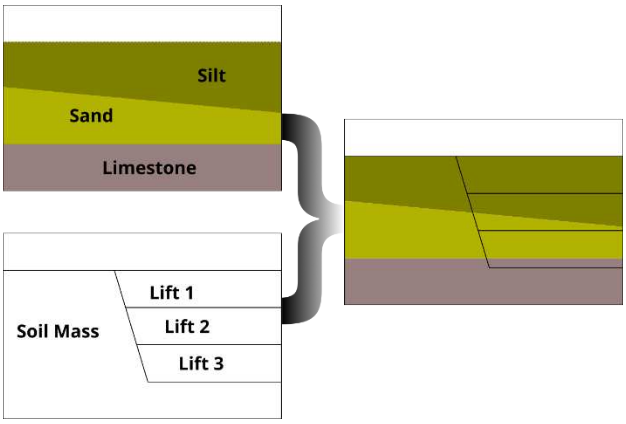

Figure 1. Domains considered for the study of a soil-nailed wall (top left: geology, bottom left: anthropogenic actions, right: intersection.

In order to be able to easily compare different potential geotechnical assets on the same site or, conversely, to be able to evaluate the relevance of an asset in different sites, the user is led to independently define a "geology" (partition of the space in blocks of soil) and an "asset" (set of anthropogenic actions). The intersection between "geology" and "asset" is made automatically, allowing the user to easily modify the properties of one or the other automatically. This flexible process proves useful on an educational level. A detailed example is presented here: the study of the stability of soil-nailed wall. This example is deliberately simplistic to make the screen output easier to read. In practice, a large number of slices must be considered and, depending on the method chosen (Fellenius, Bishop, perturbations, etc.) the geometry of the failure surface must be adapted.

2 DESCRIBING THE TERRAIN

One of the specificities of geotechnical engineering is to be placed in a merely known environment: the soil in place. This particularity leads to the definition of a terrain model, based on the study of geological maps, on the analysis of geotechnical surveys and laboratory tests as well as on extrapolations. During a project, the terrain model can therefore evolve according to new investigations and the revision of some assumptions. At the same time, the project itself may have to evolve between the different study phases. Where most software requires defining *ab initio* a domain divided into homogeneous zones, Slog suggests defining two distinct domains, one representing the terrain model (constituted of „Faciès“ objects) and the other representing the anthropogenic actions, *i.e.* the building project. The software itself generates a unique domain for the study by realizing the intersection of the two domains thus defined by the user.

As an example, let us consider a soil-nailed wall excavation project in a layered soil (See Figure 1). The user first defines a Geology consisting of a layer of Silt above a layer of Sand, itself above a layer of Limestone. Secondly, the user defines the domains linked to anthropogenic action: a "Soil Mass" domain corresponding to the part of the soil that the project plans to leave in place and three domains "Lift1", "Lift2" and "Lift3 » corresponding to successive excavation lifts. The software then generates by automatic intersection the division of the space usable for the calculations. The automatic intersection algorithm is taken from (Foster et al., 2019).

If additional tests carried out during the design study indicate that the sand layer actually has a higher modulus than the initial estimate, the user can simply modify the Geology and the terrain model automatically inherits the new properties. In the same way, if during the project, it is decided to review the dip of the sand/silt interface, this can be done on the Geology and the calculation of the intersection with the Anthropic Action immediately gives a new terrain model for the calculation. Note that, when constructing a Geology, Slog determines if all the facies constitute a partition of space, that is to say that they cover all the points of the calculation domain on the one hand and that they do not overlap each other. If this condition is not met, an error message is returned.

The Object Oriented programming implemented in Slog allows all the elements of a class to inherit the properties of this class (see Figure 2). In particular, the Geology facies like the anthropogenic geometry elements all inherit from the Polygon class. The latter are therefore the boundaries of the different geotechnical elements and the fact that they share properties facilitates the calculation of their intersection but above all allows them to be defined according to a similar procedure (list of vertices given in the trigonometric direction). This procedure facilitates interfacing between Slog and software such as GMSH (Geuzaine and Remacle, 2009), an open access mesher. Although the calculations carried out in Slog do not *a priori* require a mesh, the GMSH geometry module is very useful for generating geometries.

The Class Diagram presented in Figure 2 also allows to understand the overall structure of the terrain model. What has been presented so far actually constitutes a given configuration of the project. In particular, the division presented in Figure 1 assumes that 3 excavation lifts have been carried out but nothing prevents a subsequent configuration from including an additional lift or an embankment at the wall tip. A project is therefore a list of Configurations. Finally, for each Configuration, a set of structural elements (Geotextiles, nails, walls, etc.) are superimposed on the terrain which must also interface with the terrain model.

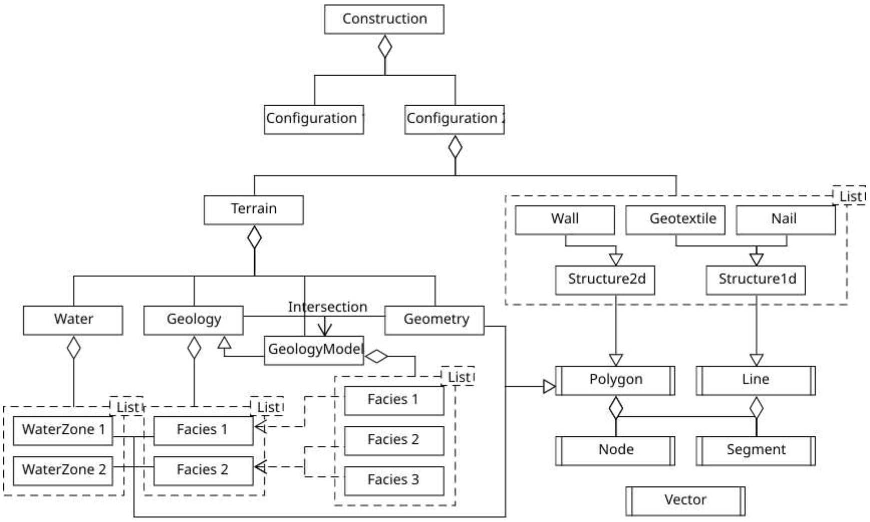

Figure 2. Slog structuring class diagram.

3 MANAGING STRUCTURAL ELEMENTS

Here we call structural elements the anthropogenic elements that can be taken into account in the calculation. We can thus think of nails, reinforcements or geotextiles in the case of reinforced soil support but more generally, these structural elements include diaphragm walls, sheet piles or even gravity walls. We could also think of injected soil elements but the strong proximity between “2d Structure” type objects and “Facies” type objects allows the user, depending on the needs of his calculation, to decide if he prefers represent jet grouting as a structural element or an element of Geology.

Note at this point that the current version of Slog is purely two-dimensional. A three-dimensional version is expected to be developed but has not yet been developed and, in engineering practice, many calculations are carried out in 2D. Thus, the truly linear elements (nails, anchors, etc.) and the surface elements (geotextiles, sheet piles, etc.) are both represented by linear elements. Unlike most geotechnical software, the structural elements being defined in parallel with the terrain model, they have no impact on the Geology and the Facies crossed remain continuous. In order to separate the upstream soil and the downstream soil of a sheet pile for example, it is necessary to translate this separation into “Anthropogenic actions”. Through the intersection of Geology and Anthropogenic Actions, this discontinuity will then be apparent in the domain study. From an educational point of view, this choice makes it possible to impose on the user a reflection on the nature of the structural elements that he introduces.

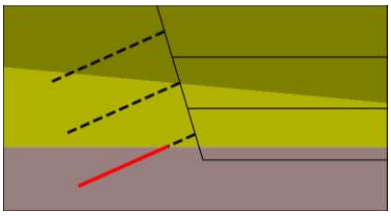

Figure 3. Addition of Nail objects and highlighting of intersection between the nails and the limestone (red).

In Figure 3 the example of the excavation presented above is completed by nails. The Polygon intersection function allows, in this case, to automatically determine the intersection between the nails and the Limestone layer. In practice, *nail3.intersect_polygon(limestone)* returns the list of intersections between nail 3 and the limestone layer. This list can contain several elements (if the limestone layer is concave) but it can also be empty (for example, if the nails were horizontal in our example). In any case, the following bit of code can be applied:

```
> for elem in clou3.intersect_polygon(limestone):
>       elem.ConstituveLaw = FZhao_limestone_grout
```

By having defined *FZhao_limestone_grout* as the law of Frank and Zhao (Frank and Zhao, 1982) with parameters corresponding to the interface between limestone and grout, these two lines of code make it possible to attribute the correct mechanical characteristics to this interface and without worrying about whether this interface actually exists. This robustness makes it possible to easily modify the geometry chosen for the project. Indeed, if we ultimately decide to consider horizontal nails, it is not necessary to redefine all of the interfaces.

4 CALCULATIONS METHOD

4.1 Choice of a failure surface and determination of safety factor

In order to determine the stability of the soil-nailed wall studied, we first choose a failure surface. On Figure 4, the example of logarithmic spirals is given but Slog includes also circles and slip lines. Currently, the definition of failure surfaces is not implemented in classes with dedicated methods and it only relies on functions defined in a separate modulus, called *slog_calculations.py*.

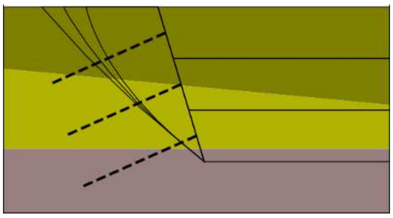

Figure 4. Computation of three logarithmic spiral failure surfaces.

In order to use the slice method, the failing block is subdivided into slices. This is enabled by the polygon intersection functions of Slog. Note that the slices thus created, inheriting from the Polygon class, are intrinsically equipped with the *self.area* method allowing the calculation of their surface and therefore their weight. On Figure 5, a slice intercepts different Facies and the calculation of its weight is easily done by subdividing it into the intersections with these different Facies.

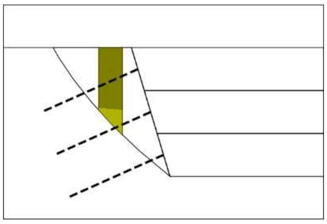

Figure 4. top left, choice of a rupture surface and slice zones; top right, automatic creation of the slices used for the stability calculation; bottom, intersection with nail.

In order to apply the slice method, it is essential to determine the resisting forces along the failure surface at the base of each slice. This requires the knowledge of the length of the slice basis as well as the maximal friction which can be mobilised on this basis. Here the basis is defined as the intersection between the failure line, the slice and the Facies *sand*. Therefore it inherits parameters from the Facies *sand* and, among them, the friction angle ϕ. This process is implemented in the function *sliceResistance* in the package *slog_calculations.slope_stability.*

In the case of nailing, specific stabilizing forces linked to the presence of nails must be added to the friction forces. To do this, it is necessary to determine the position of the intersection between the failure surface and the nails. This is, once again, done easily through the intersection method.

Once this point has been determined, the stabilizing force to be included in the slope stability calculation is computed using the intersection between the nail and the stable soil mass since the intersection inherits from its „parents“.

4.2 Modelling the nails

One way to determine these stabilizing forces is to integrate the equation of the beams along the nail taking into account the movement of the failing block and the friction at the interface between the nail and the different Facies it crosses (Delmas et al, 1986). To do this, it is necessary to implement a finite difference type calculation along this nail, which must therefore be discretized. The discretization of the nail is easily done using the `linspace` tool from the `numpy` library. As mentioned in section 3, Nail type objects have a method allowing them to define their interface with Facies. Each element of the discretized nail can therefore be assigned to a given interface and thus be provided with a friction model (e.g. Frank and Zhao law).

4.3 Encapsulation of calculation methods

Currently, the only calculation available in Slog is the slope stability (with or without nails) and it is not implemented in an Object Oriented way. Before including new calculations, the developers focus on the transition from functions packages to method libraries. However, the example of soil-nailed walls hereby presented aims at illustrating the powerfulness of the Object Oriented description of the Geology and the Geometry.

5 CONCLUSIONS

The Slog software, built in the form of a Python library offers a structure that is both robust and easy to use, allowing on the one hand the use of existing geometric libraries and on the other hand the creation of numerous calculation models. Development under a free license also allows the free use of the code by students, teachers and professionals alike, allowing everyone to have access to the entire calculation code and therefore to understand and question the assumptions underlying the results obtained. However, the „black box“ effect of geotechnical softwares can only be removed if the open source code is accompanied by a clear theroritical manuel. A significant effort has still to be carried out regarding this point.

Finally, open development should allow the creation of various modules but also the federation of a community of user-contributors around tools such as a forum. Apart from the educational interest of this software, Slog can then become a medium for disseminating research results. One of the main specificities of geotechnical calculation is based on the definition of the terrain model and close attention will be paid to this by developers.

ACKNOWLEDGEMENTS

The development of Slog was made possible by the work of Jad Sadek, as part of a mission from the Junior-Enterprise X-Projets. Through his availability, his alertness and his rigor, he made it possible to set the heart of the terrain model system.

REFERENCES

Delmas, P., Cartier, G. and Abdelhedi, A. (1986) ‘Une nouvelle méthode de dimensionnement du clouage des pentes: programme Prosper’, *Bulletin de liaison des Laboratoires des Ponts et Chaussées*, (141).

El Arja, H. (2020) Contribution à la modélisation numérique des excavations profondes. Université Paris Est. PhD thesis

Foster, E.L., Hormann, K. and Popa, R.T. (2019) ‘Clipping simple polygons with degenerate intersections’, Computers & Graphics: X, 2, p. 100007. https://doi.org/10.1016/j.cagx.2019.100007.

Frank, R. and Zhao, S.R. (1982) ‘Estimation par les paramètres pressiométriques de l’enfoncement sous charge axiale de pieux forés dans des sols fins’, Bulletin de liaison des Laboratoires des Ponts et Chaussées, (119).

Geuzaine, C. and Remacle, J.-F. (2009) ‘Gmsh : a three-dimensionnal finite element mesh generator with built-in pre- and post-processing facilities.’, International Journal for Numerical Methods in Engineering, 79(11), 1309–1331. https://doi.org/10.1002/nme.2579

Pilot, G. (1966) ‘Calcul de la stabilité des pentes. Utilisation de l’ordinateur CAE 510 du laboratoire central programmes 1.1 et 1.2’, Bulletin de liaison des Laboratoires des Ponts et Chaussées, 22, 4–1.