



HAL
open science

Slog : un logiciel libre de calcul géotechnique

Jean de Sauvage

► **To cite this version:**

Jean de Sauvage. Slog : un logiciel libre de calcul géotechnique. 11èmes journées nationales de géotechnique et de géologie de l'ingénieur, Institut National des Sciences Appliquées de Lyon [INSA Lyon], CFMS, CFMR, CFGI, Jun 2022, Lyon, France. <hal-03719954>

HAL Id: hal-03719954

<https://hal.science/hal-03719954v1>

Submitted on 11 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

SLOG : UN LOGICIEL LIBRE DE CALCUL GEOTECHNIQUE

SLOG : A FREE SOFTWARE FOR GEOTECHNICAL COMPUTATIONS

Jean de SAUVAGE¹

¹ GERS-SRO, Univ Gustave Eiffel, IFSTTAR, Marne-la-Vallée, F-77454, France

RÉSUMÉ – L'Université Gustave Eiffel développe un logiciel de calcul géotechnique, intitulé Slog, diffusé en accès libre. Développé sous Python, ce logiciel vise à fédérer une communauté d'utilisateurs-contributeurs géotechniciens. Les principales spécificités sont donc la considération de l'interaction entre géologie et actions anthropiques et la définition du modèle de terrain. Un calcul de type stabilité de pente est présenté en exemple.

ABSTRACT – The Gustave Eiffel University is developing an open-access geotechnical computation software, called Slog. Developed with Python, it aims to gather a community of geotechnical user-contributors. Thus, the main specificities are the taking into account of the interaction between geology and anthropogenic actions and the definition of the terrain model. A slope stability type calculation is presented as an example.

1. Introduction

Les modélisations numériques complexes, notamment par éléments finis, sont aujourd'hui largement développées en géotechniques, intégrant des modèles de comportement des plus complexes et, lorsqu'elles sont bien utilisées, produisent des prédictions très comparables aux observations de terrain (El Arja, 2020). Toutefois, dans la pratique de l'ingénieur, le calcul géotechnique recourt toujours à des outils numériques simplifiés, exhibant les phénomènes clés. A titre d'exemple, les méthodes de *c-phi* réduction permettent de déterminer le facteur de stabilité d'une pente mais, dans la pratique, la plupart des études sont menées à l'aide d'outils traditionnels considérant un faisceau de surfaces de ruptures potentielles et évaluant, pour chacune d'elles, le rapport des efforts moteurs et résistants (Pilot, 1966).

De tels logiciels sont importants pour l'enseignement puisqu'ils permettent de mieux appréhender les principes sous-tendant le dimensionnement d'ouvrages. On peut ici penser à l'effet de certains logiciels commerciaux de calcul aux éléments finis dont l'ergonomie donne aux étudiants l'impression « qu'il suffit d'appuyer sur un bouton ». Il serait toutefois réducteur de situer là le seul intérêt des outils numériques simplifiés. Ils sont également importants pour l'expertise ou la recherche d'une part parce qu'ils sont simples d'usage et d'autre part parce qu'ils établissent un lien clair entre les hypothèses de calcul retenues et ses résultats produits.

L'auteur de cette communication ne souhaite pas jeter l'anathème sur telle ou telle méthode de calcul mais bien souligner l'utilité pour le géotechnicien d'avoir accès à l'ensemble des hypothèses de calcul et de pouvoir les modifier à sa guise, évitant l'effet « boîte noire ». Un logiciel ouvert offre la souplesse nécessaire pour prendre en considération une situation particulière. Du point de vue pédagogique, il permet également d'intégrer dans le calcul des hypothèses aberrantes afin d'en montrer les conséquences.

L'Université Gustave Eiffel développe donc un logiciel de calcul géotechnique, intitulé Slog, diffusé librement (sous licence CeCILL-C) au travers de sa plateforme GitHub (<https://github.com/lfsttar/Slog>). Développé sous Python, langage de programmation très répandu, ce logiciel vise à fédérer une large communauté d'utilisateurs-contributeurs, en particulier parmi les enseignants et élèves en géotechnique. Accessible au plus grand nombre par l'usage de Python, Slog ambitionne

de répondre aux difficultés des géotechniciens en prenant en compte les spécificités de ce domaine

Afin de pouvoir comparer aisément différents ouvrages potentiels sur un même site ou, à l'inverse, de pouvoir évaluer la pertinence d'un ouvrage dans différents sites, l'utilisateur est amené à définir indépendamment une « géologie » (partition de l'espace en blocs de sol) et un « ouvrage » (ensemble des actions anthropiques). L'intersection entre « géologie » et « ouvrage » est réalisée automatiquement, permettant à l'utilisateur de modifier aisément les propriétés de l'un ou de l'autre automatiquement. Ce procédé souple et évolutif s'avère utile sur le plan pédagogique. L'exemple détaillé de l'étude de la stabilité d'un déblai réalisé en sol cloué est ici présenté. Cet exemple est délibérément simpliste pour faciliter la lecture des sorties d'écrans. Dans la pratique, un nombre important de tranches doit être considéré et, selon la méthode retenue (Fellenius, Bishop, perturbations...) la géométrie de la surface de rupture doit être adaptée.

2. Création du modèle de terrain

L'une des spécificités du travail de géotechnicien est de se placer dans un milieu qu'il ne maîtrise pas : le sol en place. Cette particularité conduit à la définition d'un modèle de terrain, basé sur l'étude des cartes géologiques, sur l'analyse des sondages géotechniques et des essais de laboratoires ainsi que sur des extrapolations. Au cours d'un projet, le modèle de terrain peut donc évoluer au gré des nouvelles investigations et de la révision de certaines hypothèses. Dans le même temps, le projet en lui-même peut être amené à évoluer entre la G1 (étude préalable) et la G3 (étude d'exécution).

Là où la plupart des logiciels demandent de définir *ab initio* un domaine d'étude découpés en zones homogènes, Slog propose de définir deux domaines distincts, l'un représentant le modèle de terrain et l'autre représentant les actions anthropiques, c'est-à-dire le projet de l'acte de construire. Le logiciel génère lui-même un domaine d'étude unique en réalisant l'intersection des deux domaines ainsi définis par l'utilisateur.

A titre d'exemple, considérons un projet de déblai réalisé par clouage de sol dans un sol stratifié (Voir Figure 1). L'utilisateur définit dans un premier temps une *Géologie* constituée d'une couche de Limon au-dessus d'une couche de Sable, elle-même au-dessus d'une couche de Calcaire. Dans un second temps, l'utilisateur définit les domaines liés à l'action anthropique : un domaine « Massif » correspondant à la partie du sol que le projet prévoit de laisser en place et trois domaines « Passe1 », « Passe2 » et « Passe3 » correspondant aux passes successives d'excavation. Le logiciel génère alors par intersection automatique le découpage de l'espace utilisable pour les calculs. L'algorithme d'intersection automatique est tiré de Foster *et al.* (2019).

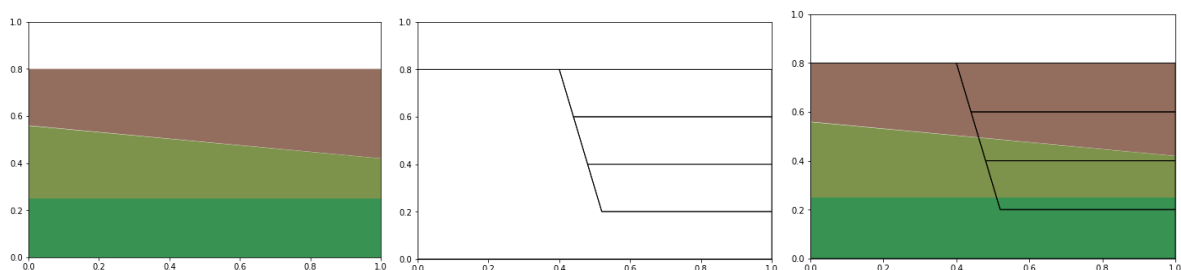


Figure 1 : Différents domaines considérés pour l'étude d'un mur cloué (gauche : géologie, milieu : action anthropique, droite : intersection des deux). De haut en bas, les couches géologiques sont Limon, Sable, Calcaire.

Si des essais complémentaires réalisés en G2 (étude de conception) indiquent que la couche de sable possède en réalité un module Ménard deux fois plus élevé que l'estimation initiale, il suffit de modifier la *Géologie* et le modèle de terrain hérite

automatiquement des nouvelles propriétés. De la même manière, si au cours du projet, on décide de revoir le pendage de l'interface sable/limon, cela peut-être fait sur la *Géologie* et le calcul de l'intersection avec l'*Action Anthropique* donne immédiatement un nouveau modèle de terrain pour le calcul. Notons que, lors de la construction d'une *Géologie*, Slog détermine si l'ensemble des *faciès* constitue une partition de l'espace, c'est-à-dire qu'ils recouvrent tous les points du domaine de calcul d'une part et qu'ils ne se recouvrent pas l'un l'autre. Si cette condition n'est pas respectée, un message d'erreur est renvoyé.

Notons à ce stade que la programmation Orientée Objet mise en œuvre dans Slog permet à l'ensemble des éléments d'une classe d'hériter des propriétés de cette classe (Voir Figure 2). En particulier, les *faciès* de la *Géologie* comme les éléments de géométrie anthropiques héritent tous de la classe *Polygon*. Ces derniers sont donc les frontières des différents éléments géotechniques et le fait qu'ils partagent des propriétés facilite le calcul de leur intersection mais permet surtout de tous les définir selon une procédure similaire (liste de sommets donnés dans le sens trigonométrique). Cette procédure facilite l'interfaçage entre Slog et des logiciels tels que GMSH (Geuzaine et Remacle, 2009), un mailleur libre d'accès. Bien que les calculs effectués dans Slog ne nécessitent a priori pas de maillage, le module géométrique de GMSH est très utile pour générer des géométries.

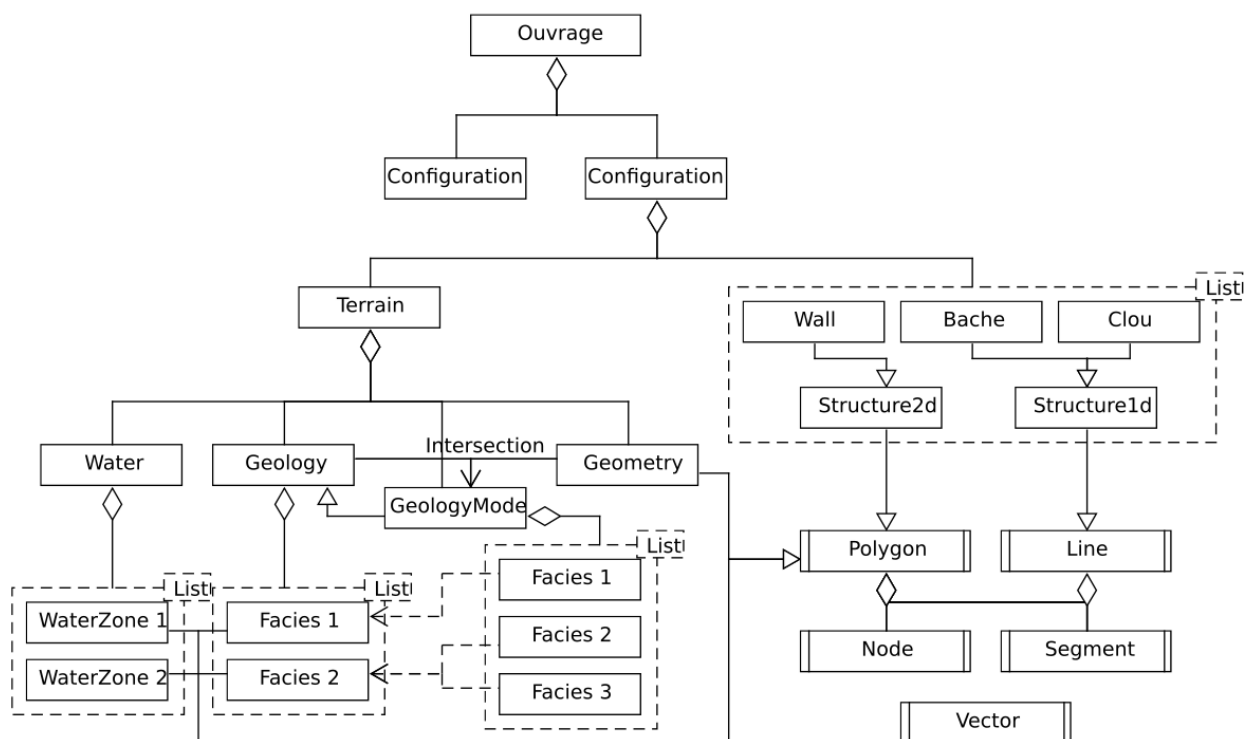


Figure 2 : Diagramme des classes structurant Slog

Le Diagramme des classes présenté ci-dessus permet également de saisir la structure globale du modèle de terrain. Ce qui a été présenté jusque-là constitue en réalité une configuration donnée du projet. En particulier, le découpage présenté en Figure 1 suppose que 3 passes d'excavation ont été réalisées mais rien n'interdit une configuration ultérieure de comporter une passe supplémentaire ou un remblai en tête de mur. Un projet est donc une liste de Configurations.

Enfin, pour chaque Configurations, se superposent au terrain un ensemble de structures (Géotextiles, clous, murs, ...) qui devront également s'interfaçer avec le modèle de terrain.

3. Gestion des éléments de structure

On appelle ici élément de structure les éléments apportés par l'homme dont la prise en compte est nécessaire au calcul. On peut ainsi penser à des clous, des armatures ou des géotextiles dans le cas d'un soutènement en sol renforcé mais plus généralement, ces éléments de structure incluent paroi moulée, palplanche ou même mur poids. On pourrait aussi penser à des éléments de sol injecté mais la forte proximité entre les objets de type « Structure 2d » et les objets de type « Faciès » permettent à l'utilisateur, en fonction des besoins de son calcul, de décider s'il préfère représenter le *jet grouting* comme un élément de structure ou un élément de la *Géologie*.

Notons à ce stade que la version actuelle de Slog est purement bidimensionnelle. Une version tridimensionnelle est appelée à être développée mais elle ne l'est pas encore et, dans la pratique de l'ingénieur, beaucoup de calculs sont réalisés en 2D. Ainsi, les éléments réellement linéiques (clous, ancrages, ...) et les éléments réellement surfaciques (géotextiles, palplanches, ...) sont tous deux représentés par des éléments linéiques. A la différence de la plupart des logiciels de géotechnique, les éléments de structure étant définis en parallèle du modèle de terrain, ils n'ont pas d'impact sur la *Géologie* et les *Faciès* traversés restent continus. Afin de séparer le sol amont et le sol aval d'une palplanche par exemple, il est nécessaire de traduire cette séparation dans les «*Actions anthropiques* ». Par l'intersection de la *Géologie* et des *Actions anthropiques*, cette discontinuité sera alors apparente dans le domaine d'étude. Du point de vue pédagogique, ce choix permet d'imposer à l'utilisateur une réflexion sur la nature des éléments de structure qu'il introduit.

Rappelons enfin que les fonctionnalités de Slog sont appelées à s'étoffer avec les contributions des utilisateurs. Différents objets héritant de la classe *Structure 1D* peuvent donc être définis avec des méthodes propres. Ainsi l'objet *Clou* comprend naturellement une méthode d'intersection avec les *Faciès* afin de définir des paramètres de frottement tandis qu'un objet *Ancrage* n'en aurait pas nécessairement besoin sur sa longueur libre.

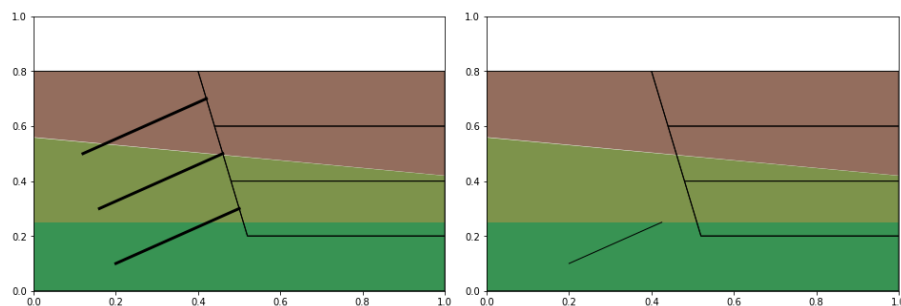


Figure 3 : Ajout d'objets de type *Clou* (à gauche : ensemble des clous, à droite : intersection entre les clous et la couche de calcaire)

Sur la Figure 3 l'exemple du clouage présenté plus haut est complété par des clous. La fonction d'intersection des *Polygon* permet, dans ce cas, de déterminer automatiquement l'intersection entre les clous et la couche de Calcaire.

En pratique, `clou3.intersect_polygon(calcaire)` renvoie la liste des intersections entre le clou 3 et la couche de calcaire. Cette liste peut contenir plusieurs éléments (si la couche de calcaire est concave) mais elle peut aussi être vide (par exemple, si les clous étaient horizontaux dans notre exemple). Dans tous les cas, le morceau de code suivant est applicable :

```
> for elem in clou3.intersect_polygon(calcaire) :
>     elem.ConstituteLaw = FZhao_calc_coulis
```

En ayant défini au préalable *FZhao_calc_coulis* comme la loi de Frank et Zhao (Frank et Zhao, 1982) avec des paramètres correspondants à l'interface entre calcaire et coulis, ces deux lignes de codes permettent d'attribuer les bonnes caractéristiques mécaniques à cette interface et sans se préoccuper de savoir si cette interface existe réellement. Cette robustesse permet de modifier aisément la géométrie retenue pour le projet. En effet, si l'on décide finalement de considérer des clous horizontaux, il n'est pas nécessaire de redéfinir l'ensemble des interfaces.

4. Méthodes de calcul

Comme évoqué plus haut, la principale vocation de Slog est de proposer un cadre global suffisamment souple et adapté pour réaliser l'ensemble des calculs géotechniques. Les modules développés initialement proposent des calculs numériques simplifiés issus des logiciels développés par le LCPC (Delmas *et al.*, 1986 ; Frank *et al.*, 1990) mais la définition du modèle de terrain étant compatible avec un maillage de type GMSH, il est tout à fait possible d'envisager des modules de calcul par éléments finis si l'utilisateur le souhaite.

L'usage de Python permet d'accéder à une grande famille de bibliothèques prédéfinies et donc de faciliter l'implémentation d'un nouveau type de calcul. Les éléments de type *Node*, constituant les sommets des éléments de type *Polygon* sont dotés de la méthode *self.in_polygon(poly)* qui permet de déterminer si le *Node* appartient au *Polygon* nommé *poly*.

Choix d'une surface de rupture et calcul de stabilité

Afin de déterminer la stabilité du mur cloué étudié, on choisit dans un premier temps une surface de rupture et, afin d'utiliser la méthode des tranches, on subdivise le bloc en rupture en tranches (voir Figure 4). La définition de tranches d'épaisseur constante est réalisable à l'aide de la fonctionnalité *linspace* de la bibliothèque *numpy*. La création des tranches utilisables pour le calcul est ensuite permise par les fonctions d'intersection de polygones de Slog. On notera que les tranches ainsi créées, héritant de la classe *Polygon* sont intrinsèquement dotée de la méthode *self.area* permettant le calcul de leur surface. Connaissant le poids volumiques du sol, le poids de la tranche est donc directement calculable. Enfin, la troisième tranche interceptant différents *Facies*, le calcul de son poids se fera en la subdivisant en ces intersections avec les différents *Facies*.

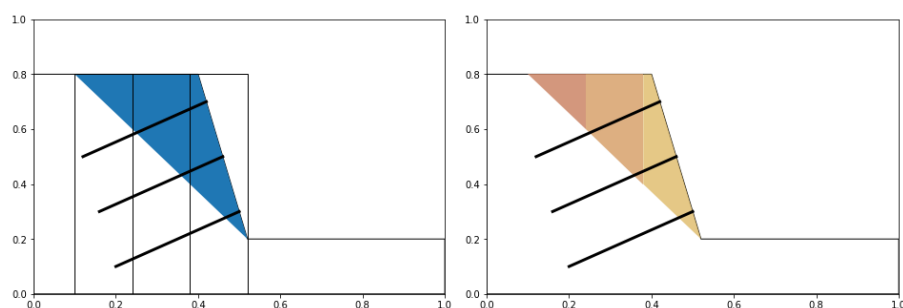


Figure 4 : à gauche, choix d'une surface de rupture et de zones tranches ; à droite, création automatique des tranches utilisées pour le calcul de stabilité.

Afin d'appliquer la méthode des tranches, il est essentiel de déterminer les efforts résistants le long de la surface de rupture à la base de chaque tranche. Dans notre cas, la surface de rupture est représentée par l'objet *rupture* de type *Line*. La Figure 5 présente la détermination des efforts résistants à la base de la tranche 2, représenté par l'objet *rupT2*, de type *Polygon*. La base de la Tranche, *IRup2* est obtenue par :

```
> IRup2 = rupture.intersect_polygon(rupT2)[0]
```

IRup2 est représenté sur la Figure 5 par un trait noir. Il s'agit d'un objet de type *Line*. En réalisant son intersection avec tous les *Faciès*, on peut déterminer les paramètres de frottement à prendre en considération. Dans le cas du clouage, il faut ajouter aux forces de frottement des efforts stabilisateurs ponctuels liés à la présence des clous. Pour cela, il est nécessaire de déterminer la position de l'intersection entre la surface de rupture et les clous.

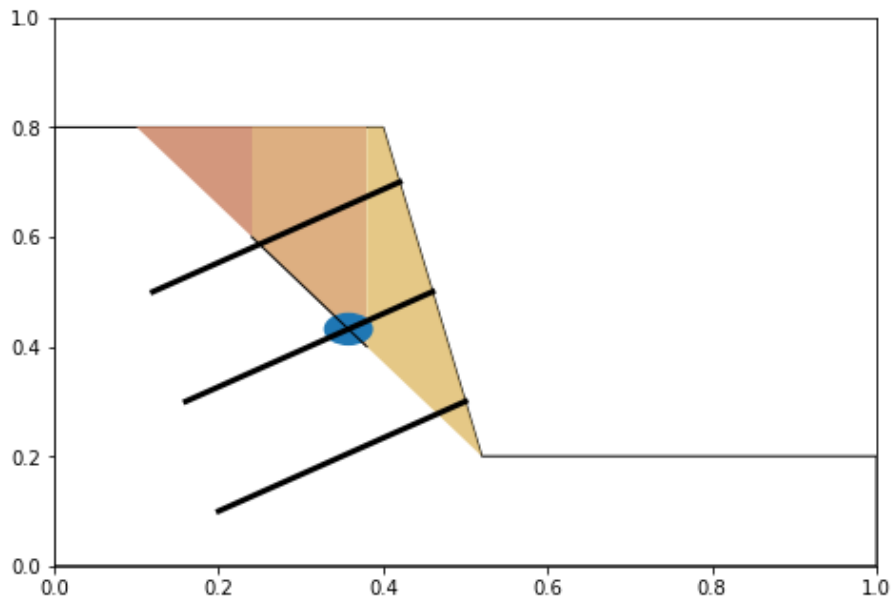


Figure 5 : Détermination des efforts résistants s'exerçant sur la base de la tranche 2.
Représentation de *IRup2* et *effPonct2*

La figure 5 représente par un disque bleu le point d'intersection entre *IRup2* et le clou 2, point que nous appellerons *effPonct2*. Ce dernier est déterminé par :

```
> effPonct2 = IRup2.segments[0].intersect(clou2.segments[0])
```

Une fois ce point déterminé, il est possible de calculer l'effort stabilisateur à intégrer dans le calcul de stabilité de pente en fonction de la position d'*effPonct2* dans le clou.

Modélisation des clous

Une façon de déterminer ces efforts stabilisateurs est d'intégrer l'équation des poutres le long du clou en tenant compte du mouvement du bloc en rupture et du frottement à l'interface entre le clou et les différents *Faciès* qu'il traverse (Delmas *et al.*, 1986). Pour ce faire, il faut implémenter un calcul de type différences finies le long de ce clou, qu'il faut donc discrétiser. La discrétisation du clou se fait aisément à l'aide de l'outil *linspace* de la bibliothèque *numpy*.

Comme évoqué en section 3, les objets de type *Clou* sont dotés d'une méthode permettant de définir leur interface avec les *Faciès*. Chaque élément du clou discrétisé peut donc être affecté à une interface donnée et ainsi être doté d'un modèle de frottement (ex : loi de Frank et Zhao). Enfin, les coordonnées du point *effPonct2* étant connues, l'effort normal et l'effort tranchant dans le clou à ce point sont calculés et réinjectés dans le calcul de stabilité par la méthode des tranches.

5. Compléments sur la création du modèle de terrain

Si la structure du code de Slog et l'utilisation de Python permettent de simplifier le développement de nombreux modules de calcul, il reste essentiel de pouvoir disposer

d'une manière adaptée de définir un modèle de terrain. Dans la pratique, la définition de ce modèle de terrain n'est pas univoque et la résolution des conflits entre sources d'informations n'est pas toujours aisée. Du point de vue pédagogique, il sera donc important de développer différents modules de définition de terrain. Cela sera permis par la structure générique du code. On peut noter ici la bibliothèque Python *GemPy* (Varga *et al.*, 2019) capable de représenter de modèles géologiques complexes sous divers formats. Un interfaçage entre *GemPy* et *Slog* est rendu possible par l'utilisation des mêmes bibliothèques graphiques. Toutefois, les modules actuellement présents sous *Slog* sont en 2D et l'interfaçage avec *GemPy* ne sera pertinent que lorsque les modules 3D seront développés.

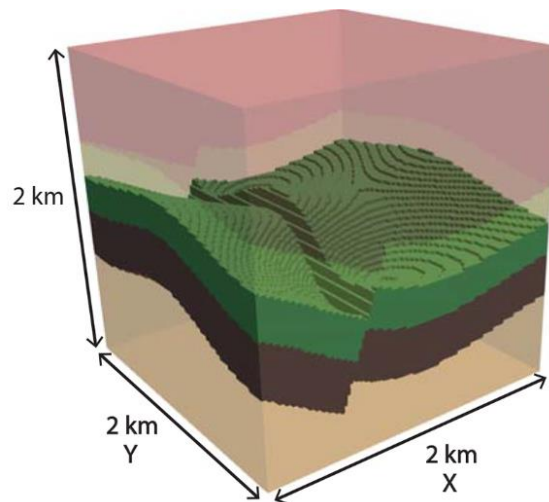


Figure 6 : Visualisation sous Paraview d'une géologie créée sous GemPy (Varga *et al.*, 2019)

6. Conclusions

La suite logicielle *Slog*, construite sous la forme d'une bibliothèque Python dispose actuellement d'un nombre très limité de modules mais elle propose une structure à la fois robuste et simple d'utilisation permettant d'une part l'utilisation de bibliothèques géométriques existantes et d'autre part la création de nombreux modèles de calculs.

Le développement sous licence CeCILL-C permet de plus la libre utilisation du code tant par les étudiants et les enseignants que les professionnels, permettant à chacun d'avoir accès à la totalité du code de calcul et donc de comprendre et remettre en question les hypothèses sous-tendant les résultats obtenus.

Enfin, le développement libre devra permettre la création de modules variés mais aussi la fédération d'une communauté d'utilisateurs-contributeurs autour d'outils tels qu'un forum. En dehors de l'intérêt pédagogique de cette suite logicielle, *Slog* peut alors devenir un support de diffusion de résultats de recherches.

Une des principales spécificités du calcul géotechnique repose sur la définition du modèle de terrain et une forte attention y sera portée de la part des développeurs.

7. Remerciements

Le développement de *Slog* a été rendu possible par le travail de Jad Sadek, dans le cadre d'une mission de la Junior-Entreprise X-Projets. Par sa disponibilité, sa vivacité d'esprit et sa rigueur, il a permis de mettre en place le cœur du dispositif de création de modèle de terrain.

8. Références bibliographiques

- Delmas, P., Cartier, G. et Abdelhedi, A. (1986). Une nouvelle méthode de dimensionnement du clouage des pentes: programme Prosper. *Bulletin de liaison des Laboratoires des Ponts et Chaussées* 141
- Dezert, T., Fargier, Y., Lopes, S.P. et Côte P. (2019) Geophysical and geotechnical methods for fluvial levee investigation: A review. *Engineering Geology*. 3;260:105206.
- El Arja, H. (2020). Contribution à la modélisation numérique des excavations profondes. *Thèse de doctorat*, Université Paris-Est.
- Foster, E.L., Hormann, K. et Popa, R.T. (2019). Clipping simple polygons with degenerate intersections. *Computers & Graphics: X. 2019 Dec* 1;2:100007.
- Frank, R., Bangratz, J.-L. et Kutniak, M. (1990). PILATE-LCPC: Programme de calcul d'un pieu isolé soumis à des efforts de flexion en tête et à des poussées latérales de sol. *Bulletin de liaison des Laboratoires des Ponts et Chaussées*
- Frank, R. et 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. et 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, pp 1309-1331.
- 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
- Varga, M.D., Schaaf, A., Wellmann, F. (2019). GemPy 1.0: open-source stochastic geological modeling and inversion. *Geoscientific Model Development*, 12(1):1-32.