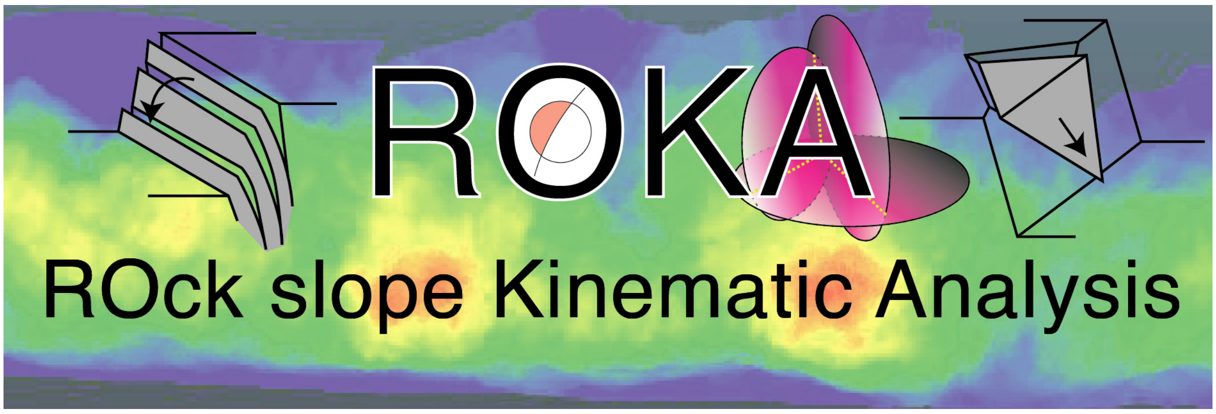


USER MANUAL
of

ROKA v.1.0

Index

Preface

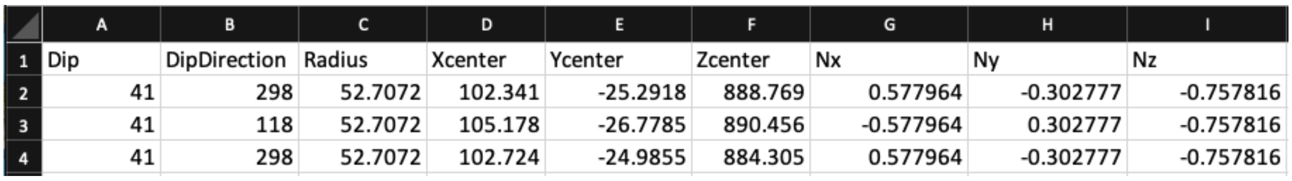
This user manual describes how to use the ROck slope Kinematic Analysis (ROKA) algorithm written in MATLAB language.
The algorithm uses the point cloud and the 3D mapped discontinuity information to perform an ad-hoc kinematic analysis for all the different portion of the slope, taking into account the local orientation of the slope and the position, dimension and attitude of the mapped discontinuity planes.
Moreover, the algorithm can perform a kinematic test for all the real discontinuities intersections, calculated by the real intersection of the mapped discontinuity plane.
In the next pages, all the steps necessary for the application of the ROKA algorithm will be described.
A detailed explanation of the algorithm will be available in Menegoni et al. 2021 (https://doi.org/10.3390/app11041698).

1) Input data formatting

The input data of the ROKA algorithm are (1) the point cloud with normals representing the rock slope and (2) the geometric information about the 3D mapped discontinuities.

(1) The point cloud must be a text files (.txt) containing a series of row where the *x*, *y* and *z* point coordinates and the *Nx*, *Ny* and *Nz* normal vector components must be defined for each point of the point cloud. The information must be order as *x*, *y*, *z*, *Nx*, *Ny*, *Nz* and must be delimited by space. To simplify the algorithm calculation, a local coordinates reference system is preferred to respect a geographical one (es. WGS84/UTM) due to the reduction of the coordinates digits.

(2) The geometric information about the 3D mapped discontinuities must be stored in a xlsx file and must contain at least 9 headed columns regarding dip and dip direction angles, the radius of the discontinuity, the x, y and z coordinates of the center of the discontinuity and the Nx, Ny and Nz components of the discontinuity plane normal vectors. As depicted in Fig. 1, the headers must be named as *Dip*, *DipDirection*, *Radius*, *Xcenter*, *Ycenter*, *Zcenter*, *Nx*, *Ny* and *Nz*, and can be reported in any order.



	A	B	C	D	E	F	G	H	I
1	Dip	DipDirection	Radius	Xcenter	Ycenter	Zcenter	Nx	Ny	Nz
2	41	298	52.7072	102.341	-25.2918	888.769	0.577964	-0.302777	-0.757816
3	41	118	52.7072	105.178	-26.7785	890.456	-0.577964	0.302777	-0.757816
4	41	298	52.7072	102.724	-24.9855	884.305	0.577964	-0.302777	-0.757816

Fig. 1 Example of the format of the Excel file containing the geometric information of the 3D mapped discontinuities.

Option to create the discontinuity information Excel file:

i. In the main folder of the algorithm is present a script, called '*dxf_plane_fit.m*', that is able to fit a 3D disc and extract a Excel file with all the information required by ROKA, from a dxf file that contains all the discontinuity traces mapped onto the 3D model.

ii. using the Cloud Compare open-source software, there are several way to map the discontinuities (e.g. tool Trace Polyline; plugin Compass developed by Thiele et al., 2017), find their best-planes (e.g. tool-->Fit-->Plane) and then export the best-fit planes information (tool-->Batch Export--> Export plane info).

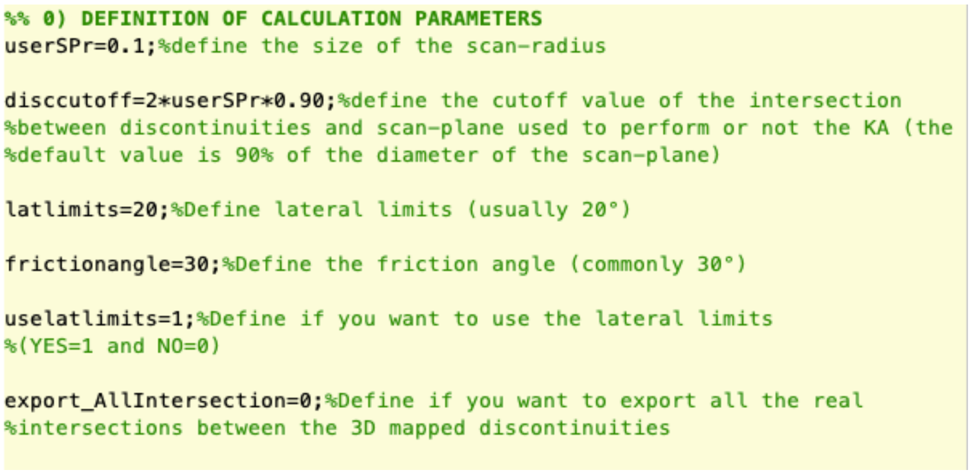
The main problem with the second method is that it fits to the mapped discontinuity a 3D planar rectangle and, therefore, the 'radius' of the Beacher's discs, that represent the discontinuity and that is required by the ROKA algorithm, is not calculated. To achieve the radius, the authors suggest to use the Pithagoras' Theorem to calculate the max extension of fitted rectangle and consider it as the diameter of the discontinuity Beacher's disc. Therefore, for example it is possible to calculate the radius from the height and width of a rectangular best-fit plane using the formula:

$$radius = \frac{\sqrt[2]{(length^2 + height^2\ )}}{2}$$

where *radius* is the radius of the Beacher's disc representing the discontinuity, *length* is the length of the rectangular best-fit plane and *height* is the height of the rectangular best-fit plane. This calculation could be done in a very simple way using spreadsheets.
The data will be load during after the launch/start/run of the code.

2) Calculation parameters definition

Before launching the ROKA algorithm analysis, the user must defined (1) the size of the scan radius, (2) the cutoff value of the intersection between discontinuity and scan-plane used to consider or not the discontinuity during the kinematic analysis, (3) the lateral limits and friction angle used for the kinematic analysis, (4) if use or not the later limits and (5) if export or not all the lines representing the intersection between the discontinuities (see Fig. 2).



```
%% 0) DEFINITION OF CALCULATION PARAMETERS
userSPr=0.1;%define the size of the scan-radius

disccutoff=2*userSPr*0.90;%define the cutoff value of the intersection
%between discontinuities and scan-plane used to perform or not the KA (the
%default value is 90% of the diameter of the scan-plane)

latlimits=20;%Define lateral limits (usually 20°)

frictionangle=30;%Define the friction angle (commonly 30°)

uselatlimits=1;%Define if you want to use the lateral limits
%(YES=1 and NO=0)

export_AllIntersection=0;%Define if you want to export all the real
%intersections between the 3D mapped discontinuities
```

Fig. 2 Portion of the MATLAB code where the parameters must be defined.

(1) The size of the discontinuities, named *userSPr*, must defined in meter units: for example, a radius of 15cm must be defined as '*userSPr* = 0.15'.

(2) The cutoff value of the intersection between discontinuity and scan-plane used to consider or not the discontinuity during the kinematic analysis, named '*disccutoff*', must be expressed in meters units. The default value is the 90% of the scan-plane diameter.

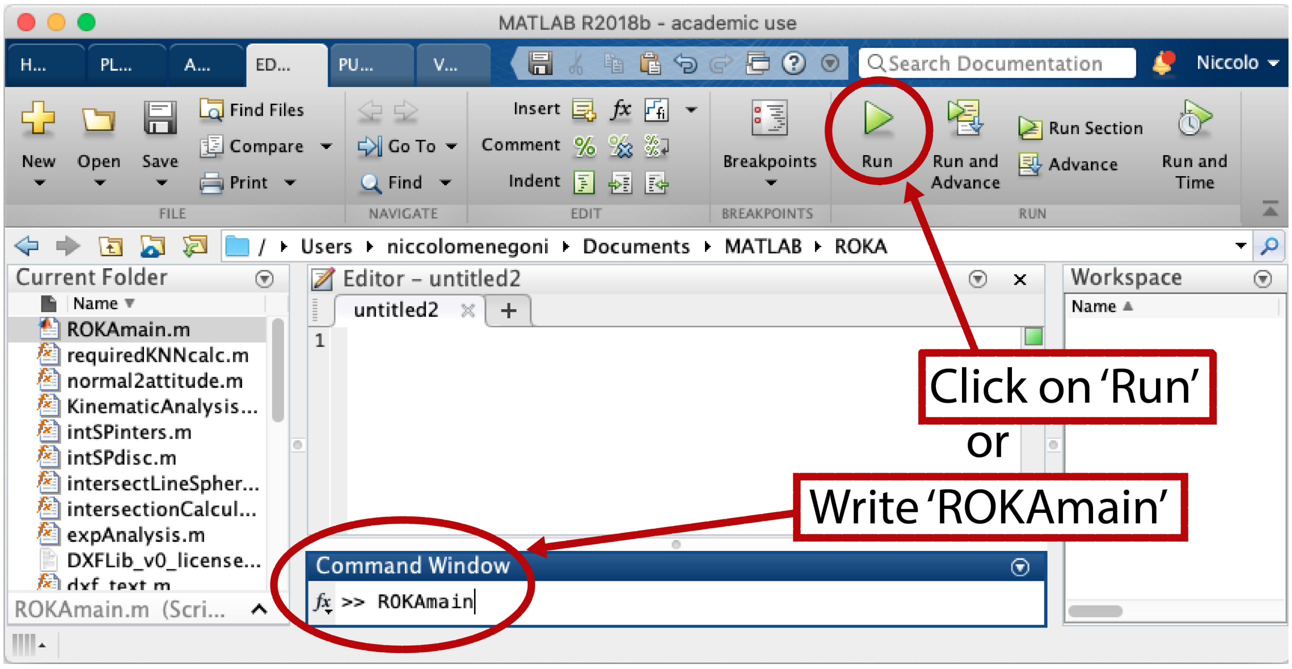
(3) The lateral limits and the friction angle, named '*latlimits*' and '*frictionangle*', must be defined in degrees units.

(4) the binary option for the usage of the lateral limits during the kinematic analysis, named '*uselatlimits*', must be 1, for the use, or 0, for the non-use.

(5) the binary option for the exportation of all the real discontinuities intersections, named '*export_AllIntersection*', must be 1, for the exportation, or 0, for the non-exportation.

3) Run the algorithm

To run the algorithm, the user must open MATLAB software, set the ROKA algorithm folder as *current folder*, open the main script called '*ROKAmain.m*' and then launch the code.
To launch the code, the user can write '*ROKAmain*' directly in the Command Window or click onto '*Run*' button in *Editor* toolbar (see Fig. 3).



```
>> ROKAmain
```

Fig. 3 How to launch the code in MATLAB.

4) Output data description

The outputs of the ROKA algorithm are stored in a folder called '*KinAnalysis*' created in the same path of the input data. The outputs can be identified with (1) a point cloud with all the useful information and (2) the 3D files of the critical discontinuity planes and intersections.

(1) The output point cloud is exported as a headed text file (.txt) and for each point it reports in the following order : the x, y and z point coordinates (X Y Z); the local orientation of the slope as dip direction and dip angles (DipDir Dip), the number of discontinuity planes and intersections that intersect the slope (AllDisc_Inters AllInt_inters), the local critical value for the planar sliding, flexural toppling, wedge sliding and direct toppling modes of failure (PlanarSliding FlexuralToppling WedgeSliding DirectToppling).
The critical values reported onto the different point of the cloud correspond to the maximum critical value of the discontinuity that intersect the different portion of the point cloud.

(2) The critical discontinuity planes and intersection are stored as a 3D file (.dxf) in different folder representing the different modes of failure for the discontinuity planes and intersections. For example, a discontinuity plane that appears to be critical during the planar sliding kinematic analysis, using the lateral limits, is stored in the path *'KinAnalysis → Disc → PlanarSlidingLatLimits*' . Instead, a discontinuity intersection that appears to be critical for the direct toppling is stored in the path '*'KinAnalysis → Inters→ DirectToppling.*'
The 3D files of the critical discontinuity planes are named using these information in the following order: the critical value of the critical discontinuity and the index of the critical discontinuity planes. For example, '*100_CriticDisc_1*' means that the critical value and the index of the critical discontinuity are equal to 100 and 1, respectively.
The 3D files of the critical discontinuity intersections are named using these information in the following order: the critical value of the critical discontinuity intersection; the index of the critical intersection; the indexes of the two discontinuity planes that form the critical intersection. For example, '*75_Inters_1_Discs_1_5*' means that the critical value and index of the critical intersection are respectively 75 and 1, and that the intersection is formed by the discontinuity planes with index 1 and 5.

For a description of how these output information are calculated please see the Menegoni et al. 2021 (https://doi.org/10.3390/app11041698).

4) Authors contact

Niccolò Menegoni, Ph.D
Department of Earth and Environmental Sciences - University of Pavia
niccolo.menegoni@unipv.it
niccolo.menegoni@live.it