

Luis Alejandro Erazo Martínez
Lúdrger O. Suárez Burgoa

VOLTEOLAB 0.50

PROGRAMA PARA MODELAR LA FALLA POR
VOLCAMIENTO EN DOS DIMENSIONES

DICIEMBRE , 2016

Manual de Usuario

Universidad Nacional de Colombia – Medellín

Los nombres de compañías y productos mencionados en este libro son marcas corporativas o registradas de sus respectivos dueños.

©Copyright 2016 Universidad Nacional de Colombia.

©Copyright 2016 Luis Alejandro Erazo Martínez & Ludger O. Suárez-Burgoa.

Este manual se encuentra bajo la licencia Creative Commons Attribution 3.0.

Descripción catalográfica

Título: volteoLab 0.50
Subtítulo: programa para modelar la falla por volcamiento en dos dimensiones
Edición: primera
Características: 17 cm × 24 cm, 31 pp., 5 il., español, 2016
Subárea: 624.151 Ingeniería geológica
Autor(es): Erazo Martínez, Luis Alejandro (Colombia), 1989–
Suárez Burgoa, Ludger Oswaldo (Bolivia), 1975–

A Dios por sembrar y cultivar cada día la semilla de la investigación y perseverancia en mi vida, y a mi madre, padre, hermano, por ayudarme a llevar este barco hacia buen puerto. (LAEM)

A los desarrolladores latinoamericanos de software libre en temas de Geociencias. (LOSB)

Prefacio

El programa *volteoLab 0.50* fue diseñado para modelar el fenómeno de falla por volcamiento en dos dimensiones (2D), teniendo en cuenta los conceptos tales como el coeficiente de transferencia y el peso de volcamiento bajo la metodología planteada por (Liu et al.,2009)[4].

Es importante tener en cuenta que el resultado que se espera obtener mediante el análisis de la falla por volcamiento en macizos rocosos es la *fuerza de anclaje* o *fuerza tensora* que posibilita garantizar el equilibrio de fuerza en todo el sistema. Sin embargo, es importante a su vez tener claridad en la concepción geométrica del problema, ya que sin ésta no es posible entender la orientación y acción de las distintas fuerzas que intervienen.

Se presenta entonces una alternativa, en forma de código abierto en un lenguaje de programación, llamada *volteoLab 0.50* que posibilite al usuario entender, plasmar y orientar frente a este concepto y problema analítico en geotecnia. El fin es el de generar la retroalimentación conceptual y quizás poder nutrir el código para futuras y mejores versiones; y siendo a su vez motor de motivación para los nuevos desarrolladores de software en la parte de la geomecánica computacional.

Medellín, diciembre de 2016.

Luis Alejandro Erazo Martínez

Geotecnista

Maestría en Ingeniería Geotecnia

Universidad Nacional de Colombia

Ludger O. Suárez-Burgoa

Profesor Asistente

Departamento de Ingeniería Civil

Universidad Nacional de Colombia

Índice general

1. Descripción	1
1.1. El objetivo motivo de estudio	1
1.2. Tipos de volcamiento	1
1.3. Análisis físico general de la falla por volcamiento	7
1.4. El programa	9
2. Instalación	13
2.1. Descarga	13
2.2. Instalación	13
2.3. Requerimientos del sistema	14
2.4. Ejecución del programa	14
3. Primera simulación	15
3.1. Determinación de los coeficientes a_g y a_s	15
3.2. Posición del bloque de cresta m	15
3.3. Altura de todos los bloques de análisis h_i	16
3.4. Altura de bloque de cresta h_m	17
3.5. Radio de esbeltez χ	17
3.6. Calculo de los pesos brutos W_i	17
3.7. Matriz de coeficientes de transferencia ψ_i	17
3.8. Matriz de pesos geométricos o de volcamiento W'_i	18
3.9. Matriz de fuerzas normales N_i	18
3.10. Factor de estabilidad al deslizamiento f_i	19
3.11. Fuerza de anclaje o soporte P	20
3.12. Busque ayuda	20

4. Opciones avanzadas	21
4.1. Modelado geométrico macizo rocoso	21
4.2. Modifique el código a su agrado	25
5. Ejemplos	27
5.1. Cálculo de la falla por volcamiento para 13 bloques	27
5.2. Análisis de la falla por volcamiento para 34 bloques.	29
Referencias	33
Anexos	35
A.1. Alojamiento y desarrollo del código	35
A.2. Licencia	35
A.3. Descargo de responsabilidades	36
Índice alfabético	37

Capítulo 1

Descripción

1.1. El objetivo motivo de estudio

El objetivo principal del código abierto `volteoLab` en su versión 0.50 (`volteoLab 0.50`) es el de posibilitar al usuario modelar el tipo de falla geotécnica denominada *volcamiento* en dos dimensiones y a su vez calcular la fuerza de soporte necesaria para lograr que el sistema de bloques de análisis se encuentre en equilibrio; teniendo en cuenta lo siguiente:

- conjuntos de bloques en equilibrio, volcamiento y deslizamiento;
- identificación de bloques sometidos a deslizamiento para el cálculo de la fuerza de soporte P .

1.2. Tipos de volcamiento

En la literatura, autores como (Goodman y Bray, 1976)[2] han definido dos tipos de falla por volcamiento que son detectados en campo: *volcamiento por flexión* y *volcamiento de bloques* los cuales se definen a continuación.

1. Volcamiento de bloques.

Este tipo de falla está caracterizado por la formación de dos tipos de columnas de roca. La primera tiene una abundancia de discontinuidades que buzan en la cara del talud, y la segunda forma articulaciones, las cuales tienen un comportamiento mucho más marcado a la rotación, Ver Figura 1.1.

En el caso de la primera columna, de una altura relativamente mayor a las demás existe junto con la segunda una fuerza de empuje hacia los bloques de la pata de talud, lo cual deja a la cara principal de los bloques de mayor altura la libertad para que a través

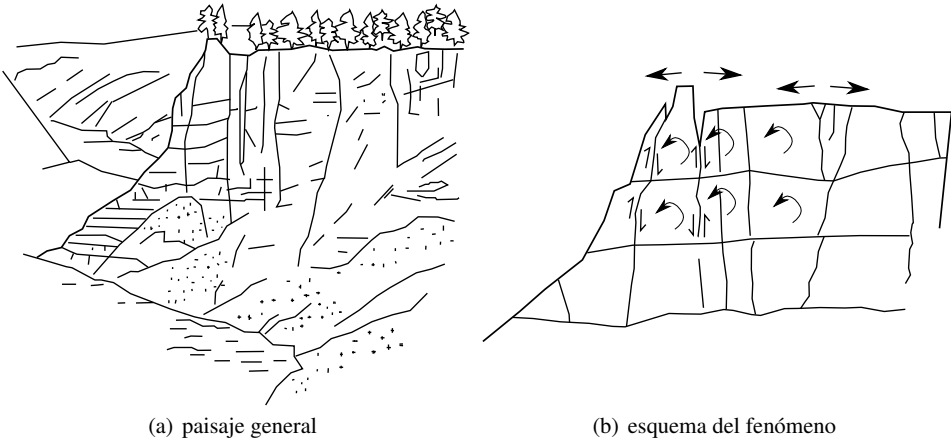


Figura 1.1 Mecanismo sugerido de volcamiento de la cara norte del deslizamiento de Vajont (Muller, 1968)[5] (Tomado de (Muller, 1968)[5]).

del vencimiento de fuerzas se presente el proceso cíclico de volteo de los bloques. Un ejemplo ilustrativo es el planteado por los autores (Cundall, 2013)[1], el cual muestra la simulación del volteo de bloques a través de un software de computadora (Figura 1.2).

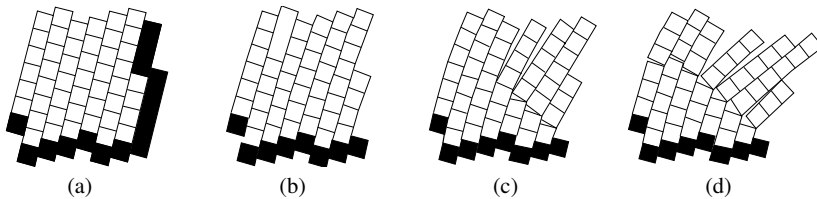


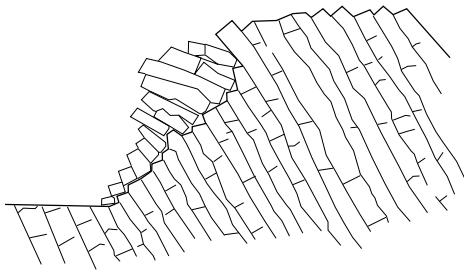
Figura 1.2 Modelo de cómputo generado de la falla por volcamiento; los bloques sólidos son fijados en el espacio mientras que los bloques abiertos están libres a movimiento (Tomado de (Cundall, 2013)[1]).

2. Volcamiento por flexión.

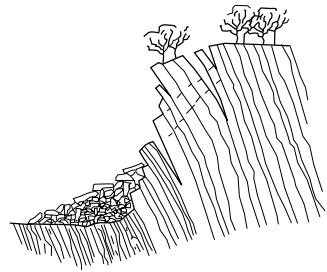
Este tipo de volcamiento se desarrolla en columnas continuas de roca (Figura 1.3(b)), las cuales están separadas por discontinuidades de alta persistencia. Se presenta falla por flexión debido a la relación altura ancho de los bloques y al desplazamiento del centro de gravedad de estos, ya que estos se doblan hacia adelante (Hoek y Bray, 1981)[3]. Los materiales geológicos identificados en este tipo de falla son los esquistos en finas capas, las pizarras y las lutitas, ya que la discontinuidad ortogonal no se desarrolla de

una manera adecuada. En cuanto al plano de base de este tipo de fallas no está tan bien definido como lo está en la falla de volcamiento de bloques.

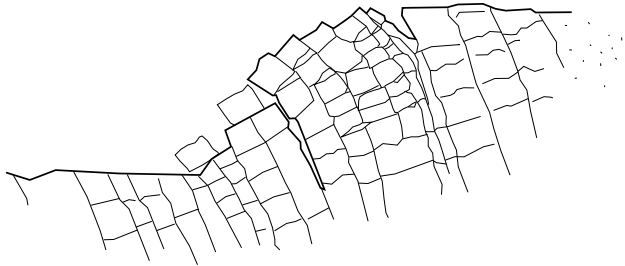
El proceso detonante de este tipo de falla consta de erosión en la pata del talud, que hace que el primer bloque se voltee; lo que origina pérdida de soporte del resto del conjunto y por lo tanto da lugar a la formación de grietas profundas que hacen que el proceso sea sucesivo en toda la serie. Muchas veces la parte inferior de los taludes está cubierta por partes de bloques los cuales conforman columnas ya falladas, y por esta razón es difícil de detectar este tipo de falla en los macizos rocosos. Pruebas de campo han demostrado que en el momento de presentarse la falla de la columna de la cara externa del talud, o la columna 1 de la serie n , se presenta deslizamiento interno de las demás columnas, lo cual hace que haya exposición de los demás bloques (Figura 1.3(a)).



(a) volcamiento de bloques de columnas de roca que contiene juntas espaciadas ampliamente



(b) volcamiento por flexión de losas de roca orientadas en sentido interno a la cara del talud



(c) volcamiento de bloques y flexión caracterizado por flexión pseudo-continua de columnas de gran longitud que acumula desplazamientos a lo largo de juntas cruzadas

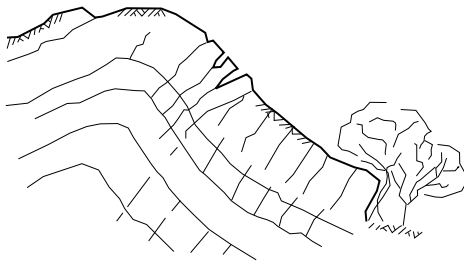
Figura 1.3 Tipos comunes de fallas por volcamiento (Tomado de (Goodman y Bray,1976)[2]).

3. Volcamiento combinado por flexión y del tipo de bloques.

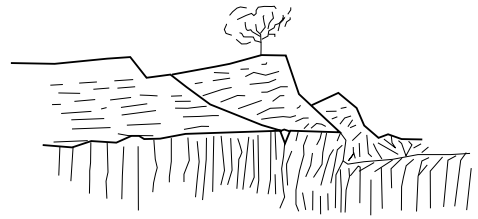
Este tipo de falla combinada en macizos rocosos se caracteriza por la flexión pseudo-continua a lo largo de grandes columnas, las cuales son divididas por numerosas uniones cruzadas (Hoek y Bray, 1981)[3] (Figura 1.3(c)). El cambio de la falla por flexión resulta en volcamiento por flexión, donde el volcamiento de las columnas resulta en este caso de la acumulación de desplazamientos en las juntas cruzadas (Hoek y Bray, 1981)[3]. Debido a la gran cantidad de pequeños movimientos, en este tipo de volteo, hay menos grietas por tensión que en el volcamiento por flexión y menos contactos eje contra cara y vacíos en el bloque de la cara frontal (Hoek y Bray, 1981)[3].

4. Modos secundarios de Volcamiento

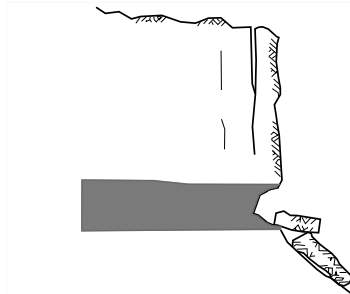
En la figura 1.4 se muestra los modos de falla por volcamiento alternativos sugeridos por (Goodman y Bray, 1976)[2]; donde como parámetro general existe la acción de un agente detonante como lo es la socavación por agentes naturales o por actividades antrópicas (Hoek y Bray, 1981)[3]. Inicialmente se presenta un deslizamiento en la parte de pata de talud, lo cual desencadena en volteo en la parte superior de los taludes (Figura 1.4(a) y 1.4(b)). En la Figura 1.4(c), se muestra un tipo común de volcamiento de areniscas sobre camas de esquisto. Este último es mucho más susceptible a la meteorización que la arenisca, mientras que ésta contiene a menudo juntas de relajación verticales. En algunos casos se ha presentado el volcamiento de grandes columnas de roca de manera intempestiva (Hoek y Bray, 1981)[3].



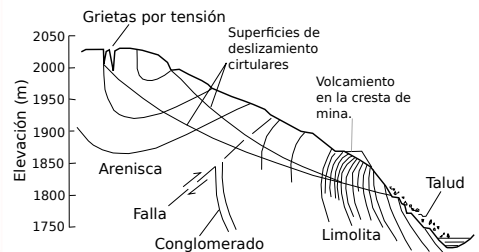
(a) volcamiento en la punta superior del deslizamiento;



(b) volcamiento en la pata del deslizamiento con movimiento cortante en la parte superior del talud (Goodman y Bray, 1976)[2];



(c) volcamiento de columnas en materiales superiores de gran resistencia mientras hay socavación en materiales de base débiles;



(d) volcamiento en la cresta de una mina resultando en una falla circular de la corona del talud (Wyllie y Munn)[8].

Figura 1.4 Modos secundarios de volcamiento. (Tomado de (Goodman y Bray, 1976)[2]).

Como modo de presentación de estos modos de falla combinados se presenta un ejemplo en la Figura 1.4(d), de volcamiento de un talud en una mina de carbón a cielo abierto (Wyllie y Munn, 1979)[8], donde la serie de bloques tiene una inmersión de 70° sobre el macizo rocoso, y su ángulo es casi paralelo a la cara del talud (Hoek y Bray, 1981)[3]. El proceso de extracción minera caracterizado por un corte de 50° inició en la falla por volcamiento en la cresta de la mina, lo cual a su vez generó una falla tipo circular, la cual se extiende a una altura de 230 m por encima de la base del sitio de volcamiento. Al realizarse el monitoreo del talud se descubrió un movimiento cercano a los 30 m en la ladera por encima de la mina, resultando en una abertura de grietas en la cresta de esta montaña, las cuales en muchos casos tenían hasta 9 m de profundidad (Hoek y Bray, 1981)[3].

El monitoreo de estos movimientos posibilitó que se procesa en el talud un movimiento y donde al final, la falla de este gran talud se controló mediante llenado de la mina. La Figura 1.5 muestra distintos tipos de falla por volcamiento los cuales dependen de la

profundidad y distribución de las discontinuidades, las cuales pueden causar fenómenos de deslizamiento precediendo finalmente al volcamiento de bloques (Sjoberg,2000)[7].

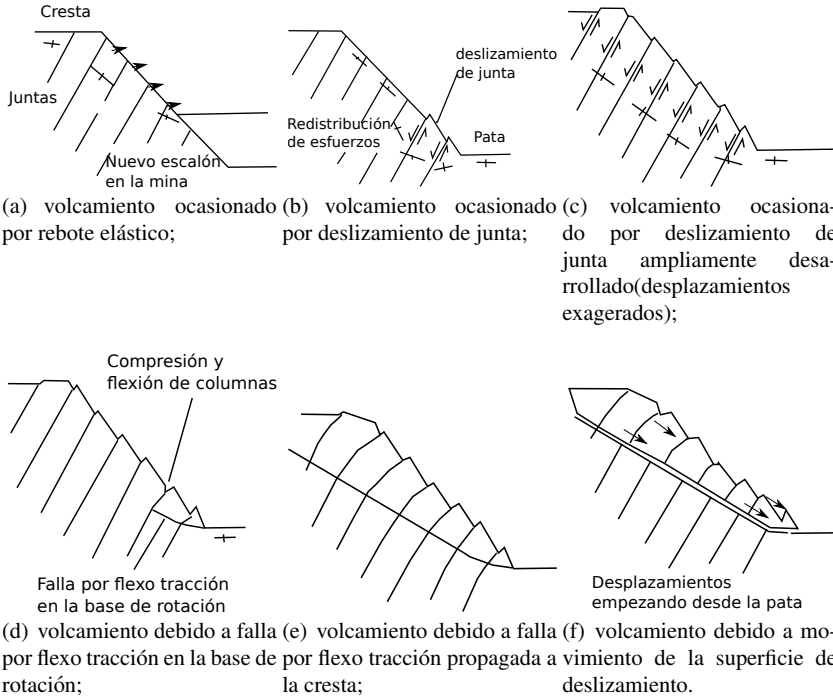


Figura 1.5 Estados de falla para volcamiento a gran escala en taludes. (Tomado de (Sjoberg, 2000)[7]).

1.3. Análisis físico general de la falla por volcamiento

La base primordial en el análisis de la falla por volcamiento es apoyado en la segunda ley de Newton, la cual posibilita analizar el estado de fuerzas actuantes y resistentes de un bloque de roca de manera individual. La posición de cada bloque individual está enmarcada por dos frentes: la pata de talud y la corona del mismo, lugar donde a su vez contiene bloques que ejercen fuerzas normales hacia el último bloque de la corona. El esquema de la Figura 1.6 corresponde a una distribución típica de bloques de roca en la falla por volcamiento.

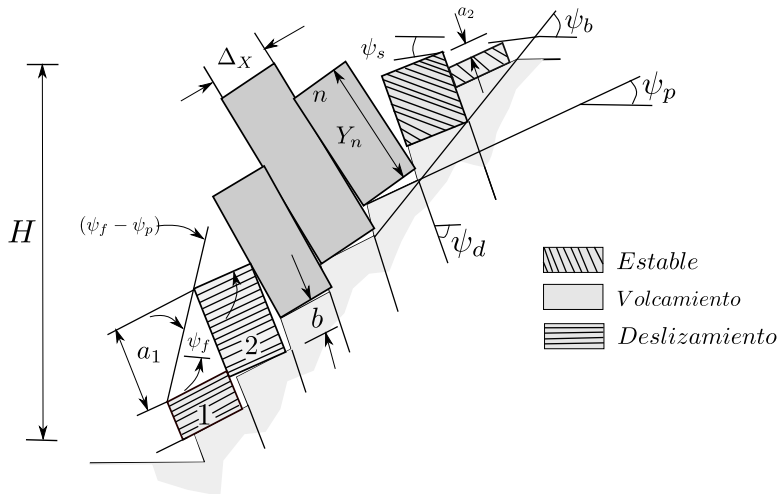


Figura 1.6 Componentes principales de la falla por volcamiento de un talud escalonado (Tomado de (Hoek y Bray, 1981)[3]).

Al tomar un bloque de este conjunto que forma el macizo rocoso, es posible mediante un análisis de cuerpo libre realizar el análisis de fuerzas que participan tanto en la estabilidad de este bloque como en su movimiento, lo cual se muestra en la Figura 1.7.

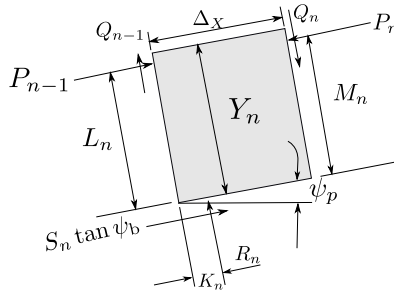


Figura 1.7 Diagrama de cuerpo libre de un bloque sometido a la falla por volcamiento; *i.e.* fuerzas actuantes en un bloque n) (Tomado de (Hoek y Bray, 1981)[3]).

Teniendo en cuenta lo anterior, las fuerzas de contacto y gravitatorias (como son el peso propio de cada bloque de roca) desencadenan movimientos de deslizamiento y de volcamiento en el macizo rocoso. Si se condiciona la posición de cada bloque de manera individual o en varios conjuntos pequeños se puede lograr un análisis (Figuras 1.8 y 1.9).

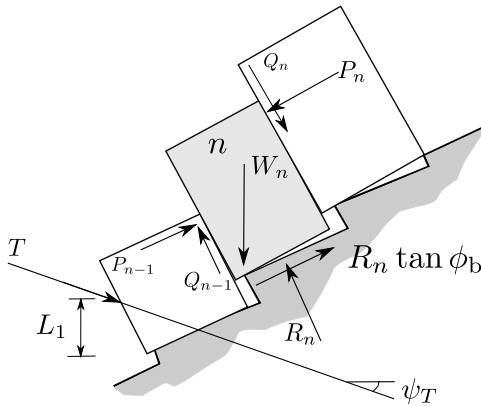


Figura 1.8 Equilibrio límite para volcamiento del bloque n (Tomado de (Hoek y Bray, 1981)[3]).

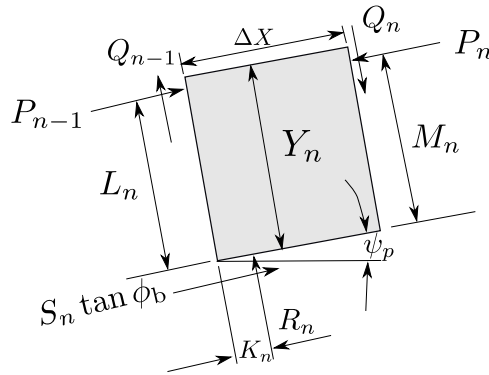


Figura 1.9 Interacción de tres bloques sucesivos en proceso de volcamiento (Tomado de (Hoek y Bray, 1981)[3]).

1.4. El programa

El código abierto está compuesto por 39 funciones que se ejecutan dentro del lenguaje de programación intérprete OCTAVE® o MATLAB®. Sin embargo, solo una función es la que se ejecuta: la `volteolab.m`, y esta va llamando a las demás de forma interna.

Para lograr el cálculo de la fuerza de soporte P el programa llama a una sencilla interfase gráfica, a partir de la cual invoca a nuevas ventanas de diálogo para lograr el objetivo y la posterior conservación de la información generada en un archivo digital no codificado.

La estructura de toda la *caja de herramientas* del programa es la que se muestra en la Figura 1.10 y 1.11. Ahí observará que existe dos carpetas con los archivos con extensión `m`. Una carpeta alberga las funciones secundarias que realizan los cálculos iniciales y primordiales, (*i.e.* `auxcode`); la siguiente se encarga de realizar los cálculos geométricos de división de bloques e ilustración del esquema de estudio, (*i.e.* `geomCode`); la carpeta que posee las funciones primordiales de cálculo es la `mainCode`; y la última se encarga de las verificaciones tanto a nivel geométrico como de funciones, (*i.e.* `verifications`). Observe también que la función principal es `volteolab.m` que es la única que está en la carpeta madre.

Adicional a las dos carpetas mencionadas se tiene dos más: `ejemplo` y `verifications`. En esas dos carpetas están las imágenes de las validaciones y los ejemplos que se puede hacer como prueba, antes de usar el programa con sus propios archivos.

```

/home/userName/matlab/volteoLab/
├── volteolab.m
├── installVolteoLabSCR.m
├── auxCode
│   ├── bloqueatcresta.m
│   ├── calculaagas.m
│   ├── calcularad.m
│   ├── deg2rad.m
│   ├── heighatblocks.m
│   ├── heightblockatcrest.m
│   ├── pesobrutoagas.m
│   ├── rad2deg.m
│   └── slendernessratio.m
├── geomCode
│   ├── adjusttriangstructure2dipcoordsys.m
│   ├── blockstructuresinunrotsystem.m
│   ├── calculatel2h2.m
│   ├── createblcsinadjustedtriangstr.m
│   ├── divideadjustedblcsinslipeside.m
│   ├── divideadjustedstructurebaseblocks.m
│   ├── dividescalenetrianblocks.m
│   ├── intersecctiontwo2Dlines.m
│   ├── obtainsecondvarfromgeom.m
│   ├── obtainslipslopestructure.m
│   ├── obtainyoflinegivenx.m
│   ├── obtusescalenetriangfromtopo.m
│   ├── plotobtusescalenetriang.m
│   └── topoffromobtusescalenetriang.m
├── examples
│   ├── allfunctions13bloques.m
│   ├── allfunctions34bloques.m
│   └── allfunctions26bloquesthesis.m
└── ⋮

```

Figura 1.10 Estructura de la carpeta volteoLab.

```
/home/username/matlab/volteoLab/
├── :
├── mainCode
│   ├── coeftransf.m
│   ├── fgeometrico.m
│   ├── fuerzacortantebase.m
│   ├── fuerzanormalbase.m
│   ├── normalstressscalarray.m
│   ├── obtainstableblochs.m
│   ├── pesogeometrico.m
│   ├── positionblockatcrest.m
│   └── stsoporte.m
├── verifications
│   ├── verificationGeomCode.m
│   │   ├── exampleOfArticleLiu.mat
│   │   ├── geometricPreCalcSCR.m
│   │   ├── geometricPreCalcSCRcomments.m
│   │   └── problemGeom.pdf
│   └── verificationMainCode.m
│       ├── calculocoefftransSCR.m
│       ├── calculovariospesosbrutoSCR.m
│       └── calculovariospesosgeomSCR.m
├── alltheProblem01SCR.m
├── alltheProblem02SCR.m
├── alltheProblem03SCR.m
├── installVolteoLabSCR.m
└── volteolab.m
```

Figura 1.11 Estructura de la carpeta `volteoLab` (cont.).

Capítulo 2

Instalación

2.1. Descarga

La descarga obtiene un archivo comprimido donde se tiene el directorio del programa. Descargue el archivo en cualquier directorio, donde `userName` es el nombre del usuario que varía de acuerdo a cada máquina o preferencia del usuario. Por ejemplo, si está usando un sistema operativo basado en Unix, por ejemplo tendrá que escribir en la terminal

```
cd /home/userName/matlab
```

Luego descomprima el archivo en esa carpeta

```
tar xvfz volteoLab-0.50.tgz
```

lo que hará que se cree una carpeta llamada **volteoLab** en la ruta especificada.

Para ingresar a la carpeta escriba por ejemplo:

```
/home/userName/matlab/volteoLab/
```

¡Anote o recuerde bien la ruta de esa carpeta!

2.2. Instalación

Ejecute el intérprete que corre las funciones. Este puede ser **OCTAVE**[®] o **MATLAB**[®]; y luego adicione la ruta de la carpeta donde está **volteoLab** a la variable `path` del lenguaje intérprete mediante la función `genpath`; por ejemplo:

```
addpath( genpath('/home/userName/matlab/volteoLab/') );
```

Vea también el manual de **MATLAB**[®] u **OCTAVE**[®] la forma del cómo se añade una ruta cualquiera de forma permanente. En **WINDOWS**[®] esto se logra con la función `pathtool`.

2.3. Requerimientos del sistema

Como requisito indispensable para el modelamiento de la falla por volcamiento en 2D del software `volteoLab` se requiere de un lenguaje intérprete adecuado que lea el código desarrollado; estos son: GNU-OCTAVE[®] con su interfase gráfica Qt-OCTAVE[®] o el programa MATLAB[®].

La máquina utilizada para la simulación debe tener dos principales tipos de software: un sistema operativo y el intérprete de un lenguaje de programación específico que se describe más abajo.

Para el caso del sistema operativo, el código abierto desarrollado puede ser corrido en muchas plataformas, entre las cuales están: WINDOWS[®] (*e.g.* XP, 7 y 8), MacOS[®] o Linux (*e.g.* DEBIAN, UBUNTU[®], RedHat[®], SUSE). Las pruebas que se hicieron para este código fueron en los siguientes sistemas operativos: i486-PC-Linux-GNU (DEBIAN 4.4.5–8), UBUNTU[®] 12.04.3 LTS y WINDOWS[®] 7.

La instalación de las librerías bajo licencia libre que usa OCTAVE[®] es más fácil bajo los sistemas operativos basados en Linux.

2.4. Ejecución del programa

Dentro del lenguaje intérprete, sea MATLAB[®] u OCTAVE[®] escriba en la consola del intérprete para el caso de un simulación de un talud en dos dimensiones, teniendo en cuenta que antes de insertar la siguiente instrucción en la ventana de comando del programa intérprete deberá insertar los parámetros de medida en este caso, de orientaciones y de discontinuidades:

```
supportForce, blcsBaseNormalForce, blcsBaseTangentForce, secondVariablesSTR =
  volteolab( slopeMeasuredSTR, slipPlaneSTR, discontSTR )
```

Capítulo 3

Primera simulación

El objetivo principal de esta modelación inicial es el de contextualizar al usuario del programa sobre la funcionalidad de los dos parámetros de interés que este programa brinda. En este caso de la falla por volcamiento, es el componente de fuerzas (primer caso, referente a la determinación de dos coeficientes a) y el geométrico (segundo caso, referente a la posición del bloque de cresta) que se tiene que resolver.

3.1. Determinación de los coeficientes a_g y a_s

Según (Liu et al., 2009)[4], del cual se deriva gran parte de las referencias bibliográficas para el planteamiento de este software, los componentes a_g y a_s para este caso corresponden a la relación de ángulos entre la inclinación de la falla de base de bloques y la inclinación del corte del talud (a_g), y la relación de ángulos entre la inclinación de la falla del talud y el ángulo de la corona del talud (a_s).

Para el caso de la ejecución de la función en el interprete compilador MATLAB® u OCTAVE®, la línea que describe lo anterior es:

```
[ ag, as ] =calculaagas( betaGr, betaSr, thetaR );
```

3.2. Posición del bloque de cresta m

Según (Liu et al., 2009)[4], el *bloque de cresta* es considerado el concepto mas importante de su teoría continua. No solo es considerado como el bloque con la altura mayor en el conjunto de análisis, si no que es la posición de bloque ubicado en el intervalo $[1, n]$, que amarra a las funciones de cálculo principales; este a su vez determina el cambio entre los

bloques de corona de talud y corte del mismo. La oración que calcula ese valor, siendo un entero, es:

```
[ bloqueCresta ] =bloqueatcresta( ag,betaB, betaSr, betaBs, as, H, t );
```

donde, H es la altura de talud calculada desde la pata hasta el cambio de dirección corte corona; y β_b es el ángulo de corte del talud con respecto a la horizontal.

3.3. Altura de todos los bloques de análisis h_i

Es importante tener en cuenta que en el análisis de la falla por volcamiento existe sectorización de zonas de estabilidad de volteo y de deslizamiento (Liu et al., 2009)[4], lo cual puede observarse en la Figura 3.1. Por esta razón es importante el cálculo de las alturas de bloques como un sistema de referencia, y de la ejecución de la parte geométrica del código(ejecutar y observar esta parte), ya que esto posibilita establecer el número exacto de bloques de cálculo para esta falla; en el caso de que exista esta limitación en los parámetros de entrada. Inicialmente el cálculo del número de bloques es una función de análisis, error y corrección hasta que el cálculo de todas las alturas arroje parámetros mayores a cero; lo cual ayudado por los compiladores permite ver la posición de cada bloque con su respectiva medida. La oración de código que hace este análisis es:

```
[ hi ] =heighatblocks( radioEsbeltz, bloqueCresta, posBloque, ag, as, t );
```

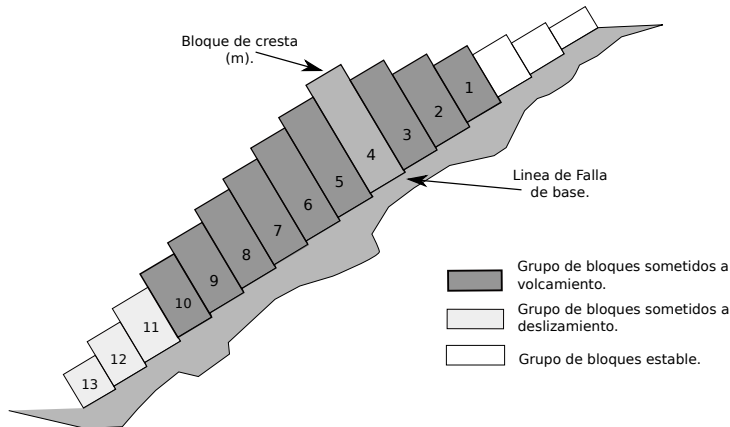


Figura 3.1 Esquema general de la falla por volcamiento en macizos rocosos (Tomado de (Liu et al., 2009)[4]).

3.4. Altura de bloque de cresta h_m

Para este caso se relaciona los factores tales como β_{sr} , β_s y a_s con H , teniendo en cuenta que es el valor puntual del bloque de todo el conjunto ubicado exactamente en la cresta del talud. Por otra parte, se tiene que en los casos de análisis el bloque ubicado en la denominada cresta de talud (Liu et al., 2009)[4] será el de mayor altura respecto a sus demás y marcará la ubicación de todos los bloques inestables (volcamiento, deslizamiento), que serán parte del análisis. En este caso la oración de código que posibilita llamar a la función la obtención de la altura del bloque de cresta es:

```
[ hm ] =heightblockatcrest( betaSr, betaS, as, H );
```

3.5. Radio de esbeltez χ

Relaciona la altura del bloque de cresta, siendo este h_h respecto al ancho unitario tomado para todos los bloques del conjunto de análisis, que en este caso figura como ancho del bloque de cresta, la obtención del valor χ se hace mediante la función:

```
[ radioEsbeltez ] =slendernessratio( hm, t );
```

3.6. Calculo de los pesos brutos W_i

En este caso los valores obtenidos son consignados en forma matricial, la cual tiene como longitud $(1 \times n)$. La dimensión n de esta matriz es la cantidad de bloques de análisis, calculada en el numeral anterior, con sus respectivos valores. El peso bruto de cada bloque por tanto se calcula con componente de gravedad de cada bloque luego de tomar en cuenta el peso unitario de los mismos y la forma de la geometría de los bloques. En este caso la oración de código que logra calcular estos pesos brutos se realiza con la siguiente función:

```
[ wi ] =pesobrutoagas( gamma, radioEsbeltez, bloqueCresta, i, t, ag, as );
```

3.7. Matriz de coeficientes de transferencia ψ_i

Esta matriz de resultados posibilita evaluar el valor nominal de la transferencia de carga entre bloques teniendo en cuenta las variables χ , m , $posBloque$, ϕ_D , a_g , a_s , β_{sr} , β_{gr} , y β_{br} . Los coeficientes de transferencia inferen sobre la inclinación con desfase del centro de

gravedad de cada bloque, donde los valores de mayores magnitudes se encuentran en la zona de bloques deslizantes ya que la mayor transferencia está en los bloques susceptibles al volteo (segundo grupo, en la Figura 3.1). En este caso la función que desarrolla esta variable será:

```
[ ct ] =coeftransf( radioEsbeltz, bloqueCresta, posBloque, phiD, ag, as, betaSr,
betaGr, betaBr );
```

3.8. Matriz de pesos geométricos o de volcamiento W'_i

Los *pesos geométricos* se explica en la referencia (Liu et al., 2009)[4] y tiene en cuenta la componente gravitatoria del peso y a su vez es relacionada con las variables geométricas tales como a_s y a_g . Los pesos geométricos posibilita inducir frente a la matriz de pesos qué bloques tienden a tener su centro de gravedad por fuera de la zona de equilibrio, originando a su vez el volcamiento y éste último a la vez desencadena en un deslizamiento de los bloques cercanos a la pata de talud. La función de cálculo para este caso es:

```
[ wip ] =pesogeometrico( gamma, t, radioEsbeltz, bloqueCresta, posBloque, ag, betaB,
betaBr, as, betaSr );
```

3.9. Matriz de fuerzas normales N_i

En la matriz de fuerzas normales se encuentran los valores de las fuerzas de contacto entre bloques (Figura 3.2), las cuales representan el contacto frente a un brazo o altura y en este caso los factores P_n y P_{n-1} equivalen a N_i y N_{i-1} . Lo interesante de esta parte es que la función que describe esta componente es acumulativa; es decir, que se obtiene la matriz acumulada de fuerzas normales. La función de cálculo en este caso corresponde a:

```
[ Narray ] =normalstressscalarray( ct, wip, nArray );
```

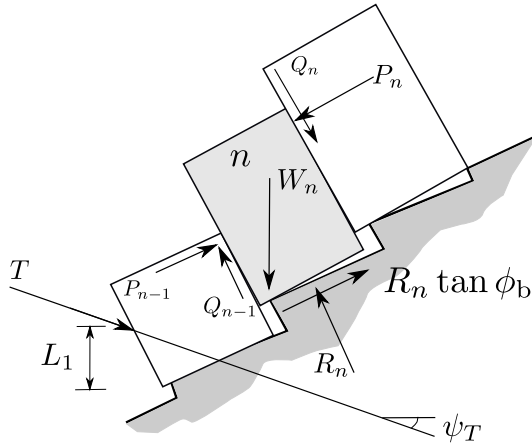


Figura 3.2 Esquematación general de las fuerzas internas en el caso de la falla por volcamiento en 2D (Tomado de (Hoek y Bray, 1981)[3]).

3.10. Factor de estabilidad al deslizamiento f_i

El factor de estabilidad al deslizamiento relaciona dentro de su formulación matemática los componentes tales como, peso normal, peso de volcamiento, coeficiente de transferencia, y fuerza normal, basados en el concepto de equilibrio límite.

El numerador de la relación del concepto del factor de seguridad donde (correspondiente a este caso a la Ecuación 3.1 (Liu et al., 2009)[4]) es la sumatoria de las fuerzas actuantes en el sistema de análisis, las cuales originan o permiten que los bloques estudiados fallen a deslizamiento; y el denominador es la sumatoria de las fuerzas resistentes, que posibilitan establecer un sistema de equilibrio. De este modo, el factor de estabilidad tiene en cuenta que un bloque o un conjunto de bloques fallan por deslizamiento y no por volteo si f_i mayor a 1. Al existir un conjunto de bloques deslizantes existirá a su vez la sumatoria de las componentes de sus pesos lo cual permitirá determinar la fuerza de anclaje que origine el equilibrio del sistema.

Para encontrar el factor de estabilidad, se usa la siguiente función:

```
[ fi ] =fgeometrico( wi, betaB, wip, ct, N, betaBr, phiD, phiB, i );
```

$$f_i = \frac{W_i \sin \beta_b - W'_i + (\psi_{i-1} - 1) N_{i-1} (\cos \beta_{br} + \tan \phi_d \sin \beta_{br})}{W_i \cos \beta_b - [W'_i + (\psi_{i-1} - 1) N_{i-1} (\tan \phi_d \cos \beta_{br} - \sin \beta_{br}) \tan \phi_b]} \geq 1 \quad (3.1)$$

3.11. Fuerza de anclaje o soporte P

A través del planteamiento anterior se induce el procedimiento de este numeral, el cual consta de determinar la componente de tensión o anclaje la cual posibilite sostener la serie fallada o deslizada y asegurar un equilibrio de fuerzas en el sistema estudiado.

La función que logra esto para MATLAB® u OCTAVE® es:

```
[ P ] =stsoporte( wi, betaB, wip, ct, N, betaBr, phiD, phiB, i);
```

3.12. Busque ayuda

Si se tiene dudas del manejo de la función `volteoLab` o si se desea ver las opciones de esta función, se puede llamar a la ayuda en cualquier momento desde la línea de comando de OCTAVE® o MATLAB® con

```
help volteoLab;
```

Capítulo 4

Opciones avanzadas

4.1. Modelado geométrico macizo rocoso

El programa `volteoLab` ofrece una gran ventaja, la cual permite intrínsecamente visualizar en gráficos y esquemas el macizo rocoso que se presente analizar para el caso de la falla por volcamiento en 2D. A su vez que otorga la posibilidad de comprobar los parámetros base, a través de una interfase de salida que se genera en la ventana de comando de MATLAB® u OCTAVE®, teniendo en cuenta que inicialmente el usuario del software pretenda modelar los dos ejemplos mostrados en el presente manual.

Las variables adicionales a tener en cuenta frente a esta reconstrucción geométrica son las siguientes y es posible observarlos en la Figura 4.1:

- l_2 longitud horizontal tomada desde la intersección entre corona y corte de talud hasta el fin de la corona de talud;
 - h_2 altura o longitud vertical tomada desde la intersección entre la línea imaginaria de corona y corte de talud hasta el fin de la corona de talud;
 - h_h altura total tomada desde la pata verdadera de talud hasta el fin de corona de talud (es la suma de altura planteada en la referencia (Liu et al., 2009)[4], con la denominada h_2).
- :

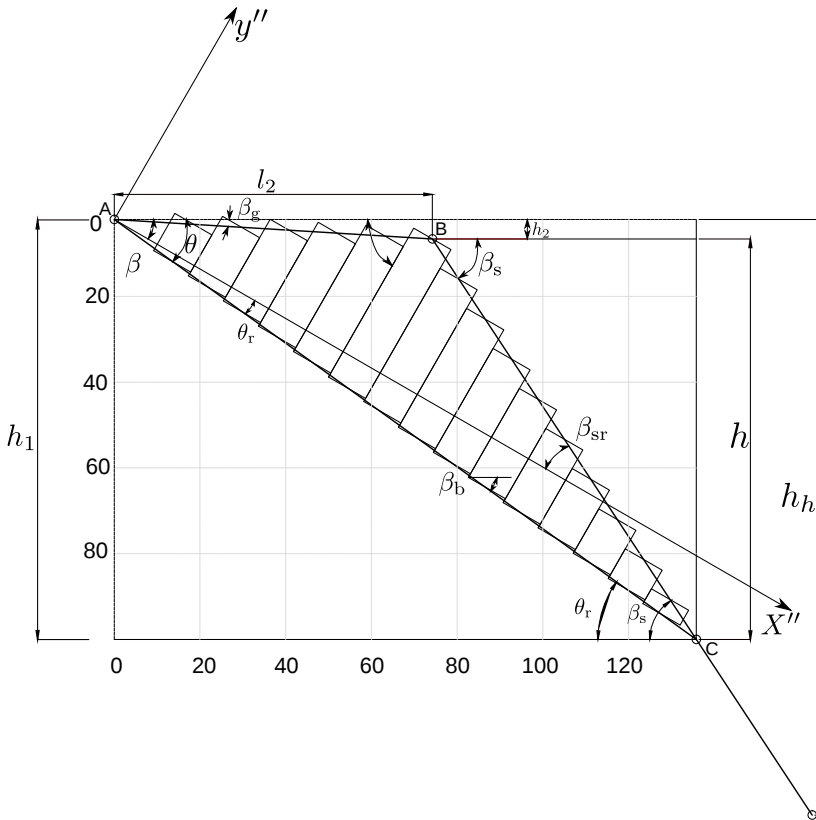


Figura 4.1 Planteamiento inicial de la parte geométrica del software volteoLab.

Es importante comentar que en la realización del planteamiento geométrico del software volteoLab se inscribe los bloques de estudio dentro del triángulo base, lo cual posibilita obtener medidas geométricas verdaderas y difieren en términos centesimales de las medidas de campo dado el caso de optar por la toma de las mismas.

Para lograr obtener la interfase geométrica de volteoLab, se realiza el siguiente procedimiento:

- las carpetas del software volteoLab tales como auxCode, geomCode, mainCode deberán estar activas;
- ejecutar en la ventana de comando de MATLAB®, la oración `geometricPreCalcSCR`.

Al realizar la anterior ejecución en la ventana de comando de MATLAB® u OCTAVE®, se obtiene la interfase gráfica que muestra la distribución de los bloques de estudio teniendo en cuenta que es una interpretación gráfica que el software realiza a las variables espaciales insertadas inicialmente (Figura 4.2).

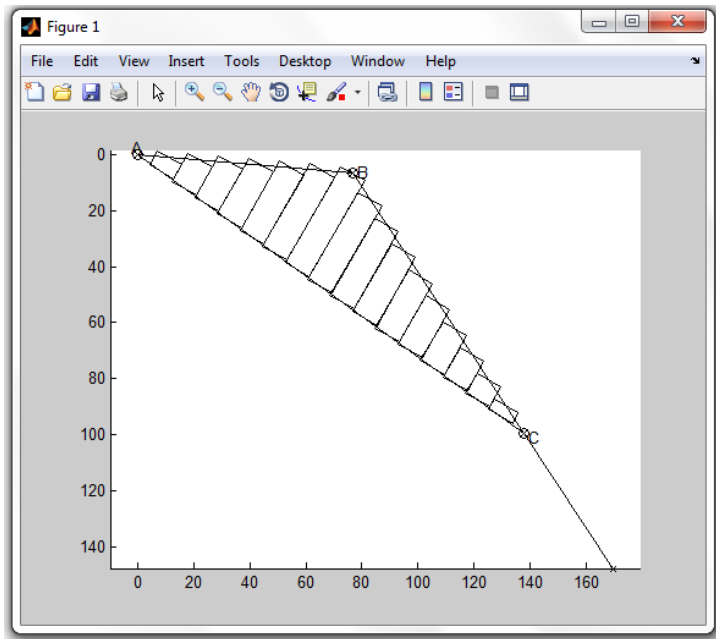


Figura 4.2 Interfase de salida geométrica del software *volteoLab*.

Posteriormente, para el caso de la presentación como formato de interfase visual de salida en la ventana de comando de MATLAB® se plantea la información relevante que maneja el software *volteoLab*, en cuanto a variables físicas, ángulos internos del talud, coeficientes geométricos a_g y a_s , y relaciones de importancia tales como el radio de esbeltez χ y la posición del bloque de cresta. Lo anterior en términos de verificación de la entrada correcta de los parámetros para el modelamiento de la falla por volcamiento en 2D, es una herramienta que posibilita tener certeza de que los datos de entrada al software son los adecuados.

Para lograr obtener la información de salida guía en este software se realiza lo siguiente.

1. Inserte las condiciones de medida del talud, condiciones físicas de las discontinuidades y las condiciones físicas del material con las siguientes líneas de código:

```
slopeMeasuredSTR =struct( 'l2', 6.90*scaleFact, 'h2', 0.6*scaleFact, 'betaSdeg',  
56.6, 'hh', 13.3*scaleFact );
```

```
% slip plane %
slipPlaneSTR =struct( 'phiBdeg', 38.15, 'thetaDeg', 35.8 );
% rock mass %
discontSTR =struct( 'dipDeg', 60, 'separation', 0.90*scaleFact, 'phiDdeg',
    38.15, 'unitWeight', 25 );
```

2. Ejecute en la ventana de comando de MATLAB® u OCTAVE®, la oración:

```
alltheProblem01SCR;
```

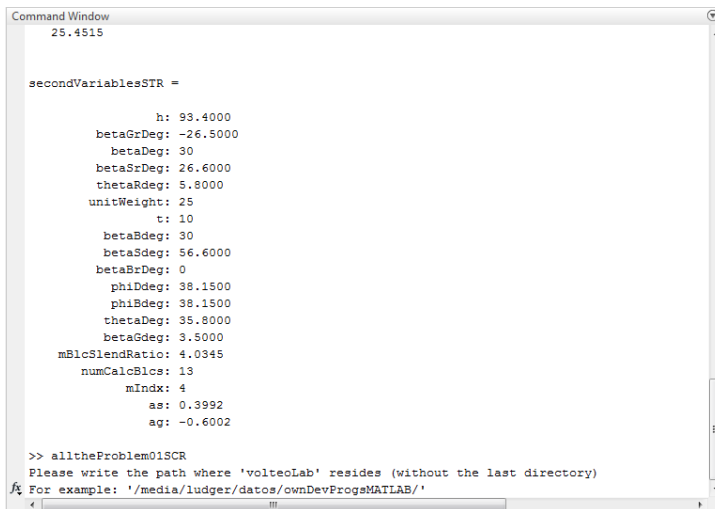
lo cual hará que la ventana de comando de MATLAB® u OCTAVE® solicite la siguiente información.

```
Please write the path where 'volteoLab' resides (without the last directory)
For example: '/media/ludger/datos/ownDevProgsMATLAB/'
```

donde el usuario deberá insertar la información de la ubicación de la carpeta destino de volteoLab, por ejemplo:

```
'\Users\Alejo\Desktop'
```

Tras ejecutar las líneas de código anteriores se visualizarán los resultados mostrados en la Figura 4.3.



```
Command Window
25.4515

secondVariablesSTR =

    h: 93.4000
    betaGrDeg: -26.5000
    betaDeg: 30
    betaSrDeg: 26.6000
    thetaRdeg: 5.8000
    unitWeight: 25
    t: 10
    betaBdeg: 30
    betaSdeg: 56.6000
    betaBrDeg: 0
    phiDdeg: 38.1500
    phiBdeg: 38.1500
    thetaDeg: 35.8000
    betaGdeg: 3.5000
    mBlcSlendRatio: 4.0345
    numCalcBlcs: 13
    mIndx: 4
    as: 0.3992
    ag: -0.6002

>> alltheProblem01SCR
Please write the path where 'volteoLab' resides (without the last directory)
For example: '/media/ludger/datos/ownDevProgsMATLAB/'
```

Figura 4.3 Interfase de salida de datos insertados en el software volteoLab.

4.2. Modifique el código a su agrado

La ventaja de tener un código libre es que usted puede mejorar y adaptar el mismo a su agrado para que él sea un complemento o una herramienta de otras aplicaciones que está desarrollando.

Si usted estudia el código de `volteoLab` podrá ver que su modificación y adaptación a sus necesidades es una tarea poco difícil. Siéntase libre de hacerlo.

Capítulo 5

Ejemplos

5.1. Cálculo de la falla por volcamiento para 13 bloques

Para efectos de este primer ejemplo, se toma el caso consignado en la referencia (Hoek y Bray, 1982)[3]. Este caso se refiere a un sistema de bloques sometidos a la falla por volcamiento el cual consta de 13 bloques, 4 de los cuales se deslizan. El problema radica en encontrar la fuerza de anclaje del sistema. La geometría del problema se muestra en la Figura 5.1 donde se esquematiza el talud con sus correspondientes medidas en ángulos y distancias, y las variables de las propiedades mecánicas del material. Las siguientes líneas código hacen el cálculo de todas las variables de importancia para este caso.

```
% alltheProblem01SCR %
% This is the main script to load and run the VolteoLab Program. %

%% % Install the toolbox volteoLab %
installVolteoLabSCR;

%% % Inputs as structures %
% Slope geometry %
h =93.4;
betaGdeg =3.5;
betaSdeg =56.6;
thetaDeg =35.8;

[ l2, h2 ] =calculate12h2( h, betaGdeg, betaSdeg, thetaDeg );

slopeMeasuredSTR =struct( 'l2', l2, 'h2', h2, 'betaSdeg', betaSdeg, ...
    'hh', 150 );
% The slip plane %
slipPlaneSTR =struct( 'phiBdeg', 38.15, 'thetaDeg', thetaDeg );
% The dominant ubiquitous discontinuity set %
betaDeg =30;
discontSTR =struct( 'dipDeg', (90-betaDeg), 'separation', 10, ...
    'phiDdeg', 38.15, 'unitWeight', 25 );

%% % Calculation of the support force %
[ supportForce, ~, ~, secondVariablesSTR ] =volteoLab( ...
    slopeMeasuredSTR, slipPlaneSTR, discontSTR );
```

```

%% % Display the result %
display( sprintf( ['The support force (paralell to the block bases) ', ...
    'applied on the slope foot necessary to make the slope stable ', ...
    'is of 5.2f force units.'], supportForce ) );
display( supportForce );
display( secondVariablesSTR );

%% % Plotting the geometry and the forces%
figure( 'Color', ones(1,3) )
hold on
set( gca, 'YDir', 'reverse' );
axis equal

```

El resultado se plasma en la Figura 5.1, que muestra la distribución de bloques para este primer caso.

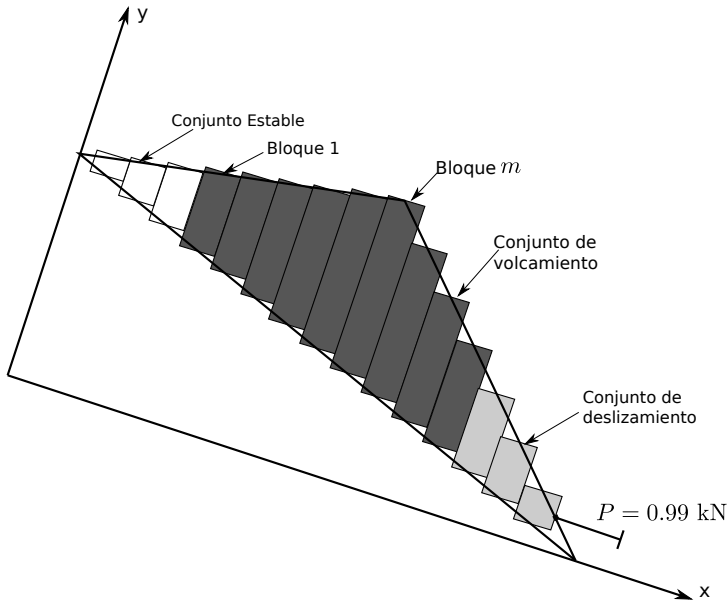


Figura 5.1 Esquematzación de análisis para la falla por volcamiento, caso 13 bloques (Tomado de (Liu et al., 2009)[4]).

De forma paralela, se muestra el Cuadro 5.2 con los resultados obtenidos, los cuales pueden ser comparados con la referencia (Liu et al., 2009)[4]. Para este caso el valor obtenido de la fuerza de anclaje fue de $P = 0.993 \text{ kN}$.

Cuadro 5.1 Resultado de las variables principales tras correr el software `volteoLab` para el caso de los 13 bloques de análisis.

S	W_i	ψ_i	W'_i	N_i	f_i
1	5499	0.541	292.2	292.2	0.690
2	6999	0.622	667.3	825.4	0.669
3	8500	0.776	1042.4	1556.0	0.66
4	10000	0.908	1620.3	2828.0	0.619
5	9002	0.897	1356.6	3924.9	0.625
6	8004	0.876	1088.8	4599.9	0.662
7	7006	0.850	814.5	4845.4	0.706
8	6008	0.811	529.9	4650.1	0.760
9	5010	0.742	226.6	3995.9	0.829
10	4012	0.596	-115.5	2849.9	0.917
11	3014	0.069	-562.7	1136.0	1.036
12	2016	4.047	-1525.9	-1447.9	1.211
13	1018	1.578	3609.2	-2250.9	1.104

5.2. Análisis de la falla por volcamiento para 34 bloques.

En este caso el ejemplo tipo fue tomado de la referencia (Liu et al., 2009)[4], teniendo en cuenta que la fuente original de información para este ejemplo es (Sagaseta, Sanchez y Cañizal, 2001)[6]. El listado siguiente muestra la secuencia de oraciones de código que hacen la solución del problema para este ejemplo.

```
% alltheProblem02SCR %
% This is the main script to load and run the VolteoLab Program. %

%% % Install the toolbox volteoLab %
installVolteoLabSCR;

%% % Inputs as structures %
% Slope geometry %
h =48;
betaGdeg =15;
betaSdeg =55;
thetaDeg =42;

[ l2, h2 ] =calculate12h2( h, betaGdeg, betaSdeg, thetaDeg );
slopeMeasuredSTR =struct( 'l2', l2, 'h2', h2, 'betaSdeg', betaSdeg, ...
    'hh', 60 );
% The slip plane %
slipPlaneSTR =struct( 'phiBdeg', 39, 'thetaDeg', thetaDeg );
% The dominant ubiquitous discontinuity set%
betaDeg =25;
discontSTR =struct( 'dipDeg', (90-betaDeg), 'separation', 2, ...
    'phiDdeg', 39, 'unitWeight', 25 );

%% % Calculation of the support force %
[ supportForce, ~, ~, secondVariablesSTR ] =volteolab( ...
    slopeMeasuredSTR, slipPlaneSTR, discontSTR );
```

```

%% % Display the result %
display( sprintf( ['The support force (paralell to the block bases) ', ...
    'applied on the slope foot necessary to make the slope stable ', ...
    'is of 5.2f force units.'], supportForce) );
display( supportForce );
display( secondVariablesSTR );

%% % Plotting the geometry and the forces %
figure( 'Color', ones(1,3) )
hold on
set( gca, 'YDir', 'reverse' );
axis equal

```

Al ejecutar el lista anterior en forma de archivo por lotes, se realiza el calculo de todas las variables mencionadas en el Sección 5.1, donde se toma en cuenta que los resultados se muestran en la pantalla de la ventana de comando. Directamente la variable que sobresale en el cálculo es la fuerza de soporte P . El resultado obtenido en pantalla, tras ejecutar el código para los 34 bloques de la referencia (Liu et al., 2009)[4], se muestra el cuadro 5.1. Para este caso el valor obtenido de la fuerza de anclaje fue de $P = -24.352$ kN.

Como adición a los resultados mostrados anteriormente, se presenta además el esquema de análisis de la falla por volcamiento para los 34 bloques de análisis de la referencia (Liu et al., 2009)[4] (Figura 5.2).

Cuadro 5.2 Resultado de las variables principales tras correr el software *volteoLab* para el caso de los 34 bloques de análisis.

S	W_i	ψ_i	W'_i	N_i	f_i
1	255.348	0.675	8.642	8.642	0.546
2	303.554	0.720	18.828	24.663	0.529
3	351.760	0.753	29.015	46.764	0.519
4	399.965	0.780	39.201	74.435	0.513
5	448.171	0.801	49.387	107.444	0.509
6	496.377	0.819	59.574	145.673	0.506
7	544.583	0.834	69.760	189.055	0.503
8	592.788	0.846	79.946	237.552	0.501
9	640.994	0.935	90.133	291.139	0.499
10	689.200	0.962	109.491	381.784	0.449
11	662.038	0.960	103.615	470.715	0.446
12	634.876	0.958	97.727	549.484	0.458
13	607.714	0.955	91.824	618.086	0.469
14	580.552	0.953	85.904	676.511	0.482
15	553.390	0.950	79.964	724.750	0.496
16	526.228	0.947	74.000	762.793	0.510
17	499.066	0.944	68.010	790.629	0.526
18	471.904	0.940	61.986	808.244	0.543
19	444.742	0.935	55.922	815.624	0.562
20	417.580	0.930	49.810	812.751	0.582
21	390.419	0.924	43.637	799.604	0.604
22	363.256	0.916	37.386	776.158	0.628
23	336.095	0.908	31.036	742.383	0.654
24	308.933	0.896	24.554	698.244	0.683
25	281.771	0.882	17.891	643.693	0.716
26	254.609	0.863	10.974	578.672	0.752
27	227.447	0.837	3.680	503.098	0.793
28	200.285	0.799	-4.206	416.860	0.840
29	173.123	0.737	-13.103	319.786	0.893
30	145.961	0.619	-23.943	211.594	0.954
31	118.799	0.314	-39.322	91.731	1.026
32	91.637	-2.449	-70.148	-41.308	1.112
33	64.475	2.138	-303.163	-202.015	1.231
34	37.313	1.489	68.390	-363.549	1.330

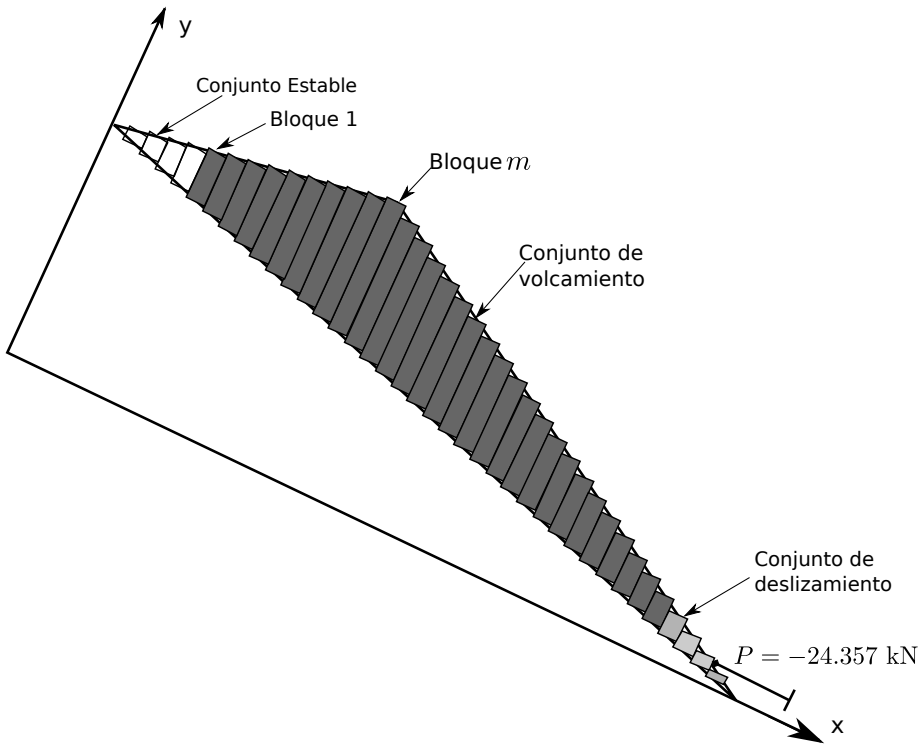


Figura 5.2 Esquema del análisis para la falla por volcamiento, caso 34 bloques (Tomado de (Liu et al., 2009)[4]).

Referencias

- [1] CUNDALL, PETER A: «A computer model for simulating progressive large scale movements in blocky rock systems». En: *Proc. Symp. Rock Fracture (ISRM), Nancy*, volumen 1, 2013.
- [2] GOODMAN, RICHARD E y BRAY, JOHN W: «Toppling of rock slopes». En: *Rock Engineering for Foundations & Slopes*, pp. 201–234. ASCE, 1976.
- [3] HOEK, EVERT y BRAY, JONATHAN D: *Rock slope engineering*. CRC Press, 1981.
- [4] LIU, CH; JAKSA, MB y MEYERS, AG: «A transfer coefficient method for rock slope toppling». *Canadian Geotechnical Journal*, 2008, **46(1)**, pp. 1–9.
- [5] MÜLLER, LEOPOLD: «New considerations on the Vajont slide». *Rock Mechanics & Engineering Geology*, 1968.
- [6] SAGASETA, C; SANCHEZ, JM y CANIZAL, J: «A general analytical solution for the required anchor force in rock slopes with toppling failure». *International Journal of Rock Mechanics and Mining Sciences*, 2001, **38(3)**, pp. 421–435.
- [7] SJOBERG, J: «Failure mechanisms for high slopes in hard rock». *Slope Stability in Surface Mining*, 2000, pp. 71–80.
- [8] WYLLIE, DC y MUNN, FJ: «The use of movement monitoring to minimize production losses due to pit slope failures». En: *Proceedings of the first international symposium on stability in coal mining. Vancouver, BC*, pp. 75–94, 1979.

Apéndice A

Anexos

A.1. Alojamiento y desarrollo del código

El desarrollo y administración del presente código estará alojado en el sitio anfitrión de proyectos informáticos denominado *MATLAB[®] Central*.

A través de este sitio se hace las respectivas descargas, aportes y peticiones de participación en el proyecto.

A.2. Licencia

El autor dirige la *Iniciativa de Software Abierto para la Ingeniería Geotecnia* del Departamento de Ingeniería Civil de la Universidad Nacional de Colombia en Medellín.

Copyright © 2015 en adelante, Universidad Nacional de Colombia.

Copyright © 2015 en adelante, Luis Alejandro Erazo Martínez & Ludger O. Suárez Burgoa.

Este código abierto es software libre: usted puede redistribuirlo y/o modificarlo bajo los términos de la Licencia BSD, ya sea la versión 2 de dicha Licencia, o (a su elección) cualquier versión posterior. Usted encontrará una copia de la Licencia BSD con el código. Caso contrario puede descargar la misma en <http://www.mathworks.com>.

A.3. Descargo de responsabilidades

El presente código computacional se distribuye con la esperanza de que sea útil, pero sin ninguna garantía; sin la garantía implícita en su comercialización o idoneidad para un propósito particular. Consulte la Licencia BSD-2 para mayores detalles.

Índice alfabético

caja de herramientas, 9

ejemplo, 23, 25

intérprete, 12

interfase gráfica, 12, 21

lenguaje, 12

simulación, 13

sistema operativo, 12