Multi-slicing strategy for the three-dimensional discontinuity layout optimization (3D DLO)

Yiming Zhang*†

Material-Technology Innsbruck (MTI), University of Innsbruck, Technikerstraße 13, 6020 Innsbruck, Austria

SUMMARY

Discontinuity layout optimization (DLO) is a recently presented topology optimization method for determining the critical layout of discontinuities and the associated upper bound limit load for plane two-dimensional and three-dimensional (3D) problems. The modelling process (pre-processing) for DLO includes defining the discontinuities inside a specified domain and building the target function and the global constraint matrix for the optimization solver, which has great influence on the efficiency of the computation processes and the reliability of the final results. This paper focuses on efficient and reliable pre-processing of the discontinuities within the 3D DLO and presents a multi-slicing strategy, which naturally avoids the overlapping and crossing of different discontinuities. Furthermore, the formulation of the 3D discontinuity considering a shape of an arbitrary convex polygon is introduced, permitting the efficient assembly of the global constraint matrix. The proposed method eliminates unnecessary discontinuities in 3D DLO, making it possible to apply 3D DLO for solving large-scale engineering problems such as those involving landslides. Numerical examples including a footing test, a 3D landslide and a punch indentation are considered, illustrating the effectiveness of the presented method. © 2016 The Authors. International Journal for Numerical and Analytical Methods in Geomechanics published by John Wiley & Sons Ltd.

1. INTRODUCTION

Topology optimization is a type of mathematical approach for searching for the optimum material layout considering a given space. In engineering fields, topology optimization is commonly used for obtaining the best design considering one or several given target variables such as weight/cost and a series of constraints such as specific loads and supports [1]. With the progress of computing power in the last few decades, presently, it is possible to efficiently analyse large-scale problems with millions of variables on normal personal computers [2–4].

Belonging to the family of topology optimization, discontinuity layout optimization (DLO) is a recently presented, elegant and promising numerical method for determining (i) the critical layout of discontinuities in a body at failure and (ii) the associated upper bound limit load [5] for plasticity problems [6–9]. Introducing large numbers of potential discontinuities permitted to crossover one another in a given domain and specifying the loading and boundary conditions (live loads, dead loads and supports), DLO can automatically generate a set of activated discontinuities corresponding to the upper bound limit load. This method has been successfully used to determine the slip-line mechanisms of several geotechnical problems [7, 10, 11], the yield-line failure of slabs [12, 13], failure mechanisms of masonry structures [14] and of retaining walls in seismic conditions [15].

*Correspondence to: Yiming Zhang, Material-Technology Innsbruck (MTI), University of Innsbruck, Technikerstraße 13, 6020 Innsbruck, Austria.
†E-mail: Yiming.Zhang@uibk.ac.at

The copyright for this article was changed on 3 October 2016 after original online publication.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2016 The Authors. International Journal for Numerical and Analytical Methods in Geomechanics published by John Wiley & Sons Ltd.
In two-dimensional (2D) DLO, the discontinuities are geometrically represented by line segments, simplifying the pre-processing. Introducing large numbers of grid points into a specific domain, the potential discontinuities are built by connecting each pair of two grid points. In three-dimensional (3D) DLO, on the other hand, the discontinuities are geometrically represented by polygons. Overlapping and crossing of different 3D discontinuities should be avoided to save computing resources [8, 9], as an important pre-processing task. For the potential wide application of 3D DLO, a highly efficient pre-processing method is necessary.

In this paper, the author presents a multi-slicing strategy for the pre-processing of discontinuities for 3D DLO, naturally avoiding the aforementioned overlap and crossing of different 3D discontinuities. Furthermore, a 3D discontinuity formulation with the shape of an arbitrary convex polygon is introduced, assuring the efficient assembly of the global constraint matrix. Computing effort during pre-processing is saved while the number of unnecessary discontinuities is greatly reduced, paving the way for the analysis of large-scale engineering problems such as landslides with 3D DLO.

This paper is organized as follows: in Section 2, DLO formulations for 2D and 3D problems, including the standard pre-processing steps, will be presented, highlighting the necessity for a new method for pre-processing of 3D DLO problems. The multi-slicing strategy and the formulation of 3D discontinuity considering a shape of an arbitrary convex polygon will be presented in Section 3. In Section 4, a footing test, a 3D landslide problem and a punch indentation problem will be investigated, proving the effectiveness of the presented method. This paper closes with concluding remarks given in Section 5.

2. BASICS OF DISCONTINUITY LAYOUT OPTIMIZATION

From this section on, the author denotes the local variable (vector, matrix) with a subscript, for example $d_i$ represents the displacement jump of the discontinuity “$i$”. Meanwhile the symbol without subscript is used to represent a global variable (vector, matrix) such as $d = [d_1, \ldots, d_m]^T$ being the displacement jump of all the discontinuities in the domain. Furthermore, for better understanding, the dimensions of the equations are provided in the form of $L$ for length and $F$ for force (mass-length-time$^{-2}$).

2.1. Kinematics

The detailed formulation of the DLO was first presented in [7]. For a domain with multiple potential discontinuities shown in Figure 1, several discontinuities become activated (i.e. displacement jump $\neq 0$) during loading. When assuming rigid body displacements of the subdomains separated by the activated discontinuities, based on energy conservation, the following relationship is obtained:

$$W(l) + W(d) - E = 0, \quad (F \cdot L)$$

where $W(l)$ is the work performed by the live load, $W(d)$ is the work performed by the dead load such as self weight and $E$ is the energy dissipated at the discontinuities.

When minimizing the live load $W(l)$ in the course of limit analysis, the target function of the optimization problem is obtained as

$$W(l) = [E - W(d')] \rightarrow \text{min.} \quad (F \cdot L)$$

In case several discontinuities are connected to one node, because of the rigid displacement of the subdomains, compatibility relationships are formulated at this node. For example, for the three

![Figure 1. Potential discontinuities become activated at failure. [Colour figure can be viewed at wileyonlinelibrary.com]](image)
2D discontinuities \( i, j \) and \( k \) connected to the node \( P \), with corresponding displacement jump\(^\ddagger\) in the normal and shear directions\(^\S\) \( \mathbf{d}_i = [s_i, n_i]^T \), \( \mathbf{d}_j = [s_j, n_j]^T \), and \( \mathbf{d}_k = [s_k, n_k]^T \) (Figure 2), following relationship is obtained:

\[
\mathbf{X}_P = \begin{bmatrix} \mathbf{B}_{P,i} & \mathbf{B}_{P,j} & \mathbf{B}_{P,k} \end{bmatrix} \begin{bmatrix} \mathbf{d}_i \\ \mathbf{d}_j \\ \mathbf{d}_k \end{bmatrix} = \mathbf{0}, \quad (L)
\]

where \( \mathbf{X}_P \) is the relative displacement of node \( P \), \( (\mathbf{B}_{P,i} \mathbf{d}_i) \) is the contribution of discontinuity \( i \) to the relative displacement of node \( P \).\(^3\) \( \mathbf{B}_{P,i} \) is a coordinate transformation matrix (\( 2 \times 2 \) for 2D, see [7] for details and \( 3 \times 3 \) for 3D, see Appendix A for details) transforming the local displacement jump \( \mathbf{d}_i \) (given in local coordinates) to the global coordinate system \([7, 8]\). In 3D DLO, this compatibility relationship is formulated for every edge \([8, 9]\), such as edge \( PQ \) illustrated in Figure 3.

The compatibility relationships are built at every node (for 2D DLO) or edge (for 3D DLO) in the whole domain. A global compatibility constraint is obtained as

\(^\ddagger\)The author would like to emphasize that \( \mathbf{d}_i \) should not be considered as 'displacements of the discontinuity \( i \)'\(^1\). \( \mathbf{d}_i \) is the 'displacement jump' between the two subdomains separated by the discontinuity \( i \).

\(^\S\)Rotational failure mechanisms are not considered in this paper, the formulation of which was given in [10].

\(^3\)For better understanding of Equation (3), the author suggests reverse process considering Figure 2(a)–(b). Because \( P_1 \), \( P_2 \) and \( P_3 \) in Figure 2(b) come into the same position as \( P \) in Figure 2(a), the overall relative displacement of the node \( P \) contributed from the discontinuities \( i, j \) and \( k \) is equal to zero.
There are two types of discontinuities in the domain as (i) boundary discontinuity and (ii) inner discontinuity (Figure 4). Considering a discontinuity $j$ with displacement jump as $d_j \in [s_j, n_j]^T$ for 2D or $d_j \in [s_j, t_j, n_j]^T$ for 3D, if $j$ is a boundary discontinuity, $d_j$ equals to the absolute displacement. The activations of the boundary discontinuities do not contribute to the dissipated energy $E$ (Figure 1).

If $j$ is an inner discontinuity, on the other hand, $d_j \neq 0$ indicates the sliding and opening of the subdomains separated by $j$. Considering an associated flow rule and Mohr–Coulomb failure, and the assumption of the rigid displacement of the subdomains, the following relationship is obtained as

$$
\begin{align*}
2D: & \quad n_j - \tan \phi_j \cdot |s_j| = 0 \\
3D: & \quad n_j - \tan \phi_j \cdot \sqrt{s_j^2 + t_j^2} = 0
\end{align*}
$$

Equation (5) indicates $n_j \geq 0$, assuring the two subdomains separated by the discontinuity will not intrude into each other.

**It was pointed out in [7] that damage could happen at the supports as well. Therefore, [7] considered discontinuities at supports as inner discontinuities. In this paper, this type of damage is ignored for simplicity. Hence, the types of discontinuities depend only on their geometrical positions.

**Equation (5) indicates $n_j \geq 0$, assuring the two subdomains separated by the discontinuity will not intrude into each other.**
where \( p_j \geq \sqrt{s_j^2 + t_j^2} \) is the conic part in 3D DLO, and where the number of the cones equals the number of inner discontinuities.

Equation (6) indicates \( (n_j = p_j \tan \phi_j) \) for 3D DLO, as also given in [8]. By substituting \( n_j \) with \( (p_j \tan \phi_j) \) for every inner discontinuities \( j \), great computing resources are saved (less unknowns regarding linear dependencies), bringing the formulation of 3D DLO later in Section 2.4.

For all inner discontinuities in the domain, the following constraint is obtained:

\[
N^{\top} \begin{bmatrix} d_m \\ p \end{bmatrix} = 0, \quad (L)
\]

2.3. Loading and boundary conditions

As mentioned in Section 2.1, the displacement field of the domain is described by \( d \), with \( W(d) \) obtained as

\[
W(d) = f d = \begin{bmatrix} f_i \cdots f_m \end{bmatrix} \begin{bmatrix} d_i \\ \vdots \\ d_m \end{bmatrix}, \quad (F \cdot L)
\]

where \( f_i \) is the load vector (dead load) applied on discontinuity \( i \) [7].

For work performed by the live load \( W(l) \), the unit displacement constraint [7] is introduced as

\[
W(l) = F (Ad), \quad (F \cdot L)
\]

with

\[
Ad = \begin{bmatrix} A_i \cdots A_m \end{bmatrix} \begin{bmatrix} d_i \\ \vdots \\ d_m \end{bmatrix} = 1, \quad (L^3)
\]

where \( F \) is the unknown limit load \((F \cdot L^{-2})\).\(^\dagger\dagger\) \( A_i \) is a vector taking into account the area of the discontinuity \( i \) (when \( i \) is loaded by the live load) and the projection of unit vector parallel to the live load on the discontinuity \( i \).

Dissipated energy \( E \) is obtained by the plastic multiplier vector as

\[
E = ep = \begin{bmatrix} e_j \cdots e_k \end{bmatrix} \begin{bmatrix} p_j \\ \vdots \\ p_k \end{bmatrix}, \quad (F \cdot L)
\]

where \( e_j \) is the dissipated energy vector of the inner discontinuity \( j \). Considering \( c_j \) as the cohesion of discontinuity \( j \), \( e_j = [c_j l_j, c_j l_j] \) (for 2D) with \( l_j \) being the length of discontinuity \( j \) or \( e_j = [c_j a_j] \) (for 3D) with \( a_j \) being the area of the discontinuity \( j \).

As mentioned in Section 2.2, for the boundary discontinuities, the displacement jump equals to the absolute displacement of the discontinuities, which is used for prescribing fixed supports, such as

- If the discontinuity \( i \) is fixed (supported) along the shear direction, set \( s_i = 0 \) (for 2D) or \( s_i = 0, t_i = 0 \) (for 3D);
- If the discontinuity \( i \) is fixed (supported) along the normal direction, set \( n_i = 0 \).

\(^\dagger\dagger\)Conventionally, the dimensionless load factor \( \lambda (-) \) is optimized [7, 8], whereas in this paper the load \( 'F' \) is optimized, with the same value as \( \lambda \), but with dimension \((F \cdot L^{-2})\), such that \( F (F \cdot L^{-2}) = \lambda (-) \cdot 1 (F \cdot L^{-2}) \). Furthermore, \( W(l) (F \cdot L) = F (F \cdot L^{-2}) \cdot 1 (L^3) \), based on Equation (9).
Another type of constraint commonly encountered is locked discontinuities. For example, if discontinuities \( i \) and \( j \) are locked together (\( d_i = d_j \)), the following constraint is added into the optimization formulation:

\[
L_{ij} \begin{bmatrix} d_i \\ d_j \end{bmatrix} = \begin{bmatrix} I & -I \end{bmatrix} \begin{bmatrix} d_i \\ d_j \end{bmatrix} = 0, \quad (\mathcal{C})
\]  

where \( I \) is the unit matrix.

2.4. Optimization formulation

Considering Equations (4), (7), (8), (9), (10) and (11), the optimization formulation (written in matrix form) for the 2D DLO problem is obtained as follows:

\[
\begin{align*}
\text{minimize} & \quad W(l) = \mathcal{E} - W(d) = \begin{bmatrix} -\mathcal{F} & \mathcal{E} \end{bmatrix} \begin{bmatrix} d \\ p \end{bmatrix}, & (\mathcal{F} \cdot \mathcal{L}) \\
\text{subject to} & \quad \begin{bmatrix} B & 0 \\ N & L \\ A & 0 \end{bmatrix} \begin{bmatrix} d \\ p \end{bmatrix} = \begin{bmatrix} \mathcal{L} \\ \mathcal{L} \\ \mathcal{L} \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, & (\mathcal{L}^3) \\
\end{align*}
\]  

with

\( \rightarrow \) If discontinuity \( i \) is fixed along the shear direction:
set \( s_i = 0 \);
\( \rightarrow \) If discontinuity \( i \) is fixed along the normal direction:
set \( n_i = 0 \);
\( \rightarrow \) \( p \geq 0 \).

As mentioned before, to save computing resources in 3D DLO, \( n_j = p_j \tan \phi_j \) is considered for every inner discontinuity \( j \). After redefining \( d_j = [s_j, t_j, p_j]^T \) for every inner discontinuity \( j \), the following optimization formulation (written in matrix form) for 3D DLO is obtained

\[
\begin{align*}
\text{minimize} & \quad W(l) = \begin{bmatrix} \mathcal{F}^{new} \end{bmatrix} \begin{bmatrix} d^{new} \end{bmatrix}, & (\mathcal{F} \cdot \mathcal{L}) \\
\text{subject to} & \quad \begin{bmatrix} B^{new} \end{bmatrix} \begin{bmatrix} d^{new} \end{bmatrix} = \begin{bmatrix} \mathcal{L} \\ \mathcal{L} \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}, & (\mathcal{L}^3) \\
\end{align*}
\]  

with

\( \rightarrow \) If discontinuity \( i \) is a boundary discontinuity:
\( d_i = [s_i, t_i, n_i]^T \);
\( \rightarrow \) If discontinuity \( j \) is an inner discontinuity:
\( d_j = [s_j, t_j, p_j]^T \), with \( B^{new}_j = B_j \begin{bmatrix} 1 \\ \tan \phi_j \end{bmatrix} \)
\( \rightarrow \) If discontinuity \( i \) is fixed along the shear direction:
set \( s_i = 0 \) and \( t_i = 0 \);
\( \rightarrow \) If discontinuity \( i \) is fixed along the normal direction:
set \( n_i = 0 \).

where \( \mathcal{F}^{new} \) and \( \mathcal{A}^{new} \) are transformed from \([ -\mathcal{F} \mathcal{E} ] \) and \( \mathcal{A} \) by considering the redefinition of \( d_j = [s_j, t_j, p_j]^T \) and \( n_j = p_j \tan \phi_j \) for every inner discontinuity \( j \). Equation (13) is equivalent to the original formulation given in [8].
Equations (12) and (13) can be solved respectively using LP and SOCP solvers. As recommended by [7, 8], the author also uses the software ‘Mosek’ for solving the optimization problems, see [16] for details.

2.5. Standard pre-processing in discontinuity layout optimization

A fully connected grid method is used for pre-processing of 2D DLO problems [7]. First, grid points are given in the 2D domain as nodes, with every two nodes connected as a discontinuity. The loading and boundary conditions will be set at the corresponding discontinuities. The activated discontinuities giving minimum limit load $F$ will be automatically obtained after optimization, as illustrated in Figure 5.

This method is very direct and elegant in 2D DLO, with the maximum number of discontinuities equal to $C(m, 2)$ ($m$ is the number of nodes) considering the mathematical combinations available. On the other hand, when using a similar method for 3D DLO, that is, considering nodes in a 3D domain and connecting every three nodes as a 3D discontinuity, the maximum number of discontinuities becomes $C(m, 3)$, which dramatically increases with increasing $m$ (Figure 6).

Figure 5. Fully connected grid method for the two-dimensional (2D) discontinuity layout optimization (DLO) (a landslide example): (a) model for analysis and (b) fully connected model for optimization (only a subset of the discontinuities is shown for sake of clarity). [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 6. Maximum number of two-dimensional (2D) and three-dimensional (3D) discontinuities as functions of the number of nodes $m$ when using fully connected grid method. [Colour figure can be viewed at wileyonlinelibrary.com]
2.6. The overlap and crossing of discontinuities in discontinuity layout optimization

In the DLO, the overlap and crossing of the different discontinuities are commonly encountered, as illustrated in Figures 7 and 8.

Overlapping of discontinuities may give incorrect displacement jumps,‡‡ because displacement jumps are ‘shared’ in overlapped discontinuities. Moreover, overlapping of discontinuities will introduce large numbers of unnecessary discontinuities into the domain. In 2D DLO, distance checking is used to avoid the overlap of discontinuities. Two nodes \( p_1 \) and \( p_2 \) will only be connected by a discontinuity if \( |l - l_1 - l_2| > \epsilon \) (\( \epsilon \) is a prescribed small value, and \( p_a \) is all the other nodes except \( p_1 \) and \( p_2 \)) for assuring no other nodes are located between \( p_1 \) and \( p_2 \) (Figure 9).

The presence of discontinuities, which cross over, does not affect the numerical stability of DLO. The fact that discontinuities can cross over one another is a great advantage of this method over other limit analysis methods (e.g. the finite element method [17], when using rigid elements) because of the resulting very wide search space. What should be borne in mind is that crossed discontinuities are not directly connected. Because there is no common node located at the intersection point, the geometrical compatibility relationship (Equation (3)) will not be built for the crossed discontinuities. Therefore, only limited failure patterns are available for crossed discontinuities (Figure 10). Hence, in 2D DLO, to take into account large numbers of failure patterns, it is necessary to use a fine grid and ensure there are enough nodes and corresponding fully connected discontinuities [7].

‡‡The author uses the word “may” because the overlap of the discontinuities will not affect the final solution if these discontinuities are not activated.
Considering the 3D situation, obviously distance checking cannot be used to avoid the overlap of two 3D discontinuities. In addition, introducing more nodes into the 3D domain and building a set of fully connected 3D discontinuities will cause a huge increase in the number of discontinuities (Figure 6). The only possible routine is to use limited nodes and properly build 3D discontinuities, which are not overlapped or crossed, thereby avoiding unnecessary discontinuities as well as taking into account a large number of possible failure patterns.

Presently, the standard method for dealing with the overlap and crossing of 3D discontinuities is to check every two 3D discontinuities in the domain then add more nodes and split the relevant discontinuities into more discontinuities [8, 9] (Figure 11). With such type of pairwise checking, the number of times of this checking (one round) is about $C(C(m,3),2)$, with $m$ being the number of nodes. This process would occupy great computing resources to conduct on a personal computer when the number of the nodes exceeds 100 (Figure 12). Furthermore, because new discontinuities...
are introduced during every round of the checking, several rounds of this checking process are necessary within every round, and more discontinuities need to be checked than in the last round. The standard pre-processing of 3D DLO greatly restrains the potential wide application of this promising method. A more efficient strategy for modelling 3D discontinuities is essential for 3D DLO.

3. MULTI-SLICING STRATEGY

3.1. Geometrical relationship between slicings and three-dimensional discontinuities

A slicing is a plane, cutting the whole domain into two subdomains. Assuming the outside surfaces of a domain are also slicings and considering a domain cut by multiple slicings into several subdomains, the section of the domain of a specific slicing is composed of several convex polygons (Figure 13).

Inspecting the section of a domain considering a slicing $i$ as shown in Figure 14, every line is an intersection line of the slicing $i$ and another slicing and every point is an intersection point of the slicing $i$ and two other slicings. If the convex polygons formed by these lines and points are recognized as discontinuities, overlapping and crossing of the discontinuities are automatically avoided. Moreover, because every edge of the discontinuities is located on the intersection lines of the slicings, the displacement jump of the discontinuities on the corresponding slicings can be transferred to each other considering the geometrical compatibility relationship.

Considering recognizing the convex polygons on one plane from several intersected lines, some highly efficient algorithms are available such as in [18]. In this paper, on the other hand, the author uses a simple algorithm for this problem as illustrated in Figure 15. First, at every node, the intersected half lines are ordered with the polar angle (clockwise suggested by the author, bringing convenience for further building 3D discontinuities). Then, a start node and a search direction are specified, with the search direction turning to the neighbouring polar angle when coming to the next node. Once the search node comes to the start node again, a polygon is recognized. This process is repeated until every polygon has been recognized.

**Figure 12.** The number of times pairwise checking is required (one round) $C(C(m, 3), 2)$ as a function of the number of nodes $m$. [Colour figure can be viewed at wileyonlinelibrary.com]
3.2. Three-dimensional discontinuity considering shape of an arbitrary convex polygon

When a convex polygon is recognized, a corresponding 3D discontinuity is defined by a node loop such as $(\alpha, \beta, \cdots, \varepsilon)$ (where, $\alpha, \beta, \cdots, \varepsilon$ are the id of the nodes), with the direction of the normal displacement jump $n$ determined by the right hand rule (Figure 16). For 3D discontinuities on the same slicing, the author suggests a unified local coordinate system of $n$, $s$ and $t$ because of the associated easy coding during the programming processes (Appendix A). In addition, for discontinuities on the surfaces (boundary discontinuities), it is very important to assure the direction of $n$ is defined from the outside of the domain point to the inside of the domain.

Considering geometrical compatibility in 3D DLO, there are two directions for each edge (such as in Figure 8, directions $P \rightarrow Q$ and $Q \rightarrow P$)[8]. Hence, the sign should be determined during the assembly of the compatibility matrix $B$ (Section 2.1). In this paper, the sign is determined by
the id of the nodes in the node loop. For example, considering discontinuity ‘o’ shown in Figure 16, the local coordinate transformation matrix is \( B_o \). The contribution of discontinuity \( o \) to the relative displacement of the edge \( \alpha - \beta \) is \( X_{\alpha-\beta,o} = B_o d_o \). When assembling \( B_o \) into \( B \) (Equation (4)), the sign is determined by the id of the two nodes of the edge as

\[
\begin{pmatrix}
\vdots & \cdots & \vdots \\
\vdots \\
_{s \cdot B_o} \\
\vdots & \cdots & \vdots \\
\end{pmatrix}
\]

with \( s = \begin{cases} 
1 & \text{if } \alpha < \beta \\
-1 & \text{if } \alpha > \beta 
\end{cases} \)

(14)

The multi-slicing procedure for 3D DLO is summarized as follows:

1. Multiple slicings are prescribed;
2. Based on the multiple slicings, intersection points of every three non-parallel slicings are calculated and defined as nodes;
3. All nodes on one specific slicing are collected; then all convex polygons on this slicing formed by intersection lines are recognized as 3D discontinuities. This process is repeated until all slicings are checked;
4. With the recognized 3D discontinuities and the constraint given in Equation (14) assembled, the optimization formulation (SOCP problem, see Equation (13)) is built and sent to the optimization solver for solving.

4. NUMERICAL EXAMPLES

A personal computer with Intel Core i7-3930K 3.20G Hz processor and 16 GB memory was used to obtain the numerical results. The SOCP problem (Equation (13)) is built via the multi-slicing strategy presented then solved by Mosek 7.1 [16].

© 2016 The Authors. International Journal for Numerical and Analytical Methods in Geomechanics published by John Wiley & Sons Ltd. DOI: 10.1002/nag
4.1. Footing problem

The weightless footing problem [19] is shown in Figure 17, with the live load ‘F’ applied perpendicular on the marked region. When assuming the angle of friction $\phi = 0^\circ$ and unified cohesion $c$ for all discontinuities, the analytical solution for the limit loading ‘F’ is equal to $(2 + \pi)c$.

Three cases with different slicings are considered (Figure 18). Key points are the points prescribed for defining the slicings, with every three non-collinear points providing a slicing. The details

---

Footnote: The footing problem shown in this paper is actually a 2D problem and only the discontinuities perpendicular to the $y-z$ plane will finally become activated. However, during the calculation, the author considers a general condition with all discontinuities given by the slicings taken into account.
Table I. Details of calculations for the footing problem.

| Case  | Number of slicings | Number of nodes | Number of discontinuities | Number of edges | CPU time for modelling (pre-processing) | CPU time for solving SOCP | Number of slicings with activated discontinuities | Error |
|-------|-------------------|----------------|---------------------------|----------------|----------------------------------------|--------------------------|-----------------------------------------------|-------|
| I     | 40                | 847            | 3652                      | 3101           | 0.11 s                                 | 3.93 s                   | 8                                             | 6.81c | 32.4%           |
| II    | 93                | 21837          | 78612                     | 72276          | 17.24 s                                | 1081.69 s                | 10                                            | 6.16c | 19.8%           |
| III   | 137               | 66,554         | 243080                    | 221665         | 155.46 s                               | 11535.81 s               | 14                                            | 5.92c | 15.1%           |

SOCP, second-order cone programming.

Figure 19. Activated discontinuities for the footing problem. [Colour figure can be viewed at wileyonlinelibrary.com]

of the calculations are given in Table I. Activated discontinuities are shown in Figure 19. The results indicate the applicability of the presented method. With the increasing number of slicings, the results are improved comparing to the analytical solution, with acceptable time cost for modelling. Theoretically, when introducing very large number of slicings, very accurate results would be obtained.

Nevertheless, unlike the LP problem used in 2D DLO, the SOCP problem is much more difficult to solve, taking a long time (Table I) and occupying great computing resources (e.g. around 10 GB memory are occupied by Mosek when solving case III), which increases dramatically with increasing number of slicings. Besides, introducing more slicings will reduce the distance between nodes. When some nodes are ‘too close’ to each other considering the computational precision of the variables stored on the computer system, linear dependencies will be introduced in the global constraint matrix, greatly affecting the numerical stability of the optimization solver.***

Considering the computational cost and assuring the numerical stability of the solver, presently, a number of slicings less than 100 is suitable, which is unfortunately not enough to obtain a mighty accurate result. For better results, some techniques considering engineering properties of the case should be used, for example, taking advantage of symmetry and introducing more slicings in the region close to the live load.

4.2. Three-dimensional landslide

A weightless 3D landslide example is shown in Figure 20, with uniform cohesion $c$ assumed. Owing to symmetry, only half of the problem is considered. The prescribed key points and the correspond-
Figure 20. Model of the three-dimensional (3D) landslide. [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 21. The discontinuities of the three-dimensional (3D) landslide: key points prescribed for slicings (every three non-collinear key points providing one slicing) and discontinuities built with multi-slicing method (only visible discontinuities are shown). [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 22. Activated discontinuities and associated limit loads $F$ for the three-dimensional (3D) landslide with the discontinuities shown in Figure 21. [Colour figure can be viewed at wileyonlinelibrary.com]

Considering angle of frictions $\phi = 0^\circ, \phi = 15^\circ$ and $\phi = 30^\circ$, the corresponding activated discontinuities and limit loads of the 3D landslide problem are shown in Figure 22, indicating a large failure zone and the possibility that the limit loads are overestimated.

Considering the failure pattern of the 3D landslide that many activated discontinuities appear at the loaded region, the author implemented a simple grid searching procedure into the multi-slicing strategy to obtain better results. Several standing points are prescribed, then a search point is added, which moves on the grid (Figure 23). The multiple slicings include the surfaces and the slicings defined by the search point and another two standing points (three points are non-collinear). In one search round, after all points on the grid are searched and calculated, the search point with minimum energy/limit load will be added as a new standing point; meanwhile, the corresponding slicings with activated discontinuities are also kept in the next search round.
With two 50 × 50 grids (Figure 23) and two search rounds on each grid,††† considering \( \phi = 30^\circ \), the obtained activated discontinuities, the standing points and search point in the end of every search round are shown in Figure 24, indicating the results are greatly improved with this search procedure. The final obtained minimum limit loads and failure patterns for the 3D landslide considering \( \phi = 0^\circ \), \( \phi = 15^\circ \) and \( \phi = 30^\circ \) are shown in Figure 25, which are more reasonable than the results given in Figure 22.

4.3. Punch indentation

As a classic example, the punch indentation analysed in [8, 20, 21] is reanalysed here. A block with its bottom fixed and lateral surfaces restrained along the normal direction (\( n = 0 \)) is loaded in the centre. Only 1/8 of the volume is considered, owing to symmetry (Figure 26), with uniform cohesion \( c \) considered.

As mentioned before, overlapping and crossing problems can also be avoided if discontinuities are transformed from the surfaces of 3D elements of a discretized domain. For testing this strategy, the model is discretized with tetrahedra (similar discretization is used in finite element method analysis in [22]) and the surfaces of the tetrahedra are transformed into discontinuities (Figure 27). The results are listed in Table II,‡‡‡ regarding the edge length of the cube equals 1/4 (denoted as cube – 1/4), 1/16 (denoted as cube – 1/16) and 1/24 (denoted as cube – 1/24). From the results, it can be found that one drawback of this strategy is that a finer mesh does not ensure a corresponding much better result, but does take much more computing time.

Then the multi-slicing method with grid searching procedure is used. The initial standing points for the search procedure are shown in Figure 26. In this example, grids similar to cobweb are

---

†††Two grids are searched, first grid 1 then grid 2. This process was run twice (two search rounds for each grid).
‡‡‡Because the CPU time listed in Table II referring the work of [8] was obtained with a different computer, direct comparison could be unfair.
prescribed on the surfaces $EFCB$, $ACFD$ and $ADEB$ (Figure 28), with four search rounds used (each surface is searched four times). Obtained and literature results are listed in Table II, with the obtained activated discontinuities and deformation pattern regarding $\phi = 0^\circ$ and $25 \times$ grid shown in Figure 29. It can be seen that using the multi-slicing method combing with the search procedure provides satisfying results even with $5 \times$ grid, which takes much less computing time than the case.
Table II. Punch indentation: comparisons with benchmark solutions (upper bound).

| Provided by       | Methods                  | \( \phi = 0^\circ \) | CPU time | \( \phi = 30^\circ \) | CPU time |
|-------------------|--------------------------|-----------------------|----------|-----------------------|----------|
| [20]              | Multi-block analysis     | \( F = 6.561c \)     | —        | \( F = 104.019c \)    | —        |
| [21]              | FEM limit analysis       | \( F = 6.051c \)     | —        | \( F = 66.936c \)     | —        |
| [8]               | 3D DLO                   | \( F = 6.226c \)     | 6400 s   | —                     | —        |
| This work         | 3D DLO (cube – 1/4)     | \( F = 6.433c \)     | 0.1 s    | —                     | —        |
| This work         | 3D DLO (cube – 1/16)    | \( F = 6.419c \)     | 140 s    | —                     | —        |
| This work         | 3D DLO (cube – 1/24)    | \( F = 6.418c \)     | 4371 s   | —                     | —        |
| This work         | 3D DLO (5\times multi-slicing) | \( F = 6.094c \)   | 2037 s   | \( F = 73.196c \)    | 6249 s   |
| This work         | 3D DLO (10\times multi-slicing) | \( F = 5.904c \)   | 11144 s  | \( F = 69.520c \)    | 38035 s  |
| This work         | 3D DLO (25\times multi-slicing) | \( F = 5.892c \)  | 80322 s  | \( F = 68.225c \)    | 272156 s |

Figure 28. Grid points for grid searching procedure (only 5\times and 10\times are shown). [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 29. Obtained activated discontinuities and deformation pattern for the punch indentation problem regarding \( \phi = 0^\circ \) and 25\times grid. [Colour figure can be viewed at wileyonlinelibrary.com]

cube – 1/24. Furthermore, the results are continuously improved with finer search grid, indicating the effectiveness and robustness of this method.

5. CONCLUSION

In this paper, in order to avoid overlapping and crossing of 3D discontinuities, a multi-slicing strategy was presented, with 3D discontinuities being arbitrary convex polygons. The presented method has been proven to be effective, and numerical results are improved with increasing numbers of slicings. Although the number of slicings is still limited because of the complexity at solving large scale SOCP problems, considering the engineering properties of the cases, the author took the advantage of the symmetry and implemented a grid searching procedure, obtaining much better results. The work proposed in this paper opens the door to the application of the 3D DLO to some large scale engineering problems.
APPENDIX A: THE COORDINATE TRANSFORMATION MATRIX IN THREE-DIMENSIONAL DISCONTINUITY LAYOUT OPTIMIZATION

As mentioned in Section 3, the author suggests a unified local coordinate system for discontinuities on the same slicing, which makes the compatibility matrix for all the discontinuities lying on this slicing the same, bringing great convenience for coding. Figure A.1 illustrates a 3D discontinuity \( o \) lying on slicing \( i \) with the corresponding local coordinate system \( s, t, n \) and global coordinate system \( x, y, z \). When \( n \) being perpendicular to the slicing \( i \), \( s \) is determined by two farthest nodes \( n_1 \) and \( n_2 \) lying on the slicing \( i \).

![Figure A.1. Three-dimensional (3D) discontinuity \( o \) lying on the corresponding slicing \( i \): local and global coordinate system.](Colour figure can be viewed at wileyonlinelibrary.com)

Assuming the unit vectors parallel to \( s, t, n \) are defined as \( \mathbf{v}_s, \mathbf{v}_t, \mathbf{v}_n \), the coordinates of \( \mathbf{v}_s, \mathbf{v}_t, \mathbf{v}_n \) in \( x, y, z \) system are \((v_{sx}, v_{sy}, v_{sz}), (v_{tx}, v_{ty}, v_{tz}), (v_{nx}, v_{ny}, v_{nz})\). Considering a change of basis, the compatibility matrix of \( o \) as \( \mathbf{B}_o \) (Equation (14)) is determined by

\[
\mathbf{B}_o = \begin{bmatrix}
 v_{sx} & v_{sy} & v_{sz} \\
 v_{tx} & v_{ty} & v_{tz} \\
 v_{nx} & v_{ny} & v_{nz}
\end{bmatrix}^{-1}, 
\tag{A.1}
\]

ACKNOWLEDGEMENTS

The author gratefully acknowledges the financial support provided by the Austrian Science Fund (FWF), grant no. P 25614 ‘Adaptive Discontinuity Layout Optimization’.

REFERENCES

1. Bendsøe P, Sigmund O. Topology Optimization: Theory, Methods and Applications (2nd ed.) Springer: Berlin, 2004.
2. Gilbert M, Tyas A. Layout optimization of large-scale pin-jointed frames. Engineering Computations 2003; 20: 1044–1064.
3. Gilbert M, Darwich W, Tyas A, Shepherd P. Application of large-scale layout optimization techniques in structural engineering practice. In 6th World Congresses of Structural and Multidisciplinary Optimization. CD-ROM: Rio de Janeiro, 2005.
4. Gilbert M, Smith CC, Hawksbee S, Tyas A. Use of layout optimization to solve large-scale limit analysis and design problems. In Direct Methods for Limit States in Structures and Materials, Spiliopoulos K, Weichert D (eds). Springer: Berlin, 2014; 157–180.
5. Chen W-F. Limit Analysis and Soil Plasticity. J.Ross: Florida, 2008.
6. Gilbert M, Smith CC. Discontinuity layout optimization: A new numerical procedure for upper bound limit analysis. IX International Conference on Computational Plasticity, Barcelona, 2007; 170–173.
7. Smith CC, Gilbert M. Application of discontinuity layout optimization to plane plasticity problems. Proceedings of The Royal Society A 2007; 463:2461–2484.
8. Hawksbee S, Smith CC, Gilbert M. Application of discontinuity layout optimization to three-dimensional plasticity problems. Proceedings of The Royal Society A 2013; 469:20130009.
9. Hawksbee S. 3D ultimate limit state analysis using discontinuity layout optimization. Ph.D. Thesis, University of Sheffield, 2012.
10. Smith CC, Gilbert M. Identification of rotational failure mechanisms in cohesive media using discontinuity layout optimization. *Géotechnique* 2013; 63:1194–1208.

11. Clarke SD, Smith CC, Gilbert M. Modelling discrete soil reinforcement in numerical limit analysis. *Canadian Geotechnical Journal* 2013; 50:705–715.

12. Gilbert M, He L, Smith CC, Le CV. Automatic yield-line analysis of slabs using discontinuity layout optimization. *Proceedings of The Royal Society A* 2014; 470:20140071.

13. He L, Gilbert M. Automatic rationalization of yield-line patterns identified using discontinuity layout optimization. *International Journal of Solids and Structures* 2016; 84:27–39.

14. Gilbert M, Smith CC, Pritchard TJ. Masonry arch analysis using discontinuity layout optimisation. *Proceedings of the Institution of Civil Engineers - Engineering and Computational Mechanics* 2010; 163:155–166.

15. Smith CC, Cubrinovski M. Pseudo-static limit analysis by discontinuity layout optimization: application to seismic analysis of retaining walls. *Soil Dynamics and Earthquake Engineering* 2011; 31:1311–1323.

16. The MOSEK optimization tools manual (Version 7). https://www.mosek.com/resources/doc, 2015. Copenhagen, Denmark: Mosek ApS.

17. Sloan SW. Upper bound limit analysis using finite elements and linear programming. *International Journal for Numerical and Analytical Methods in Geomechanics* 1989; 13:263–282.

18. Ferreira A, Fonseca MJ, Jorge JA. Polygon detection from a set of lines. *Actas do 12 Encontro Português de Computação Gráfica (12the EPCG)*, Porto, Portugal, 2003; 159–162.

19. Prandtl L. Über die Härte plastischer Körper. *Nachrichten der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1920:74–85.

20. Michalowski RL. Upper-bound load estimates on square and rectangular footings. *Géotechnique* 2001; 51:787–798.

21. Vincente da Silva M, Antão AN. Upper bound limit analysis with a parallel mixed finite element formulation. *International Journal of Solids and Structures* 2008; 45:5788–5804.

22. Martin CM, Makrodimopoulos A. Finite-element limit analysis of Mohr–Coulomb materials in 3D using semidefinite programming. *Journal of Engineering Mechanics (ASCE)* 2008; 134:339–347.