

Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán
Especialización en Geotecnia

Análisis de estabilidad de taludes.

Torres Salinas José Ernesto
No. de Cuenta 504007192



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

Introducción.	3
Capítulo I	
Análisis de estabilidad de taludes en suelos: el método de Bishop simplificado.	5
1.1 Introducción	5
1.2 El método de Bishop simplificado.	5
1.2.1 Condiciones dinámicas	8
1.3 Modificaciones al código del programa. Traducción de BASIC a FORTRAN 77.....	8
1.4 Modificaciones para encontrar el factor de seguridad mínimo.....	10
1.4.1 El método de la rejilla.....	10
1.4.2 Programando el diagrama de árbol.	11
1.4.3 Integrando el ordenamiento al programa.....	11
1.5 Una aplicación práctica. Ejemplo 1.	12
1.6 Comparación de resultados entre el programa escrito en BASIC y el escrito en FORTRAN 77. ...	17
1.7 Validación del programa.	18
Capítulo II	
Análisis de estabilidad de taludes en roca: falla plana.	20
2.1 Introducción.	20
2.2 Concepto de falla plana.....	20
2.2.1 Condiciones para que se presente una falla plana.....	20
2.2.2 Limitaciones del modelo de falla plana original.	21
2.3 El factor de seguridad.	22
2.4 Análisis de estabilidad.	22
2.4.1 Extensión del método Hoek y Bray a bloques de geometría irregular.	23
2.4.2 Método vectorial para geometrías arbitrarias.....	23
2.4.3 Obtención de las fuerzas actuantes y resistentes.....	25
2.4.4 Fuerzas hidroestáticas.....	25
2.4.4.1 Fuerza hidroestática sobre la sección sin grieta de tensión.....	25
2.4.4.2 Fuerzas hidroestática sobre la sección con grieta de tensión.	26
2.4.5 Sumatoria de fuerzas	29
2.5 Programando en FORTRAN 90.....	30
2.6 Una aplicación práctica, ejemplo 1.....	30
2.7 Validación del programa.	33
Capítulo III	
Modelo de programación, integrando programas estructurados a una interfaz gráfica.....	35
3.1 Programación estructurada.	35
3.2 Programación orientada a objetos.	35
3.3 Tcl/TK.....	35
3.4 FORTRAN.....	35
3.5 Integrando FORTRAN a Tcl/TK.....	35
3.6 El comando exec y open.....	36
3.7 Ejemplo	37
3.8 Visual Tcl (vTcl).....	39
Bibliografía	41
APÉNDICES	
A.1 Código del programa para análisis de estabilidad de taludes mediante el método de Bishop simplificado en FORTRAN.	42
B.1 Código para el programa de análisis de estabilidad de un talud en roca presentando una falla plana en FORTRAN90.	47

INTRODUCCIÓN

Contar con herramientas para el análisis de modelos geomecánicos es una parte crítica para el diseño de obras civiles. En el caso del análisis de estabilidad de taludes se requiere la ayuda de herramientas que realicen cálculos muy complejos y/o muy extensos, que analice, como es el caso del método de Bishop simplificado, una gran cantidad de combinaciones para encontrar un factor de seguridad mínimo; y con estas herramientas optimizar el diseño para dar una solución viable y económica.

Este trabajo se desarrolla con el fin de proporcionar herramientas sencillas que faciliten el análisis de estabilidad de taludes, se buscaron, estudiaron y analizaron varias posibilidades, las cuales debían cumplir ciertas características que debieran ser de utilidad y practicidad para el Ingeniero Geotecnista.

El objetivo principal es desarrollar un método que cumpla unas características específicas; el lenguaje de programación a adoptar debe ser simple y sencillo, que sea fácilmente interpretado por el Ingeniero, y que su operación sea sencilla, capturar datos, analizarlos e interpretarlos de una forma clara y ordenada.

Análisis de estabilidad de taludes en suelos y rocas.

Para el análisis de estabilidad de taludes en suelos se recurre al método de Bishop simplificado basado en el código del artículo 1. *Ramírez Piedrabuena, Alberto, Cuadernos de Posgrado, Serie B No. 1 (enero – junio) (1998) “Una aplicación de la computación en los análisis de estabilidad de taludes”* en el curso de especialización en geotecnia de la FES Acatlán, y para encontrar el factor de seguridad mínimo se utiliza el método más simple: analizar todas las combinaciones propias de la modelación, por lo que se recurrió al análisis combinatorio y se programó el algoritmo en FORTRAN, se realizaron pruebas y se analizaron 82800 combinaciones en sólo 11.5 segundos.

Para el caso de estabilidad de taludes en roca se desarrolló la aplicación para encontrar el factor de seguridad de una falla plana, el análisis geomecánico es más simple, se calculan las áreas geométricas, se propone un método vectorial para obtener áreas irregulares, las fuerzas actuantes así como las resistentes, y se obtiene el factor de seguridad.

Las características que se buscan en una aplicación para que sea útil para el Ingeniero son las siguientes:

Facilidad de programación (curva de aprendizaje corta), *portabilidad, sencillez, documentación, costo, accesibilidad.*

En la búsqueda de lenguajes de programación se estudiaron diferentes posibilidades, entre ellos: C, C++, , Ada, Ada95, COBOL, PASCAL, BASIC, y los orientados a objetos como Visual BASIC, Visual C, C++, etc.

Con base en lo anterior se concluyó que el lenguaje que representa más ventajas para este tipo de análisis es FORTRAN; las principales ventajas son:

Curva de aprendizaje corta. Este lenguaje es extensamente utilizado en Ingeniería ya sea en métodos numéricos y análisis, es el lenguaje que se enseña a nivel Licenciatura, al ser el

primer lenguaje que se enseña, el Ingeniero está familiarizado con la metodología, utilización y depuración.

Portabilidad, sencillez. FORTRAN, al ser tan extensamente utilizado se puede compilar en una gran cantidad de arquitecturas, desde la computadora del hogar más básica, estaciones de trabajo, hasta super computadoras, con diferentes tipos de procesadores, por lo que funciona bajo diferentes sistemas operativos, lo que lo hace un lenguaje muy flexible en cuanto a la plataforma, de acuerdo a la versión se puede compilar un mismo código sin necesidad de hacer cambio alguno.

Documentación, costo, accesibilidad. El proyecto GNU (grupo de herramientas, compiladores y sistema operativo de libre distribución) contiene una extensa librería de aplicaciones entre ellas gcc, g77, gfortran, (los compiladores FORTRAN) make, emacs, (herramientas generales de programación) etc. a las cuales se tiene libre acceso y sirve de plataforma para el desarrollo de aplicaciones, y al estar muy bien documentado cada aspecto del lenguaje es una excelente opción a utilizar.

Programación visual. En cuanto a la interfaz gráfica, se analizaron varias posibilidades y se llegó a la más simple y sencilla de todas: no utilizar la complejidad de la programación orientada a objetos, se busca integrar la programación estructurada de FORTRAN con un lenguaje que se encargue de crear la interfaz gráfica, ejecute la aplicación en FORTRAN, obtenga los datos a analizar y los despliegue para su análisis de una forma sencilla y comprensible para el Ingeniero. El lenguaje adoptado es Tcl/TK, un lenguaje interpretado, simple, sencillo, de curva de aprendizaje corta, multiplataforma y muy completo, encargado de interactuar con la aplicación estructurada, lo que resulta una combinación muy práctica.

El proyecto sienta las bases para el desarrollo de aplicaciones de utilidad práctica, adoptando el modelo de colaboración característico del sistema GNU/Linux de código abierto.

CAPÍTULO I

ANÁLISIS DE ESTABILIDAD DE TALUDES EN SUELOS: EL MÉTODO DE BISHOP SIMPLIFICADO.

1.1 Introducción.

Con base en el artículo 1. *Ramírez Piedrabuena, Alberto, Cuadernos de Posgrado, Serie B No. 1 (enero – junio) (1998) “Una aplicación de la computación en los análisis de estabilidad de taludes”*, se proponen varias modificaciones al programa de computadora presentado: la localización del factor de seguridad mínimo variando las superficies de ruptura, cambiando la posición del círculo de falla, el radio del círculo y graficando el factor de seguridad mínimo encontrado.

En el artículo se presenta un programa de computadora para analizar la estabilidad de un talud considerando superficies de falla circulares, para condiciones estáticas como dinámicas, basado en el Método de Bishop simplificado.

Las funciones del programa presentado en el artículo analiza un sólo círculo de falla, entrega los datos del análisis y el factor de seguridad encontrado para las coordenadas del círculo y radio dado.

En este trabajo se describe el como variar las coordenadas del centro del círculo así como el radio en forma ordenada para encontrar un factor de seguridad mínimo.

1.2 El método de Bishop Simplificado.

Mecanismos de falla.

En taludes, los mecanismos de falla que se presentan con más frecuencia son del tipo traslacional y rotacional. Los métodos de análisis más comunes se basan en considerar el equilibrio al límite o colapso inminente, donde se asume que se produce una falla como resultado de un deslizamiento a lo largo de una superficie de ruptura, en la frontera inferior de una masa de suelo en movimiento. En éstos análisis se hace variar sistemáticamente las superficies de ruptura hasta encontrar la que determine el menor factor de seguridad contra el tipo de falla considerado.

Tipos de falla en suelos.

En un suelo friccionante de longitud infinita se puede demostrar que las superficies de falla son rectas y que una familia de éstas es paralela a la superficie del talud. En arcillas las superficies de falla son más complejas, dichas superficies se asemejan a un cilindro, cuya traza con el plano en el que se calcula es un arco de circunferencia, aunque no representa exactamente al real, permite una simplificación de los cálculos.

Factor de seguridad.

Es el valor en el cual los valores de resistencia al esfuerzo cortante en la superficie de falla deben ser divididos para llevar el talud a un estado de equilibrio al límite. Si la resistencia al esfuerzo cortante de suelo se expresa en términos de esfuerzos efectivos:

$$(1) \quad \tau_f = c' + (\sigma_n - u) \tan \phi'$$

el factor de seguridad se define como:

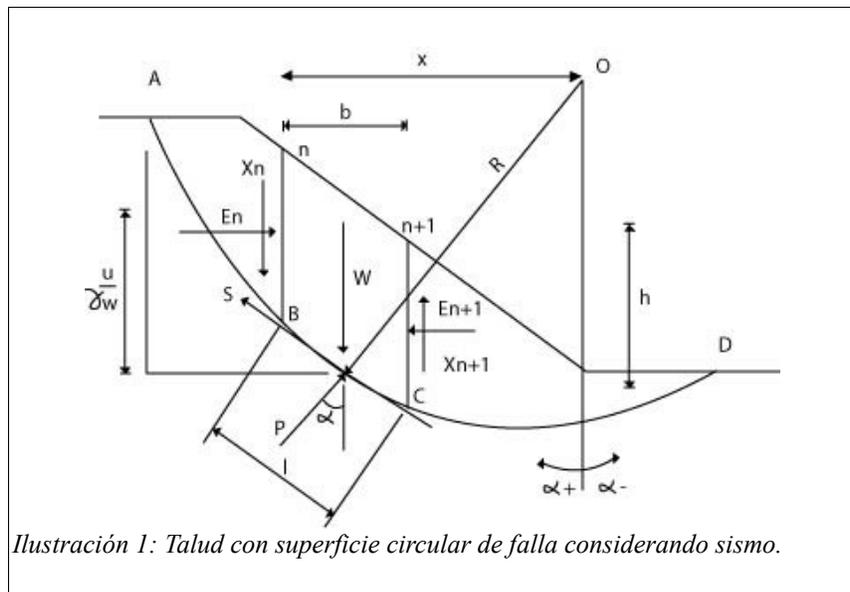
$$(2) \quad S = \frac{\tau_f}{FS} = \frac{1}{FS} [c' + (\sigma_n - u) \tan \phi']$$

S es el esfuerzo cortante en la superficie de falla que es requerido para mantener el equilibrio y FS es el factor de seguridad entre el cual se dividen los parámetros de resistencia para llevar el talud a un estado de colapso inminente.

El método de Bishop simplificado.

Para condiciones estáticas.

Se supone una sección de talud limitada por una superficie de falla circular de radio R y centro en O como en la ilustración 1, donde se considera un espesor unitario normal al papel. Para el equilibrio de la masa de suelo es necesario que el momento de las fuerzas actuantes, las que tienden a producir el deslizamiento de la masa de suelo representadas por el peso del área ABCDA, más las sobrecargas que actúan sobre la corona del talud, debe ser igual al momento de las fuerzas resistentes, representadas por la resultante de todas las fuerzas que actúan en la superficie ABCD. En ambos casos las fuerzas deberán multiplicarse por sus respectivos brazos de palanca.



Donde:

E_n, E_{n+1} resultante de las fuerzas horizontales en las secciones n y $n+1$.

X_n, X_{n+1} resultante de las fuerzas verticales cortantes.

W peso total de la dovela.

P fuerza total normal que actúa en la base de la dovela.

S fuerza cortante que actúa en la base de la dovela.

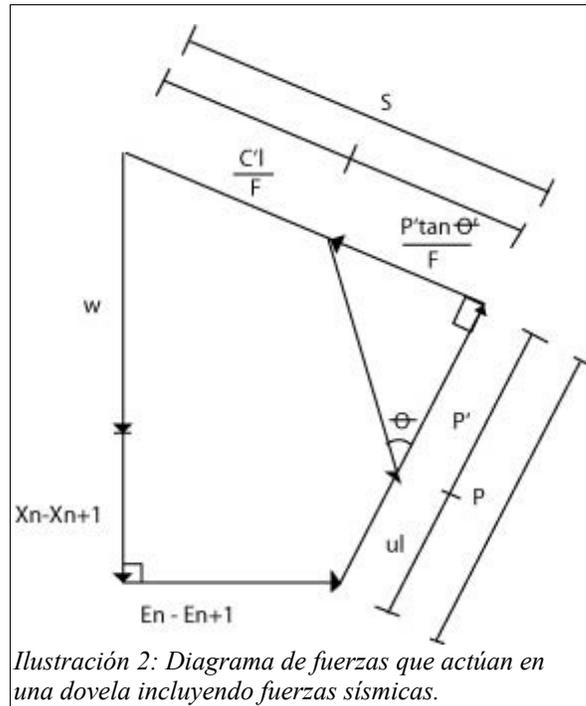
H altura de la dovela.

B ancho de la dovela.

L longitud BC.

α ángulo entre BC y la horizontal.

X distancia horizontal del centro de la dovela al centro de rotación.



De las ilustraciones anteriores se tiene que:

$$(3) \quad FS = \frac{1}{\sum \text{sen } \alpha} \cdot \Sigma [c' b + (W - ub + X_n - X_{n+1}) \tan \phi'] \cdot \frac{\text{sec } \alpha}{1 + \frac{\tan \phi' \tan \alpha}{FS}}$$

que es la ecuación conocida como el método riguroso de Bishop (3); su solución se realiza mediante aproximaciones sucesivas, simplificando se tiene que:

$$(4) \quad FS = \frac{1}{\sum \text{sen } \alpha} \cdot \Sigma [c' b + (W - ub) \tan \phi'] \cdot \frac{\text{sec } \alpha}{1 + \frac{\tan \phi' \tan \alpha}{FS}}$$

que es el método simplificado de Bishop (4).

En la ecuación (4) el factor de seguridad aparece en ambos lados de la ecuación, por lo que la solución requiere de un proceso de aproximaciones sucesivas. Una vez obtenido el factor de seguridad, se varía la posición del centro 0 y el radio para encontrar el factor de seguridad mínimo de los círculos.

1.2.1 Condiciones dinámicas.

Utilizando la expresión de Bishop simplificado se puede adaptar una aproximación pseudo-estática para analizar taludes en condiciones sísmicas. La aproximación consiste en aplicar una fuerza horizontal en el centro de gravedad de la dovela donde se realiza el análisis, donde el valor es un porcentaje de la aceleración de la gravedad inducida por el sismo. W es el peso de la dovela y K el coeficiente sísmico, el producto KW genera una fuerza cuya dirección se asocia normalmente a la horizontal; esta fuerza sólo actúa por períodos muy cortos y cambia de signo o sentido con una frecuencia que corresponde a la del sismo.

Procediendo de forma similar que para las condiciones estáticas y considerando la figura 2, para lograr el equilibrio de la masa de suelo se requiere que:

$$(5) \quad \Sigma W_x + \Sigma KWd = \Sigma SR = \Sigma sIR$$

por lo que se obtiene finalmente que:

$$(6) \quad FS = \frac{1}{\Sigma \text{sen } \alpha + \Sigma KW \left(\cos \alpha - \frac{h}{2R} \right)} \cdot \Sigma [c' b + (W - ub) \tan \phi'] \cdot \frac{\sec \alpha}{1 + \frac{\tan \phi' \tan \alpha}{FS}}$$

(6) es la ecuación de Bishop simplificado para condiciones sísmicas, definida mediante un procedimiento pseudo-estático, que se soluciona de forma similar al caso estático utilizando aproximaciones sucesivas, indicando la altura de la dovela h y el coeficiente sísmico K y variando la posición y radio del círculo.

1.3 Modificaciones al código del programa. Traducción de BASIC a FORTRAN 77.

El código fuente del programa presentado en la referencia 17. J. Cross, Civil Engineering. Oct. (1992) se tiene en lenguaje de programación BASIC, lenguaje básico y de propósito general que se ejecuta de forma interpretada.

Analizando el funcionamiento del código en BASIC se llegó a la conclusión de que la mejor forma de hacer cambios y adaptaciones es traducirlo directamente a otro lenguaje de programación adecuado para análisis numéricos. Se analizaron minuciosamente varios lenguajes de programación y se consideraron las siguientes características para la selección:

Curva de aprendizaje. Que el lenguaje sea de fácil de comprender, escribir, ejecutar, compilar y depurar.

Portabilidad. Se refiere a que sea adaptable a diferentes sistemas operativos (windows, dos, mac, unix, linux, etc.) y tipos de procesadores (x86, x64, PowerPC, Alpha, ARM, MIPS,

SPARC, etc.) sin realizar o hacer un mínimo de cambios al código fuente.

Documentación. Contar con la documentación técnica completa sobre el lenguaje.

Costo y accesibilidad. Que sea accesible ya sea para el estudiante y/o profesional.

La investigación del lenguaje de programación se llevó a cabo enteramente buscando información en internet, se investigó sobre lenguajes de programación y los que podrían ser útiles para el proyecto. Se realizó una lista sobre los lenguajes que podrían ser utilizados, se recurrió al fabricante o distribuidor, se buscó información sobre documentación y se obtuvieron compiladores para cada lenguaje, aunque en algunos casos no fue posible obtenerlo.

La segunda parte de la investigación consistió en hacer pruebas al lenguaje, para lo que se formuló un algoritmo de prueba, algoritmo que tenía que ser traducido para el lenguaje seleccionado, ejecutarlo y hacer pruebas de ejecución, mantenimiento de código, documentación, velocidad del análisis, entrada y salida de datos, si era orientado a objetos, o si se podía adaptar a una interfaz gráfica de usuario.

Los lenguajes que fueron probados son los siguientes: c, C++, Java, Ada95, COBOL, PASCAL, BASIC, FORTRAN 77 y 90, el interpretado Tcl, y el orientado a objetos Visual Basic.

La primera impresión cuando se realizaban las pruebas es que todos tienen similitudes en cuanto a resultados, tal vez la diferencia principal es la dificultad para entender el lenguaje ya que algunos son de uso general para informática y otros como FORTRAN específicos para análisis numérico.

Tras las pruebas anteriores se concluyó que el lenguaje que ofrece más ventajas para ser utilizado de forma práctica es FORTRAN. Este lenguaje es específico para análisis numérico, es de fácil comprensión, escritura, documentación, y existen un número bastante cuantioso de rutinas específicas escritas disponibles en internet, existen varios compiladores disponibles de forma libre, se puede adaptar a cualquier sistema operativo, y tipo de procesador; la documentación sobre el lenguaje es bastante extensa, específica y accesible. Además este lenguaje se enseña en las aulas y específicamente para la asignatura de métodos numéricos.

El segundo lenguaje que se tenía contemplado era el propio BASIC, pero tiene algunas fallas en cuanto a que no puede ser utilizado en algunos sistemas operativos ya que es un lenguaje interpretado y se necesita una aplicación específica para el sistema operativo y el tipo de procesador y además que en comparación resultó muy lento en comparación con FORTRAN.

Una vez seleccionado el lenguaje se procedió a hacer una traducción directa y fiel de BASIC a FORTRAN, obtenida la versión en FORTRAN se procedió a compilar y se obtuvo un ejecutable para dar paso a pruebas, en los primeros intentos se obtuvieron los mismos resultados y tras analizar algunos ejemplos, se concluyó que los resultados son los mismos.

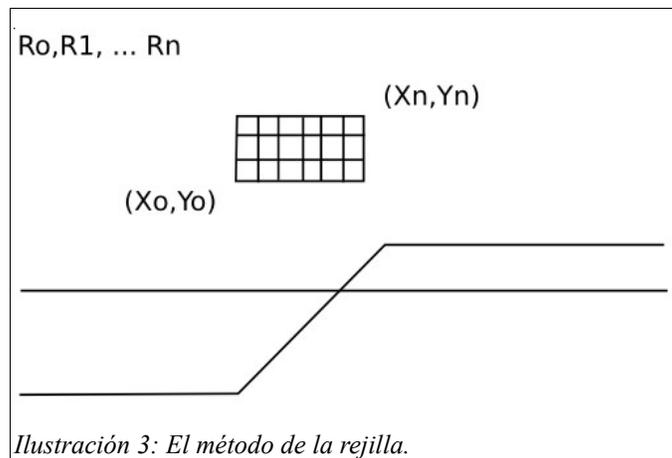
1.4 Modificaciones para encontrar el factor de seguridad mínimo.

El programa de computadora de la referencia 17. J. Cross, Civil Engineering. Oct. (1992) requiere que se lean los datos geométricos de la sección del talud, las propiedades geomecánicas de los suelos, y la posición y radio de un círculo que determinará un factor de seguridad.

El objetivo es encontrar un factor de seguridad mínimo variando la posición del círculo en el eje de las ordenadas y de las abscisas, así como también variando el radio del círculo de falla.

1.4.1 El método de la rejilla.

Este método consiste en construir una rejilla gráfica, que es una representación de las coordenadas iniciales y finales de la posición del círculo de falla a analizar, en estas coordenadas se varía la posición haciendo incrementos hasta llegar al punto final.



Del principio fundamental de cuenta: Si una cosa puede realizarse en n_1 maneras diferentes y después de esto una segunda cosa puede realizarse en n_2 maneras diferentes, ... , y finalmente una k-ésima cosa puede realizarse en n_k maneras diferentes, entonces todas las k cosas pueden realizarse en el orden especificado en n_1, n_2, \dots, n_k maneras diferentes.

La forma más simple de resolver este problema consiste en recurrir al análisis combinatorio para variar la posición del círculo así como el radio. Se tienen tres variables: R, X, y Y, se puede hacer un diagrama de árbol que representa el ordenamiento de las variables. En la ilustración 4 se muestra el diagrama de árbol.

1.4.2 Programando el diagrama de árbol.

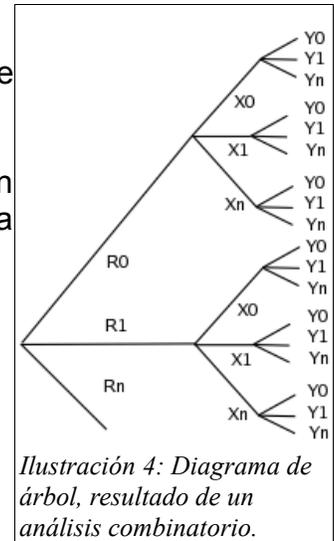
Para hacer este ordenamiento en lenguaje de programación se puede proceder de la siguiente forma:

Se requiere contar de un número inicial a un número final, con incrementos determinados, por consiguiente se puede recurrir a una sentencia DO, en FORTRAN tenemos:

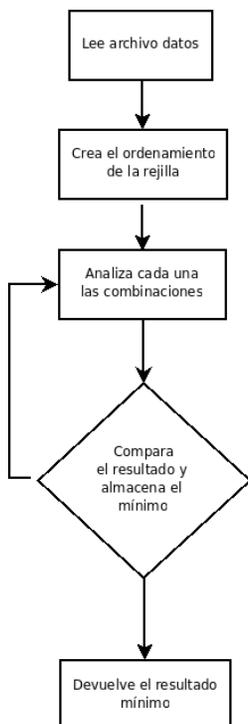
```
DO i, j, k
Instrucción
EXIT
donde i= número inicial, j=número final, k=incremento
```

así podemos anidar el conteo de R, X, Y:

```
DO R inicial R final Incremento i
  DO X inicial X final Incremento i
    DO Y inicial Y final Incremento i
      El programa analiza con X, Y y R .
    END DO
  END DO
END DO
```



1.4.3 Integrando el ordenamiento al programa.



El código del programa se divide en varios segmentos, al finalizar el análisis de los datos devuelve los resultados con un factor de seguridad, regresa y pide otro valor de las coordenadas X,Y así como otro radio R, de esta forma se puede anidar todo el código completo en el algoritmo presentado, en forma de diagrama tenemos:

Para calcular el factor de seguridad mínimo:

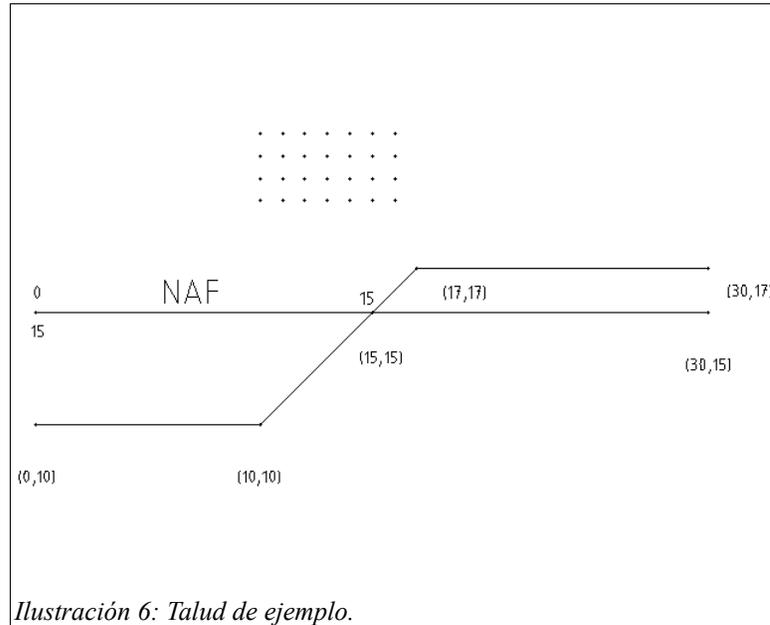
Cada vez que un arreglo se calcula se almacena el valor del factor de seguridad encontrado, se compara éste con otro valor previamente establecido que se supone más alto que el esperado y se compara, si es menor se almacena en otra variable y así sucesivamente hasta que se tiene la variable mínima del arreglo.

Ilustración 5: Diagrama de flujo que integra el ordenamiento.

1.5 Aplicación práctica. Ejemplo 1.

Para poder hacer una análisis de estabilidad con el programa se debe tener un archivo de datos. Se describirá a detalle un ejemplo de utilización basado en el ejemplo del artículo (1): Una aplicación de la computación en los análisis de estabilidad de taludes. El programa presentado al ser una traducción fiel cumple con las mismas reglas que el original.

Se tiene el siguiente talud:



1. Se necesita la geometría del talud.

Para poder realizar el análisis se realiza la representación gráfica del talud a analizar, se puede apoyar en cualquier aplicación de dibujo asistido por computadora CAD o se puede trazar la sección a mano.

El ordenamiento de los puntos que definen la sección cumple con las mismas reglas que en la aplicación del artículo (1): se dibujan todos los estratos de suelo y las superficies piezométricas, cada punto de intersección entre líneas se enumera, con la condición de que todos los puntos localizados en la superficie superior del talud sean numerados consecutivamente de izquierda a derecha, iniciando con el punto 1, los puntos bajo la superficie del talud pueden ser numerados en cualquier orden. Todos los puntos así organizados deben ser definidos con coordenadas X, Y, cuidando que toda la sección se encuentre en el primer cuadrante; las líneas se especifican asignándoles una numeración consecutiva para cada línea que se encuentre entre dos puntos, quedando definidas con un punto izquierdo, uno derecho y el número de suelo que se encuentra bajo la línea, no se pueden emplear líneas verticales pero se puede manejar asignando a las coordenadas X de los puntos de la línea una pequeña diferencia, por ejemplo 0.01. Las superficies piezométricas deben ser tratadas como interfaces de otros estratos de suelo, considerando saturación del material sobre el segmento de línea y sin saturación bajo ella. Los suelos se definen especificando su peso volumétrico, cohesión, ángulo de fricción y además indicando

si el suelo esta saturado o no (1 = si, 0=no).

Se debe proporcionar el peso volumétrico del agua, el coeficiente sísmico y la información que defina si existe una masa de agua frente a la cara del talud, que se especifica con la coordenada Y del tirante y las coordenadas X izquierda y derecha. Los datos pueden estar en cualquier sistema de unidades, siempre que exista consistencia entre ellos.

El programa puede manejar modelos incluyendo 100 puntos, 100 líneas, 20 suelos y se puede cambiar el número de dovelas cambiando el valor de S9 cambiando también el arreglo A, F y Z.

Datos del ejemplo 1.

Elevación NAF: 15

de ord: 0

a ord: 15

peso vol. Agua: 1

Factor sísmico: 0.12

Número de puntos: 6

Punto	Ord-X	Ord-Y
1	0	10
2	10	10
3	15	15
4	17	17
5	30	17
6	30	15

Número de líneas: 5

Línea	p.izq	p.der	suelo
1	1	2	2
2	2	3	2
3	3	4	1
4	4	5	1
5	3	6	2

Número de suelos: 2

Suelo peso vol. Cohes phi sat 0/1

1	2.04	1.53	30	1
2	2.04	1.53	30	0

Valor inicial de la rejilla X0,Y0 esquina inferior izquierda

10

20

Valor final de la rejilla X0,Y0 esquina superior derecha

16

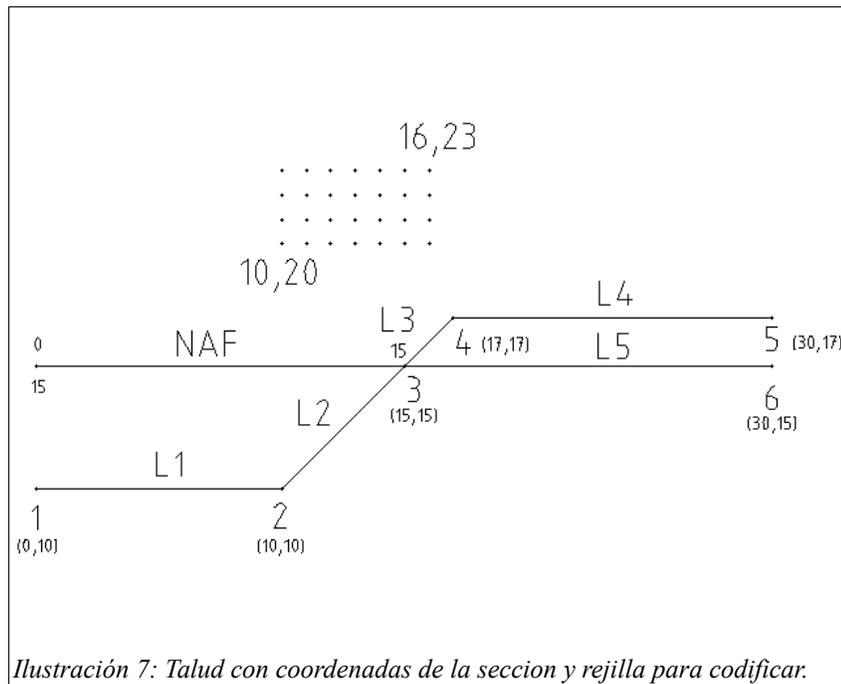
23

Valor inicial del radio de círculos a analizar R0
9.5

Valor final del radio de círculos a analizar RN
14.1

Valor de los incrementos en X,Y, y R.
0.1
0.1
0.1

Es necesario señalar que si el valor de los círculos es mayor o menor de forma tal que el círculo a analizar no intersecta el talud el programa devolverá un mensaje de error o se pueden obtener resultados aleatorios, e inclusive negativos por eso es necesario poner cuidado al seleccionar el valor mínimo y máximo de los círculos asegurándose que tanto el mayor como el menor círculo intersecta el talud de forma correcta, para esta tarea nos podemos ayudar de el programa de dibujo CAD.



Con las coordenadas seleccionadas de la rejilla y el radio inicial y final se comprueba que efectivamente en todos los puntos el círculo intersecta la sección del talud.

2. Se ejecuta el programa de análisis. Se inicia la aplicación desde la línea de comandos tenemos:

```
$/bishop.tcl
```

desde el menú archivo seleccionamos abrir, se selecciona el archivo a analizar, en este caso ejemplo.txt. Una vez abierto el archivo se dibuja la sección del talud, se puede comenzar a

analizar la estabilidad del talud con el botón ejecutar.

La velocidad del análisis depende del tipo y velocidad del procesador, y la cantidad de memoria disponible, en un primer análisis se obtienen los siguientes resultados:

Archivo: ejemplo.txt

Número de estimaciones: 82800

Tiempo de procesamiento obtenido con el comando time:

```
real 0m11.519s <== Tiempo real
user 0m4.832s
sys 0m0.024s
```



Ilustración 8: Interfaz gráfica.

Como se puede observar se tiene el menú para abrir y cerrar el archivo, dos botones uno para ejecutar la aplicación (ejecutar) y para cerrar la interface. Se observa también la disposición para obtener los resultados del análisis (Circulo mínimo) y (Estimaciones) que se obtiene calculando la combinación de posibilidades de las coordenadas de la rejilla y su incremento así como el número de círculos. Esta interfaz gráfica se creó enteramente en TK. (ver capítulo 3.)

Al terminar el análisis se muestra el círculo mínimo encontrado así como las coordenadas y el factor de seguridad mínimo encontrado.

Para hacer otro análisis basta con seleccionar en el menú el comando cerrar para limpiar la sección y el archivo de datos.

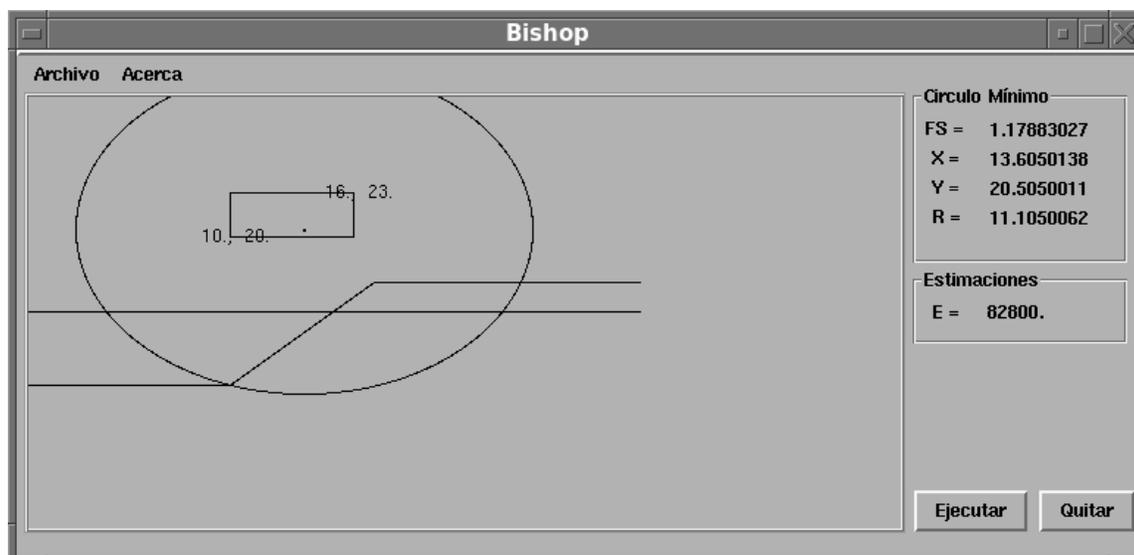


Ilustración 9: Resultado del talud de ejemplo.

Se puede comprobar el resultado recurriendo al programa original escrito en BASIC del artículo (1) y el resultado es exactamente el mismo para esas coordenadas.

Para compilar el código es necesario tener un compilador de FORTRAN 77, en este caso se utilizó el compilador G77 de GCC GNU, que es de libre distribución.

Se utiliza la versión GCC 3.4.6

Para compilarlo en la línea de comandos tenemos:

```
$g77 -o bishop bishop.f
```

con esto obtenemos el binario para poder ejecutarlo, así procedemos:

```
$/bishop
```

e introducimos el nombre del archivo a utilizar:

```
$ejemplo.txt
```

y obtenemos los resultados:

```
1.17738032 <== Factor de Seguridad Mínimo Encontrado
13.6050138 <== Coordenada X
20.5050011 <== Coordenada Y
11.1050062 <== Radio mínimo R
```

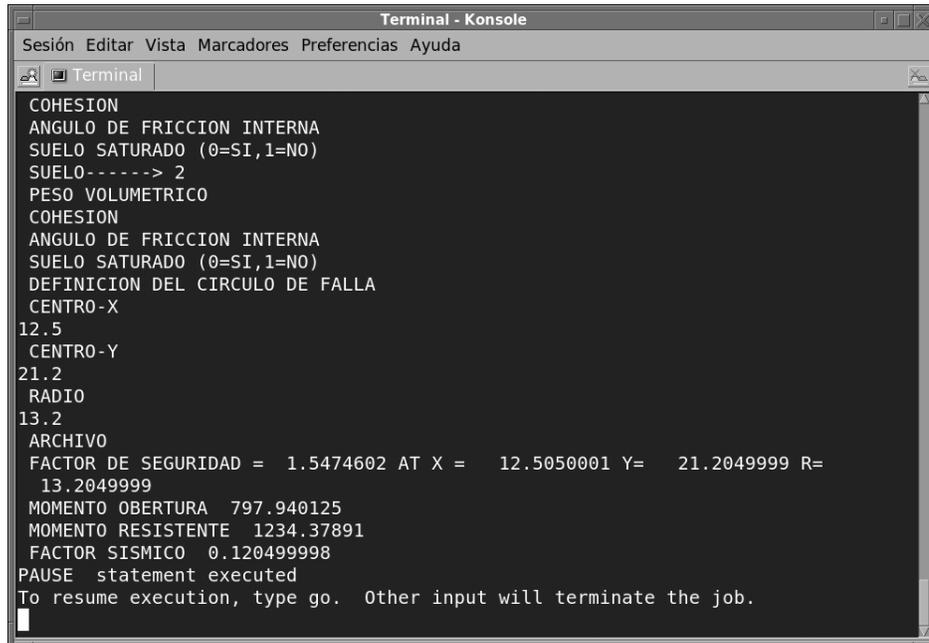
La búsqueda del factor de seguridad mínimo con este procedimiento que analiza todas las combinaciones posibles dentro de un rango establecido es la forma más simple de análisis, pero que puede llevar mucho tiempo si el análisis es muy complejo, por lo que se optó por utilizar el mejor lenguaje de programación para análisis numérico: FORTRAN, demostrando ser veloz al manejar gran cantidad de datos como es en el ejemplo analizando.

Pero sin lugar a dudas los mejores resultados para el análisis de estabilidad de taludes se

dan con una sistemática y correcta utilización de herramientas para establecer las condiciones geomecánicas, propiedades índice y mecánicas del suelo que forma el talud.

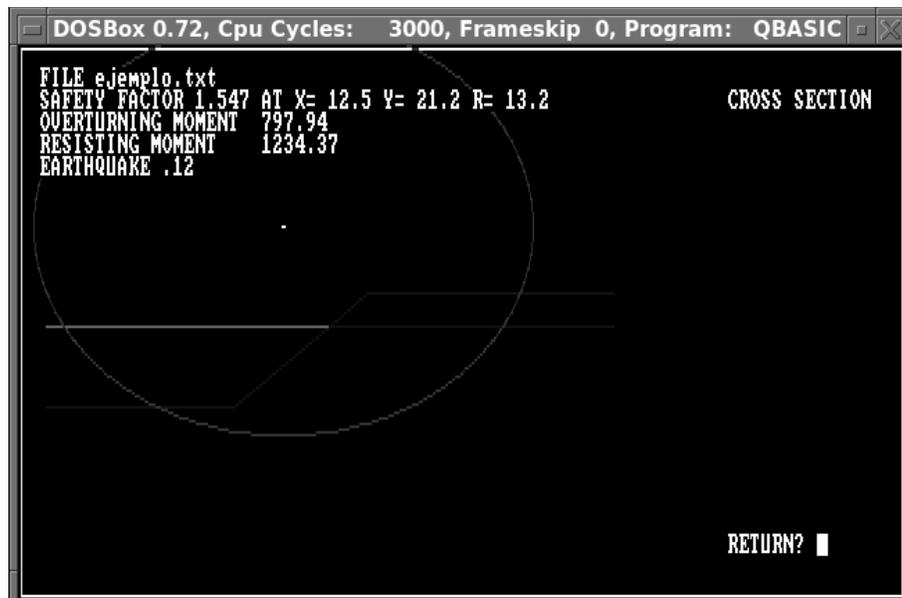
1.6 Comparación de resultados entre el programa escrito en BASIC y el escrito en FORTRAN 77.

Se realizó una traducción directa y sin cambios del código escrito en BASIC a FORTRAN, para comprobar la correcta implementación se presenta dos capturas de pantalla con los resultados del ejemplo del artículo mencionado con los resultados.



```
Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
Terminal
COHESION
ANGULO DE FRICCION INTERNA
SUELO SATURADO (0=SI,1=NO)
SUELO-----> 2
PESO VOLUMETRICO
COHESION
ANGULO DE FRICCION INTERNA
SUELO SATURADO (0=SI,1=NO)
DEFINICION DEL CIRCULO DE FALLA
CENTRO-X
12.5
CENTRO-Y
21.2
RADIO
13.2
ARCHIVO
FACTOR DE SEGURIDAD = 1.5474602 AT X = 12.5050001 Y= 21.2049999 R=
13.2049999
MOMENTO OBERTURA 797.940125
MOMENTO RESISTENTE 1234.37891
FACTOR SISMICO 0.120499998
PAUSE statement executed
To resume execution, type go. Other input will terminate the job.
```

Ilustración 10: Programa Bishop mostrando los resultados del talud de ejemplo.



```
DOSBox 0.72, Cpu Cycles: 3000, Frameskip 0, Program: QBASIC
FILE ejemplo.txt
SAFETY FACTOR 1.547 AT X= 12.5 Y= 21.2 R= 13.2 CROSS SECTION
OVERTURNING MOMENT 797.94
RESISTING MOMENT 1234.37
EARTHQUAKE .12
RETURN? █
```

Ilustración 11: Programa del artículo (1) mostrando los resultados del talud de ejemplo.

1.7 Validación del programa.

Para validar el programa se recurre a resultados de problemas de prueba, se puede comparar estos resultados con los obtenidos de literatura ya conocida.

Problema 1.

Este problema es un corte vertical de un material puramente cohesivo. En 1927 Fellenius demostró que el factor de seguridad de una falla circular para este caso es 1.00, si la altura del corte es $3.83c/W$, donde c es la cohesión del material y W es su peso volumétrico. Los datos se han modelado para que el resultado pueda ser verificado, el resultado está muy cerca del 1.00.

En este ejemplo el suelo 2 tiene una cohesión de 1.63 t/m^2 y un peso volumétrico de 1.63 t/m^3 y la altura del corte es de 3.83 m. Las propiedades del suelo 1 son irrelevantes en este caso.

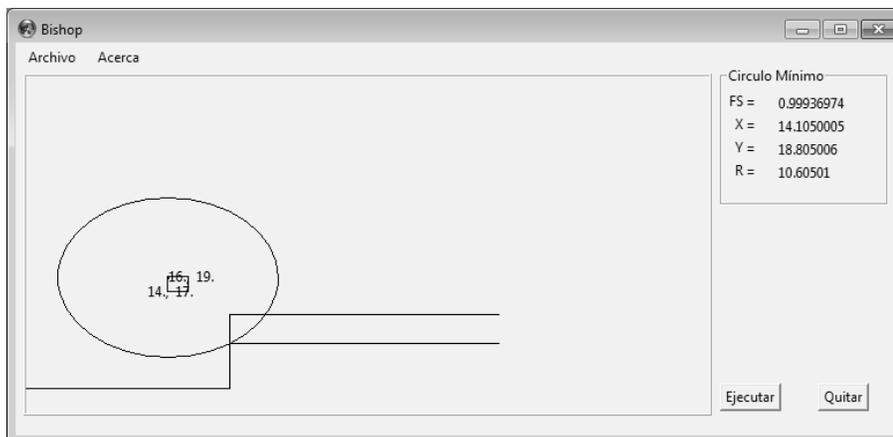
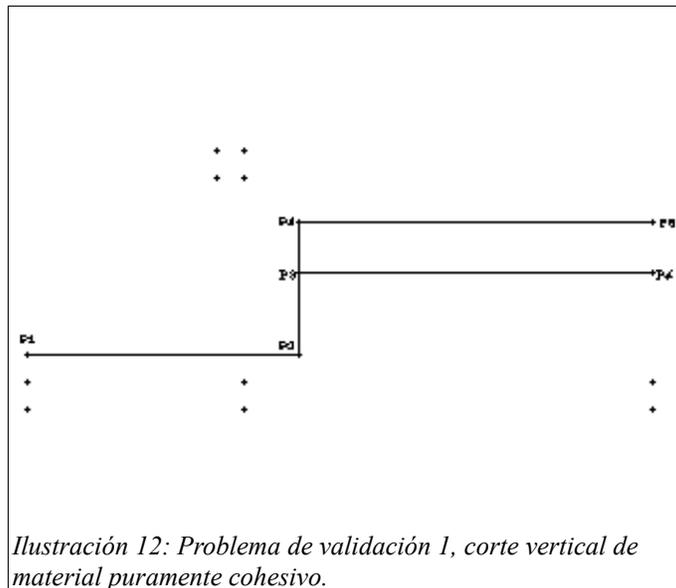
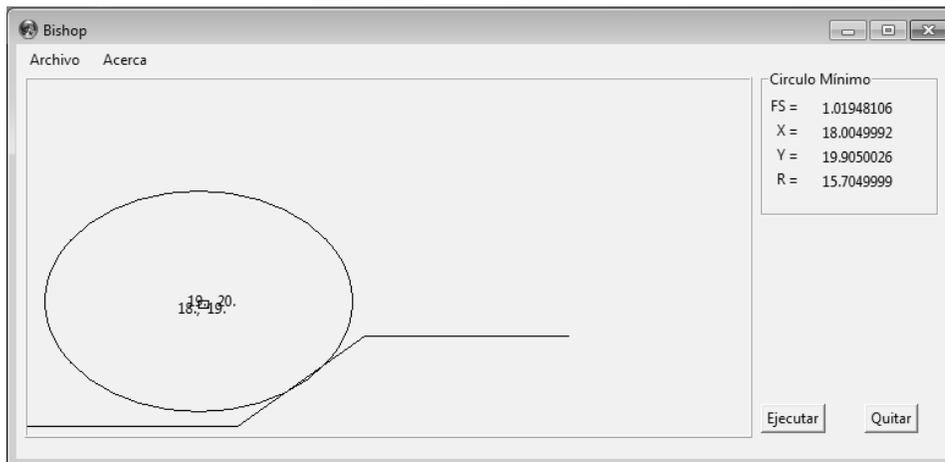
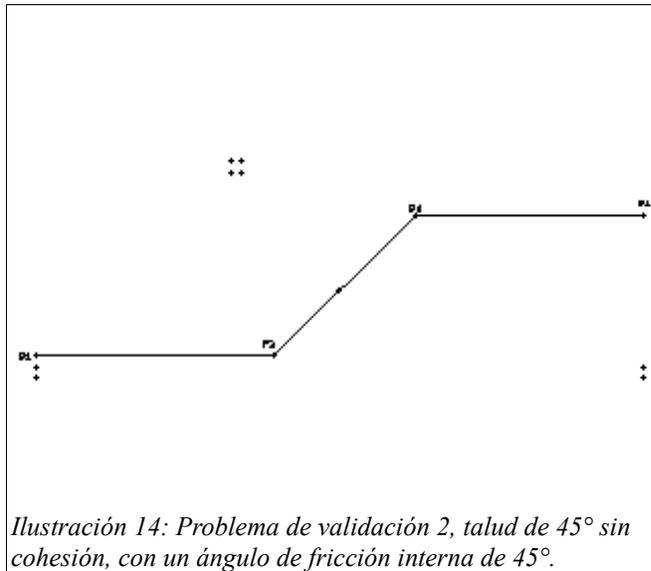


Ilustración 13: Resultados del problema que validan los resultados del programa.

Problema 2.

Este problema consta de un talud de 45° de un suelo sin cohesión con un ángulo de fricción interna de 45° . Para este talud un círculo de falla se ubicara muy cerca de la inclinación del talud, dando un valor muy cercano a 1.00, (ver Lambe y Whitman). Modelando el problema se obtienen valores muy cercanos a 1.00 .

Se puede verificar que si el nivel de agua freática satura la sección del círculo de falla también el factor de seguridad es muy cercano a 1.00.



CAPÍTULO II

ANÁLISIS DE ESTABILIDAD DE TALUDES EN ROCA: FALLA PLANA.

2.1 Introducción.

En el presente capítulo se analizará el concepto de falla plana en la estabilidad de taludes, las condiciones geomecánicas que deben presentarse para que ocurra dicha falla, y las condiciones que se asumen para su análisis.

2.2 Concepto de falla plana.

Las fallas planas ocurren a lo largo de una superficie aproximadamente plana y se analizan como un problema en dos dimensiones, sólo se tiene en cuenta el efecto de la discontinuidad principal ya que existen discontinuidades que definen los límites laterales de los movimientos, el tamaño de las fallas puede ir de unos cuantos metros cúbicos a montañas enteras.

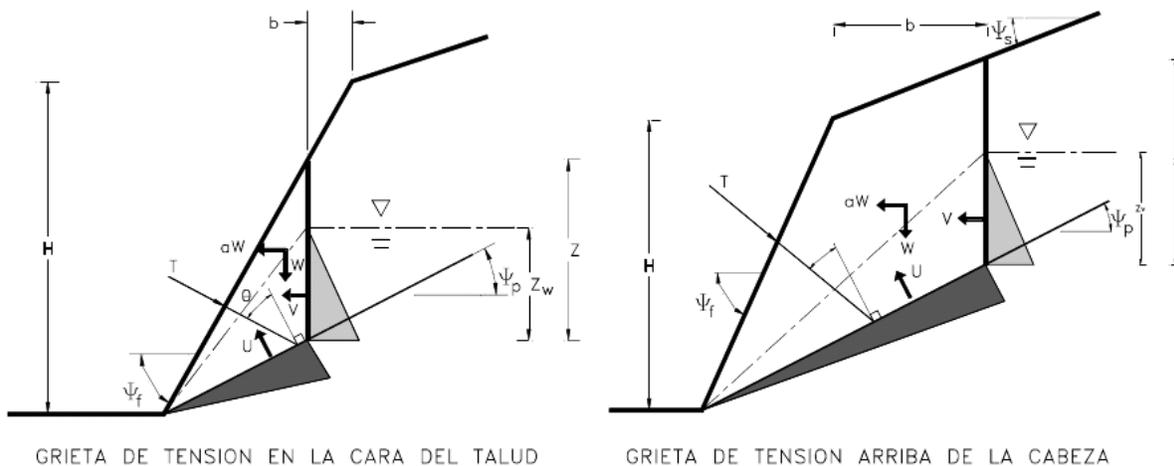


Ilustración 16: Concepto de falla plana, talud con grieta de tensión y sin grieta. 8. Suárez Díaz, Jaime. Deslizamientos y estabilidad de taludes en zonas tropicales, Ingeniería de Suelos Ltda., (1998)

2.2.1 Condiciones para que se presente una falla plana.

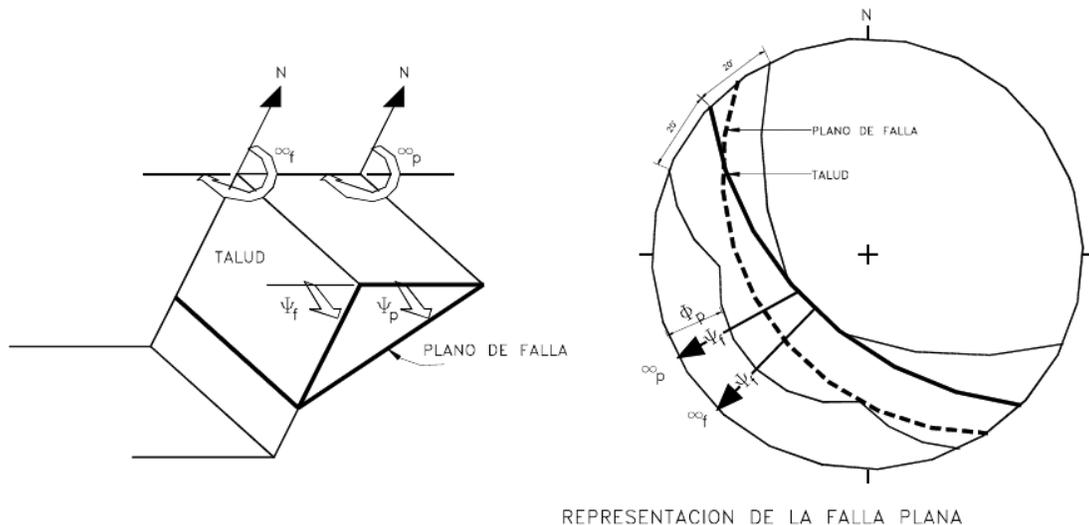
- La masa deslizante se asume como cuerpo rígido a través de la superficie deslizante.
- No ocurre ninguna rotación del cuerpo rígido.
- Todas las fuerzas que actúan sobre el bloque deslizante pasan a través del centroide.
- Se considera que el ancho es unidad.
- Como todo método de equilibrio al límite, se asume que todos los puntos a través del plano de falla están al borde de falla inminente.
- La distribución del esfuerzo a través del plano de falla es constante.

Considerando lo anterior, se tiene un bloque de roca que se desliza a través de una sola falla en una masa de roca homogénea. El mecanismo de falla y el método de análisis es por lo tanto muy similar al de el clásico deslizamiento en un plano inclinado.

El análisis cinemático cumple cuatro condiciones estructurales:

- La dirección de de la discontinuidad debe estar a menos de 20 grados de la dirección de la superficie del talud.
- El echado de la discontinuidad debe ser mayor que el echado de la superficie del talud.
- El echado de la discontinuidad debe ser mayor que su ángulo de fricción interna.
- La extensión lateral de la masa potencial de falla debe ser definida por superficies laterales que no contribuyen a la estabilidad.

Si las condiciones se cumplen la estabilidad puede evaluarse por el método del equilibrio al límite. El análisis de estabilidad requiere la solución de fuerzas perpendiculares y paralelas a la superficie de falla potencial.



REPRESENTACION DE LA FALLA PLANA

Ilustración 17: Estereografía que representa una falla plana. 8. Suárez Díaz, Jaime. Deslizamientos y estabilidad de taludes en zonas tropicales, Ingeniería de Suelos Ltda., (1998)

2.2.2 Limitaciones del modelo de falla plana original.

Todos los modelos de falla plana tienen dos limitaciones fundamentales, las cuales involucran ecuaciones de equilibrio al límite de un bloque deslizante que tiene una grieta de tensión y además se asume que la corona del talud es horizontal como se muestra en la geometría de la ilustración 18.

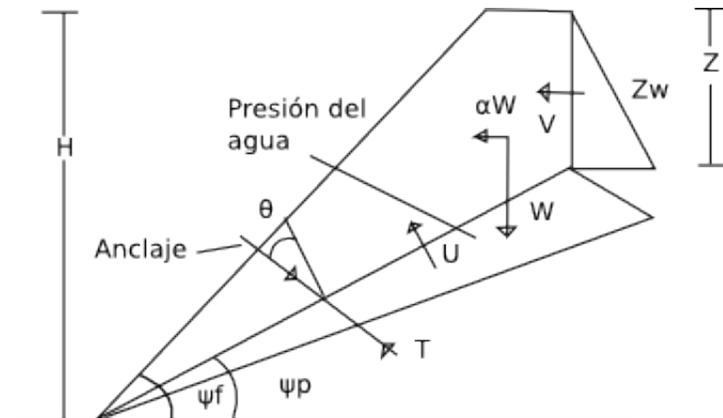


Ilustración 18: Modelo original de talud con grieta de tensión y corona horizontal.

Estas dos consideraciones restringen la generalidad de los métodos de análisis de la falla plana.

2.3 El factor de seguridad.

En el análisis de equilibrio al límite se considera que todos los puntos de deslizamiento están cerca de la falla. El factor de seguridad se define como la relación de las fuerzas resistentes a las fuerzas actuantes.

$$F = \frac{\text{Fuerzas Actuantes}}{\text{Fuerzas Resistentes}}$$

Donde las fuerzas resistentes son la resistencia al esfuerzo cortante en el plano de falla y otras fuerzas que ayudan a la estabilización de la cuña formada. Las fuerzas actuantes consisten en el peso del bloque deslizante, fuerzas producidas por la aceleración sísmica, la presión del agua que actúa sobre la masa deslizante y fuerzas externas sobre la corona del talud.

También se puede escribir el factor de seguridad como:

$$F = \frac{\text{Resistencia al esfuerzo cortante del plano de falla}}{\text{Fuerza cortante actuando en el plano de falla}}$$

2.4 Análisis de estabilidad.

La ventaja del análisis de falla plana sobre otros métodos de análisis como el análisis de dovelas verticales (Bishop, Janbu, Spencer etc.), la cuña de dos dimensiones (Sarma, Método generalizado de cuña) o el análisis tridimensional, reside en su simplicidad. Los análisis de falla plana pueden realizarse a mano utilizando las ecuaciones de Hoek y Bray; La simplicidad de los cálculos permite el análisis estadístico de riesgo, para conocer la incertidumbre de los valores de entrada y los riesgos de falla, facilitando la simulación estadística de métodos como el análisis estadístico Monte Carlo y el muestreo del Hipercubo Latino.

2.4.1 Extensión del método Hoek y Bray a bloques de geometría irregular.

Varios autores como Sarma y Frolid extendieron el método de análisis de falla plana para condiciones de grietas de tensión no verticales, y coronas de talud no horizontales. Sin embargo el problema es la complejidad de las formulaciones, que es la obtención del área del bloque de una forma arbitraria. Se propone una nueva formula que involucre la combinación de taludes de diferentes ángulos, grietas de tensión no rectas y la corona del talud no horizontal.

2.4.2 Método vectorial para geometrías arbitrarias.

Se conocen las siguientes variables: H, b, a, j, q T, O, aplicando álgebra vectorial se tiene que calcular B, C, D, y las longitudes L, Q, N, M.

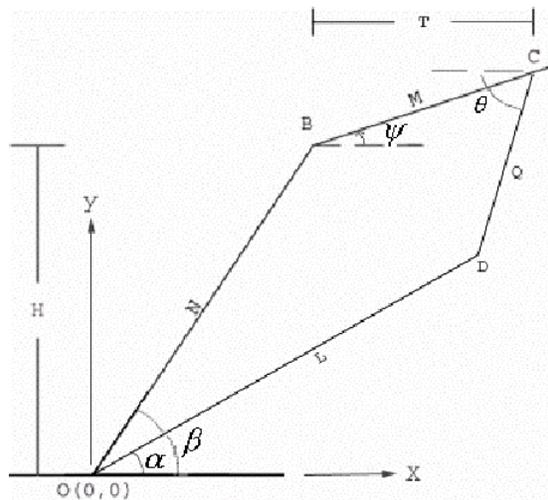


Ilustración 19: Método vectorial para análisis de geometrías arbitrarias.

Utilizando trigonometría, la longitud N y coordenadas (Bx, By) del vértice B se obtiene:

$$N = \frac{H}{(\sin \beta)} \quad (1)$$

$$B_x = H \cot \beta \quad (2)$$

$$B_y = H \quad (3)$$

Conociendo las coordenadas del punto B, y la longitud de T y el ángulo θ se puede obtener la longitud M y las coordenadas del punto C usando las siguientes relaciones:

$$M = \frac{T}{(\cos \varphi)} \quad (4)$$

$$C_x = B_x + T \quad (5)$$

$$C_y = B_y + T \tan \varphi \quad (6)$$

para obtener las longitudes L y Q, y las coordenadas D_x , D_y , se resuelve la intersección del vector de orientación θ que comienza en el punto C y un vector con la orientación α con origen O.

$$Q = \frac{(C_y \cot \alpha - C_x)}{(\sin \theta - \cot \alpha - \cos \theta)} \quad (7)$$

$$L = \frac{(C_y - Q \sin \theta)}{(\sin \alpha)} \quad (8)$$

$$D_x = L \cos \alpha \quad (9)$$

$$D_y = L \sin \alpha \quad (10)$$

una vez obtenidas las coordenadas y las distancias, tenemos toda la información requerida para definir los vectores que conectan los vértices de la sección, que puede dividirse en dos triángulos, ODB y BDC, de álgebra vectorial sabemos que el área del triángulo es la mitad del producto cruz de dos vectores con origen en cualquiera de los vértices, por lo tanto podemos obtenerla de la siguiente forma:

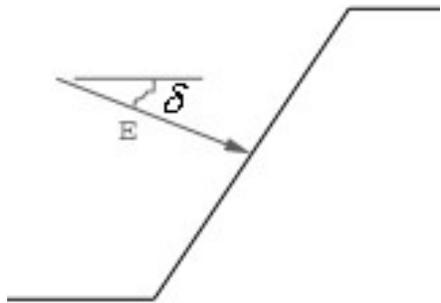
$$A = \frac{1}{2} \|B_x D_y - B_y D_x\| + \frac{1}{2} \|(D_x - B_x)(C_y - B_y) - (D_y - B_y)(C_x - B_x)\| \quad (11)$$

Obtenida el área del bloque se puede obtener el peso con el producto del peso específico de la roca γ .

2.4.3 Obtención de las fuerzas actuantes y resistentes.

Una vez obtenida la geometría de la sección, se obtienen los esfuerzos que actúan sobre la cuña formada, y las fuerzas que ayudan a la estabilidad de la sección.

Fuerza externa.



$$E_x = E \cos \delta$$

$$E_y = E \operatorname{sen} \delta$$

2.4.4 Fuerzas hidroestáticas

Los esfuerzos que produce la presencia de agua en la sección pueden actuar de diferentes maneras, y cambia con la presencia de grieta de tensión en la cuña, ya que las fuerzas que inciden dependen de la geometría, y a su vez cada caso se subdivide en condiciones distintas que dependen de la forma como se modele o incida la presión sobre la sección. Estos esfuerzos actúan de manera uniforme sobre el plano de falla o sobre la grieta de tensión.

2.4.4.1 Fuerza hidroestática sobre la sección sin grieta de tensión.

En este caso se tienen dos consideraciones:

Presión máxima al centro de la cuña, y presión máxima al pie del talud.

Caso 1: Presión máxima al centro de la cuña.

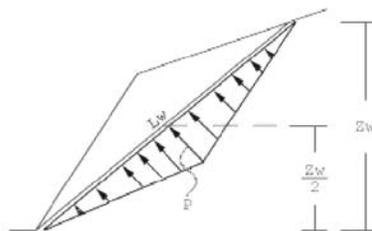


Ilustración 20: Caso 1: Presión hidroestática máxima actuando al centro de la cuña, talud sin grieta de tensión. (15).

Donde:

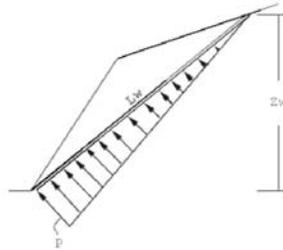
$$0 \leq Z_w \leq L \sin \alpha$$

$$L_w = \text{Longitud del \u00e1rea h\u00fameda} = \frac{Z_w}{\sin \alpha}$$

$$P = \text{Presi\u00f3n m\u00e1xima hidroest\u00e1tica} = \frac{1}{2} Z_w \gamma_w$$

$$U = \frac{Z_w^2 \gamma_w}{4 \sin \alpha} \quad (12)$$

Caso 2: Presi\u00f3n hidroest\u00e1tica m\u00e1xima al pie del talud.



Ilustraci\u00f3n 21: Caso 2: Presi\u00f3n hidroest\u00e1tica m\u00e1xima al pie del talud, talud sin grieta de tensi\u00f3n. (15).

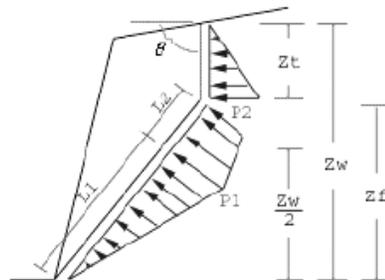
Donde: $L_w = \frac{Z_w}{\sin \alpha}$

$P = \gamma Z_w$

$$U = \frac{Z_w^2 \gamma_w}{2 \sin \alpha} \quad (13)$$

2.4.4.2 Fuerza hidroest\u00e1tica sobre la secci\u00f3n con grieta de tensi\u00f3n.

Caso 1: Presi\u00f3n hidroest\u00e1tica a la mitad de la longitud del talud.



Ilustraci\u00f3n 22: Caso 1: Presi\u00f3n hidroest\u00e1tica a la mitad de la longitud del talud con falla de tensi\u00f3n. (15)

Donde:

$$Z_t = Z_w - Z_f$$

$$Z_f = L \sin \alpha$$

U= Fuerza hidroestática en el plano de falla.

V= Fuerza hidroestática en la falla de tensión.

Se presentan tres variaciones de acuerdo a la geometría del talud:

Tipo A.

$$\text{Si } Z_w \leq Z_f$$

$$U = \frac{Z_w^2 \gamma_w}{4 \sin \alpha}; V = 0 \quad (14)$$

Tipo B.

$$\text{Si } Z_w > Z_f \text{ y } Z_w/2 < Z_f$$

$$L_1 = \frac{Z_w}{2 \operatorname{seno} \alpha}; L_2 = L - L_1; P_1 = \frac{1}{2} z_w \gamma_w; P_2 = \gamma_w Z_t; U = \frac{1}{2} P_1 L_1 + \frac{1}{2} (P_1 + P_2) L_2; V = \frac{Z_t^2 \gamma_w}{2 \operatorname{seno} \theta} \quad (15)$$

Tipo C.

$$\text{Si } Z_w > Z_f \text{ y } Z_w/2 \geq Z_f$$

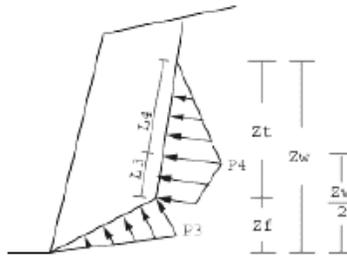


Ilustración 23: Caso 1 tipo C: presión hidroestática de un talud con falla de tensión. (15).

$$P_3 = \gamma Z_f; P_4 = \frac{1}{2} \gamma Z_w; L_3 = \frac{[\frac{Z_w}{2} - Z_f]}{\operatorname{seno} \theta}; L_4 = \frac{Z_w}{2} \operatorname{seno} \theta; U = \frac{1}{2} L P_3; V = \frac{1}{2} (P_3 + P_4) L_3 + \frac{1}{2} P_4 L_4 \quad (16)$$

Caso 2: Presión máxima al pie del talud.

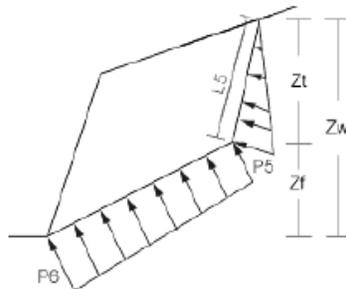


Ilustración 24: Caso 2: Presión máxima al pie del talud con grieta de tensión. (15).

$$P_5 = \gamma Z_t; P_6 = \gamma Z_w; L_5 = \frac{Z_t}{\text{seno}} \theta; U = \frac{1}{2} (P_5 + P_6) L; V = \frac{1}{2} P_5 L_5 \quad (17)$$

Caso 3: Presión máxima en la base de la grieta de tensión.

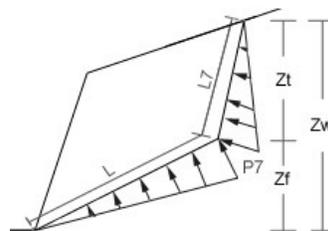


Ilustración 25: Caso 3: Presión hidroestática máxima en la base de la grieta de tensión. (15).

Donde:

$$Z_t = Z_w - Z_f$$

$$P_7 = \gamma Z_t; L_7 = \frac{Z_t}{\text{seno}} \theta; U = \frac{1}{2} P_7 L; V = \frac{1}{2} P_7 L_7 \quad (18)$$

2.4.5 Sumatoria de fuerzas.

Una vez obtenida la sección y los esfuerzos del agua si los hubiera, también hay que considerar otro tipo de fuerzas que pueden actuar: fuerza externa, fuerza sísmica, y la resistencia que presentan los anclajes tanto pasivo como activo.

Fuerzas Normales y Cortantes en el plano de falla.

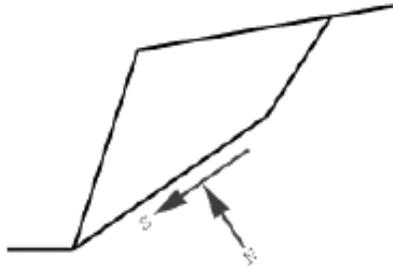


Ilustración 26: Fuerza Normal sobre el plano de falla.(15).

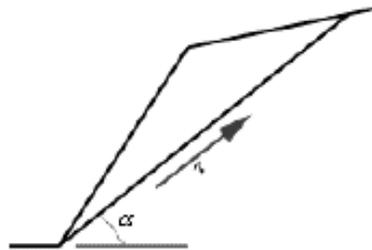


Ilustración 27: Fuerza cortante sobre el plano de falla.(15).

La fuerza normal y cortante se obtiene descomponiendo el peso de la cuña, las fuerzas que actúan sobre la sección, como la fuerza externa, la fuerza sísmica, los anclajes, la fuerza actuante del agua, obteniéndose de la siguiente forma:

$$\sum F_v + \uparrow F_y = W_y + E_y + S_y + J_y + V_y$$

$$F_y = -A\gamma - E \operatorname{sen} \delta - S \sin \Omega - J \operatorname{sen} \Delta_a + V \cos \theta \quad (19)$$

$$\sum F_x + \rightarrow F_x = E_x + S_x + J_x + V_x$$

$$F_x = E \cos \delta + S \cos \Omega + J \operatorname{sen} \Delta_a - V \operatorname{sen} \theta \quad (20)$$

$$N = -(F_y + K_y) \cos \alpha + (F_x + K_x) \operatorname{sen} \alpha - U \quad (21)$$

$$S = -F_y \operatorname{sen} \alpha - F_x \cos \alpha \quad (22)$$

donde:

W = peso de la cuña

E = Fuerza externa

δ = ángulo de incidencia de la fuerza externa

S = Fuerza sísmica

Ω = ángulo de incidencia de la fuerza sísmica

J = Fuerza anclaje activo

Δ_a = ángulo de incidencia del anclaje activo

K = Anclaje pasivo

Δ_p = ángulo de incidencia del anclaje pasivo

V = Fuerza del agua sobre la grieta de tensión

U = fuerza del agua sobre el plano de falla

Se utiliza el criterio de Mohr-Coulomb para obtener la resistencia al esfuerzo cortante definida como:

$$\tau = c + \sigma_n \tan \phi \quad (23)$$

donde c es la cohesión, σ es el esfuerzo normal actuando en el plano de falla, y ϕ es el ángulo de fricción.

Aplicando el modelo de resistencia de Mohr Coulomb el factor de seguridad es:

$$FS = \frac{(cL + W \cos \alpha \tan \phi)}{(W \sin \alpha)} \quad (24)$$

2.5 Programando en FORTRAN 90

Teniendo en cuenta lo anterior se puede programar de forma estructurada el modelo de falla plana:

1. Selección del tipo de talud: con grieta o sin grieta.
2. Introducción de la geometría del talud y propiedades geomecánicas.
3. Selección del agua presente o sin agua, y forma de actuación.
4. Presencia de fuerzas externas y ángulos de actuación.
5. Cálculo de Fuerzas actuantes.
6. Obtención de resultados.
7. Representación gráfica.

2.6 Una aplicación práctica, ejemplo 1.

Para el análisis se considera el siguiente ejemplo:

Con las siguientes propiedades:

Altura 60m
 Ángulo del talud $\alpha = 50^\circ$
 Ángulo de la discontinuidad $\beta = 35^\circ$
 Ángulo de la corona $\Psi = 0^\circ$
 Ángulo de la discontinuidad $\Theta = 0^\circ$
 Cohesión 10 t/m²
 Ángulo de fricción interna 35°
 Peso específica del agua = 1

Se considera no saturado y no actúan fuerzas externas.

Se insertan la geometría y las propiedades geomecánicas en la aplicación:

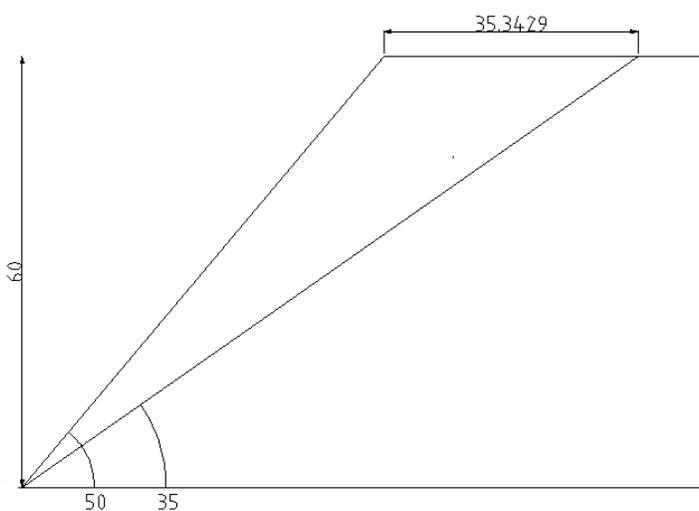


Ilustración 28: Ejemplo 1: talud presentando una falla plana.



Ilustración 29: Insertando los datos del problema de ejemplo para su análisis en la aplicación.

y se calcula el resultado:

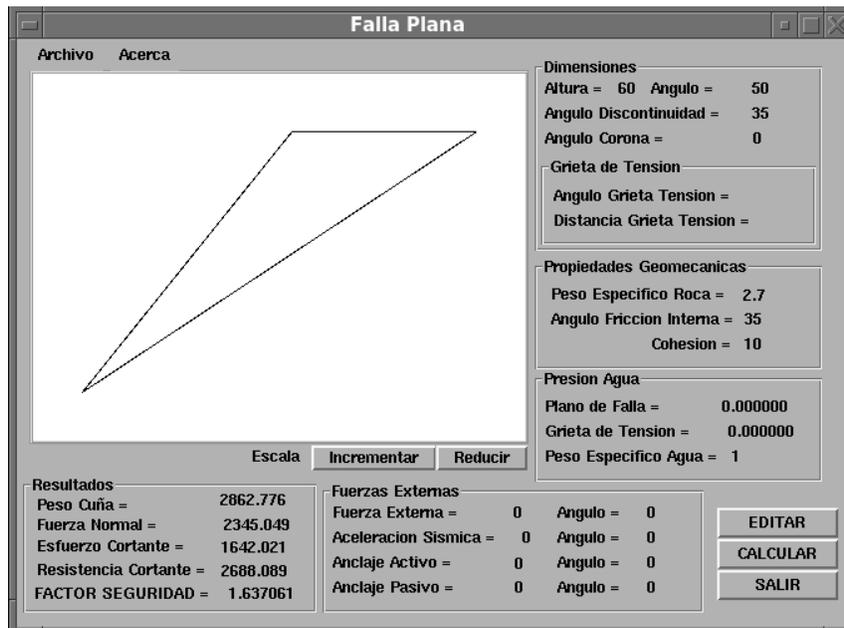


Ilustración 30: Resultado numérico y gráfico del ejemplo 1 de falla plana.

Para cotejar el resultado se recurre al programa RocPlane 2.0.2 de RocScience en su versión demostración y se obtiene el mismo resultado:

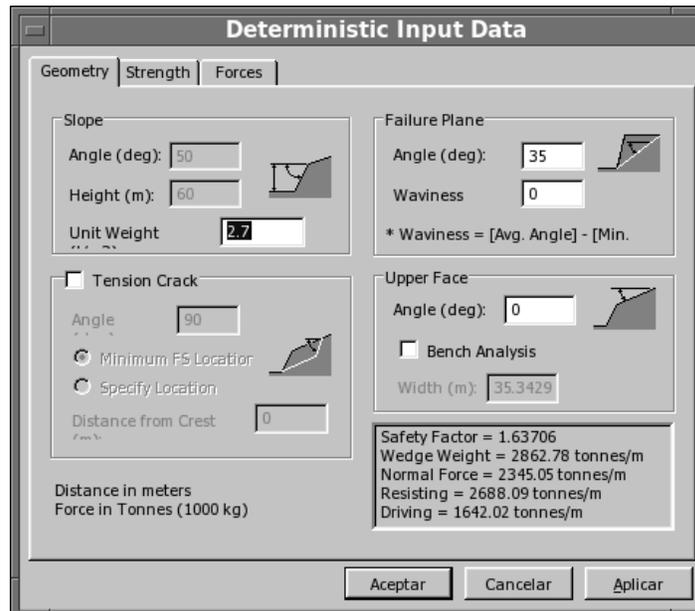


Ilustración 31: Ventana de Resultados del programa RocPlane versión 2.0.2 que valida el resultado del ejemplo 1.

Se realizaron varias combinaciones de fuerzas actuantes así como sísmicas sobre el ejemplo 1 y se obtuvieron los mismos resultados:

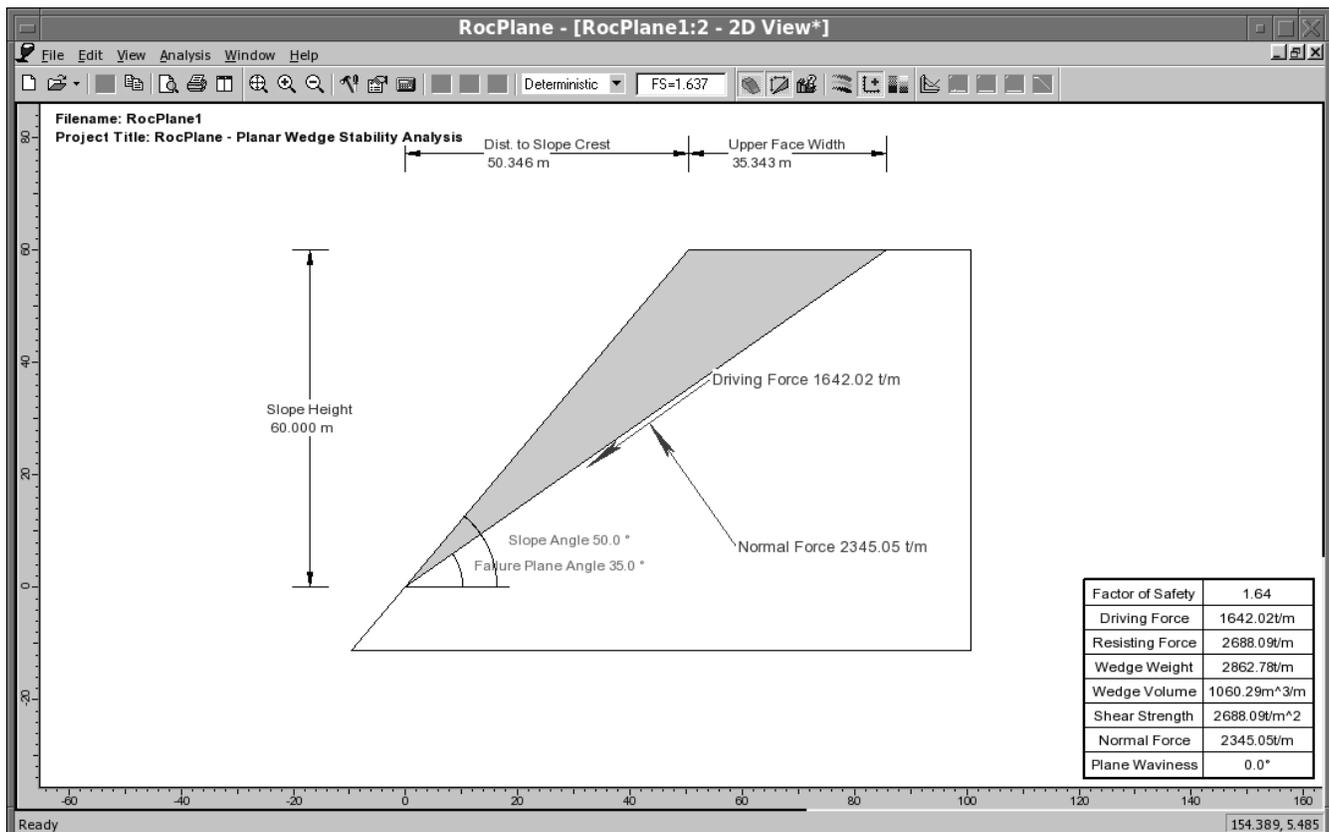


Ilustración 32: Ventana del programa RocPlane que valida los resultados obtenidos del ejemplo 1.

Tabla comparativa de resultados obtenidos con el programa RocPlane en su versión 2.0.2 contra los obtenidos con la aplicación realizada.

Geometría de la sección y condiciones	Aequilibrium Scopulus	RocPlane 2.0.2
Geometría del ejemplo 1 sin presión de agua	1.6370	1.6370
Geometría del ejemplo 1 Presión de agua máxima.	0.9679	0.9679
Geometría del ejemplo 1 Presión máxima pie talud.	0.2988	0
Geometría del ejemplo 1 con grieta de tensión $\Theta=90^\circ$, $d=10\text{m}$, $\psi=10^\circ$, sin presión de agua.	1.5634	1.5634
Geometría del ejemplo 1 con grieta de tensión. Presión de agua máxima	0.7078	0
Geometría del ejemplo 1 con grieta de tensión Presión agua máxima en la base del talud	1.0000	1.0000
Geometría del ejemplo 1 con grieta de tensión Presión máxima pie talud.	0.0000	0

2.7 Validación del programa.

Para validar el programa se recurre a problemas resueltos conocidos de la literatura. (5) Practical Rock Engineering, Hoek, 2000, págs. 92-104

Problema de verificación 1.
Se tiene el siguiente talud:

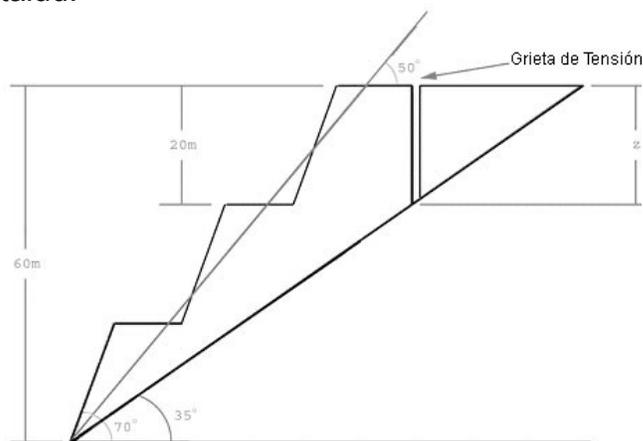


Ilustración 33: Ejemplo de validación 1.

Con las siguientes características:

Altura = 60 m

Pendiente del tauld = 50°

Pendiente de la falla = 35°

Coefficiente sísmico = 0.08 g

Peso volumétrico de la roca = 2.7 t/m³

Peso volumétrico de agua = 1 t/m³

Cohesión = 10 t/m³

Ángulo de fricción interna = 35°

Análisis sin grieta de tensión.

Se considera el talud saturado y que la fuerza máxima actúa al centro de la cuña que forma el talud.

Del ejemplo de (5) el factor de seguridad obtenido es de F.S. = 0.8184254

Iniciamos la aplicación, insertamos los datos del talud y analizamos los datos. El resultado del factor de seguridad es el mismo que el obtenido en el ejemplo.

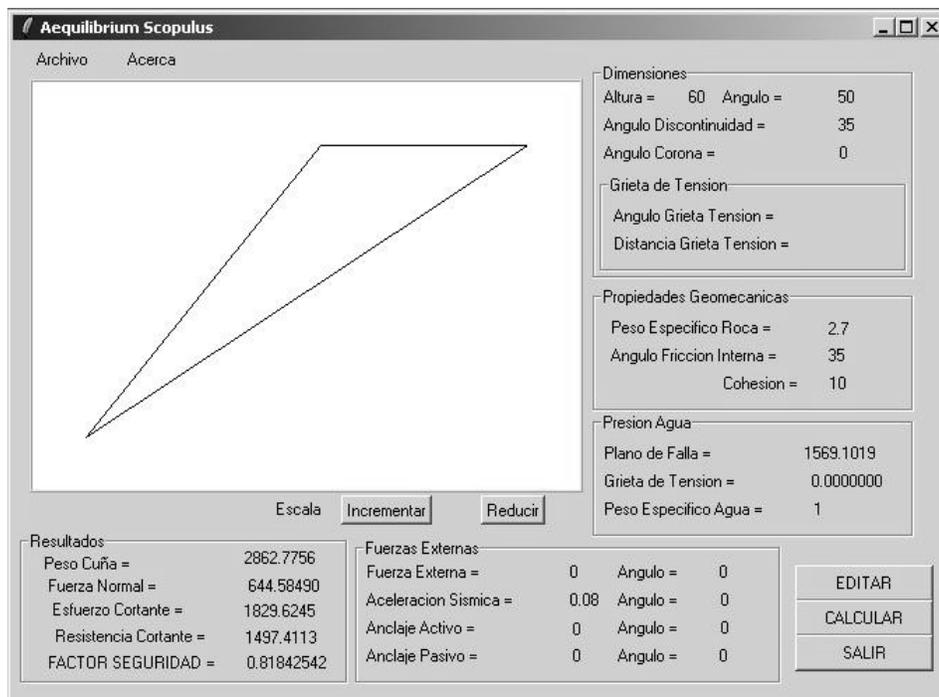


Ilustración 34: Resultado del ejemplo de validación 1: el resultado del factor de seguridad es el mismo al del ejemplo de la referencia (5).

CAPITULO III

MODELO DE PROGRAMACIÓN, INTEGRANDO PROGRAMAS ESTRUCTURADOS A UNA INTERFAZ GRÁFICA.

3.1 Programación estructurada.

La programación estructurada es una forma de programación en la cual se parte de un diseño “Top Down” que consiste en la aplicación sistemática de descomposición en subprogramas cada vez más simples, la programación estructurada se basa en lo siguiente:

1. Diseño del programa de lo general a lo particular.
2. Teorema de estructura. Todo el programa se puede diseñar utilizando solamente las estructuras básicas:

- Secuencial
- Alternativa
- Repetitiva

3.2 Programación orientada a objetos.

La programación orientada a objetos es una forma de programación que utiliza objetos, ligados mediante mensajes, para la solución de problemas. Puede considerarse como una extensión natural de la programación estructurada en un intento de potenciar los conceptos de modularidad y reutilización del código. Los mecanismos básicos de programación orientada a objetos son: objetos, mensajes, métodos y clases.

3.3 TcL/TK.

TcL es un lenguaje de programación interpretado, multiplataforma, de comandos cuya sintáxis es muy simple, y es muy utilizado para generar “scripts” en sistemas operativos tipo UNIX, genera código fácil de entender y de mantener. Se le puede añadir extensiones que forman parte del entorno.

TK es una extensión de Tcl que permite generar interfaces gráficas, una aplicación basada en TK tiene un control de flujos de eventos para poder ejecutar comandos en respuesta a estos eventos.

3.4 FORTRAN

Es un lenguaje de programación estructurado, de propósito general, enfocado a la computación numérica y científica, por lo que es eficiente en su ejecución, genera código fácil de comprender, de mantener y modificar.

3.5 Integrando FORTRAN a TcL/TK

Integrar la programación estructurada de FORTRAN y el complejo sistema de una interfaz gráfica se logra mediante Tcl y de TK, que no es más que un conjunto de comandos

que generan “widgets” y que no cumple con el paradigma de la programación orientada a objetos, al contrario, se genera estas pequeñas unidades visuales, botones, dibujos, textos, etc. Y que se comunican entre ellos mediante un comando “bind” que es el encargado de enviar las señales para la comunicación ya sea entre el intérprete Tcl, un mismo comando Tk u otra extensión, archivo o programa ejecutable externo. Así Tk se encarga de mostrar solo la interface, comunicarse con otra aplicación externa que en este caso esta programada en FORTRAN y que se encarga de los cálculos numéricos complejos.

El lenguaje estructurado requiere de la interacción del usuario para poder insertar datos y después de ejecutado se devuelve un conjunto de resultados para su interpretación. Este modelo es complementado con la interfaz gráfica, que sirve para la inserción de los datos, y una vez alimentado el programa estructurado devuelve los datos que son recogidos por la interfaz gráfica y que son desplegados ya sea en formato numérico o gráficamente para su análisis.

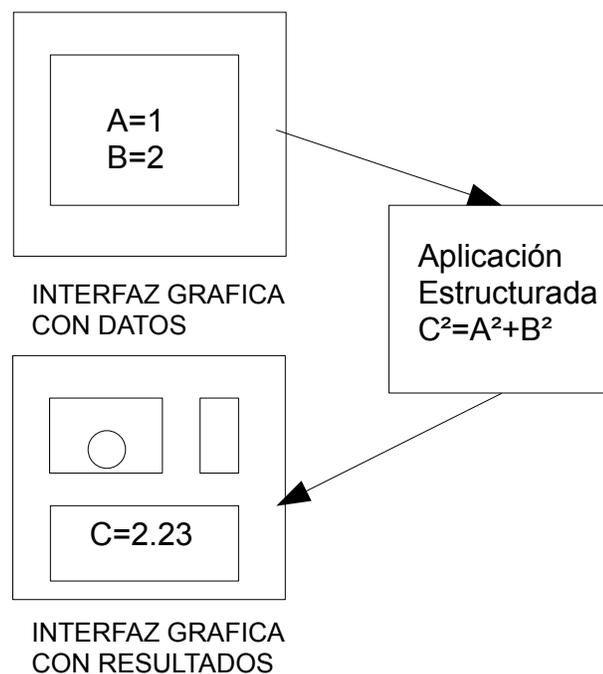


Ilustración 35: Diagrama que muestra la integración de un programa estructurado a una interfaz gráfica.

3.6 El comando *exec* y *open*.

El comando *exec* es utilizado para abrir aplicaciones desde Tcl, la salida del programa se captura mediante el comando *catch*.

El comando *open* sirve para acceder archivos y realizar operaciones de entrada y salida, una opción interesante es la capacidad de abrir una aplicación interactiva que espere recibir datos y que devuelva los datos ya analizados, como en el caso de la programación estructurada, los comandos para realizar las operaciones de entrada y salida son *puts* y *gets* respectivamente.

Ejemplo en Tcl.

Comando para abrir una aplicación con open y comunicarse con los comandos puts y gets.

```
set archivo [open ./ejemplo r+]
puts $archivo data.o
gets $archivo result.o
close $archivo
```

```
# Abre un ejecutable ./ejemplo, le asigna la variable archivo y asigna permisos de lectura y escritura
# Escribe el archivo de datos data.o para ser leído por el ejecutable.
# Obtiene el dato de salida y lo asigna a la variable result.o
# Cierra el archivo con la variable $file
```

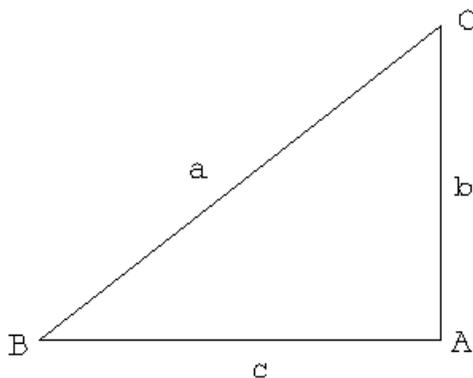
Con este procedimiento se puede tener una total interacción con los programas estructurados que reciben los datos para poder ser analizados, sin embargo cuando el programa es muy complejo se puede resolver utilizando un sólo nombre de archivo, el cual puede estar previamente formateado y listo para ser leído por el programa estructurado, o se puede utilizar la interfaz gráfica y un pequeño código en Tcl para generar estos archivos formateados, y posteriormente analizarlos, y con la salida graficar y/o analizar los resultados.

3.7 Ejemplo.

Para explicar cómo integrar una aplicación estructurada a una interfaz visual se realizará y explicará paso a paso un ejemplo sencillo.

La hipotenusa de un triángulo rectángulo.

Se tiene el siguiente triángulo rectángulo, se conocen dos catetos $b=50$ m, $c=64$ m se requiere obtener la hipotenusa.



$b=50$ m
 $c=64$ m

Del teorema de Pitágoras:

$$c^2 = b^2 + a^2 \quad (1)$$

Aplicación en FORTRAN.

Se requiere el ingreso de dos números, se realiza la operación y devuelve un sólo resultado.

hipotenusa.f90

!Análisis de un triángulo rectángulo
!Obtención de la hipotenusa dados dos catetos.
!Ejemplo capítulo 3

```
Program hipotenusa
  implicit none

  real::a,b,c
  read*,b
  read*,c
  a=SQRT((b*b) + (c*c))
  print *,a

end program hipotenusa
```

En una línea de comandos tenemos:

```
$/hipotenusa
50 <== Ingreso de b
64 <== Ingreso de c
81.21576 ==> Devuelve a
$
```

Interfaz gráfica:

Se requieren los siguientes objetos:

4 Etiquetas
CATETO *b*
CATETO *c*
HIPOTENUSA =
Cateto *a* (Invisible a la primera ejecución)

2 Cajas de entrada

Las cajas de entrada se configuran para que al ingresar un dato se le asigne la variable ingresada, en este caso la primera caja toma el valor de *b*, y la segunda *c*, en el lenguaje Tcl/TK las variables serían *\$b* y *\$c*.

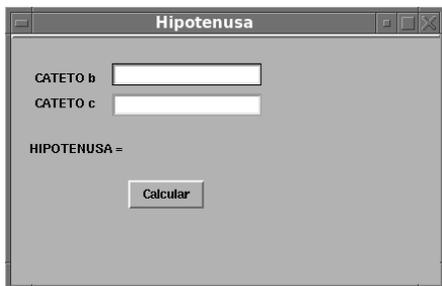


Ilustración 36: Interfaz gráfica para programar el problema de ejemplo.

1 botón

Ejecuta el “script” para llamar a la aplicación escrita en FORTRAN, introduce los valores de los catetos tomados de las cajas de entrada, obtiene el resultado de la aplicación y lo asigna a la etiqueta del cateto a (antes invisible).

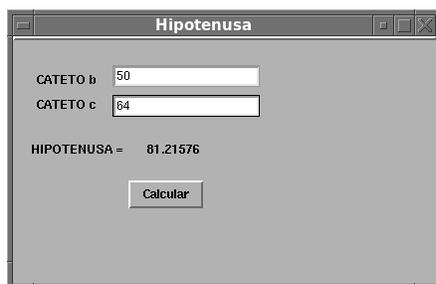


Ilustración 37: Insertando los datos y presentación del resultado.

El código del botón 1:

```

-command for .top60.but70
set hipotenusa [open "| ./hipotenusa" r+]
fconfigure $hipotenusa -buffering line
puts $hipotenusa $b
puts $hipotenusa $c
gets $hipotenusa a
close $hipotenusa

```

```

set hipotenusa [open "| ./hipotenusa" r+]
fconfigure $hipotenusa -buffering line

```

Este “script” crea el archivo *\$hipotenusa*, el comando open entre corchetes abre el programa hipotenusa escrito en FORTRAN (./hipotenusa es la forma de llamar el programa) y con los atributos *r+* de escritura y

lectura.

El comando *fconfigure \$hipotenusa -buffering line* declara que el archivo (la aplicación en FORTRAN) va a ser “interactivo” y se pueda comunicar con él hasta que este sea cerrado. (Se comunica por la salida estándar de entrada/salida I/O de la línea de comandos).

```

puts $hipotenusa $b
puts $hipotenusa $c
gets $hipotenusa a

```

Los siguientes comandos insertan las variables *\$b* y *\$c* y recoge y asigna el valor de la variable *\$a* para ser mostrada (etiqueta cateto a).

close \$hipotenusa

cierra el archivo creado.

3.8 Visual Tcl.

Visual Tcl (vTcL) es un entorno de desarrollo escrito puramente en Tcl/TK y genera código Tcl/TK por lo que el resultado es un código portable, es multiplataforma y liberado bajo la licencia GNU.

Este entorno de programación es el utilizado para desarrollar todas las aplicaciones de este trabajo.

BIBLIOGRAFÍA

1. Ramírez Piedrabuena, Alberto, Cuadernos de Posgrado, Coordinación de Producción Editorial a cargo del Dr. Arturo Mireles Hernández, (Publicación Semestral de la Coordinación de Estudios de Posgrado) Serie B No. 1 (enero – junio) (1998)
2. Verruijt, Arnold. Computational Geomechanics, Dordrecht/ Boston/London, Kluwer Academic Publishers (1995)
3. Murray R., Spiegel Ph. D. Probabilidad y estadística, México, McGraw-Hill/Interamericana de México, S.A. De C.V. (1992)
4. Johnsonbaugh, Richard. Matemáticas discretas, Belmont, California, Wadsworth Internacional/Iberomericana (1998)
5. Hoek, E. Bray, J.W. Rock Slope Engineering, The Institute of Mining and Metallurgy, (1981)
6. Sharma, S. Raghuvanshi, T.K. Anbalagan, R. Plane Failure Analysis of Rock Slopes, Artículo, Geotechnical and Geological Engineering. (1995)
7. Frolidi, P. Some developments to Hoek & Bray's Formula for the assessment of the stability in case of plane failure, Bulletin of the International Association of Engineering Geology (1996)
8. Suárez Díaz, Jaime. Deslizamientos y estabilidad de taludes en zonas tropicales, Ingeniería de Suelos Ltda., (1998)
9. Deitel, H.M. Deitel P.J. How to Program C++, Prentice-Hall (1998)
10. Coleman, F.T. Van Loan, Handbook of matrix computations, Society for Industrial Mathematics (1987)
11. Hermans, Miguel. Parallel programming in Fortran 95 using openMP, School of Aeronautical Engineering, Spain (1992)
12. Decyck, Viktor K. How to express C++ concepts in FORTRAN 90, Department of Physics and Astronomy, University of California, CA
13. Felleisen, Bruce Findler, Flatt, Krishnamuruthi, How to design programs, An Introduction to Computing and Programming, The MIT Press, Cambridge Massachussets, London England. (2001).
14. PRESS, TEUKOLSKY, VETTERLING, FLANNERY, Numerical Recipes in FORTRAN 90, The Art of Parallel Scientific, Computing Cambridge University Press, (1996).
15. Rocscience Inc. Theory Manual, RocPlane Plannar sliding analysis for rock slopes. Manual del usuario del programa RocPlane 2.0.2. (2001).

16. Rocscience inc. Verification Manual, RocPlane Plannar sliding analysis for rock slopes. Manual del usuario del programa RocPlane 2.0.2. 2001

17. J. Cross, Civil Engeneering. Oct. (1992)

Tutorial de Tcl/TK <http://www6.uniovi.es/tcl/tutorial/>

Manual de Tcl/TK <http://www.tcl.tk/software/tcltk/>

Colección de compiladores gnu gcc <http://gcc.gnu.org/>

Manual de FOTRAN 95 gfortran <http://gcc.gnu.org/fortran/>

Manual de FORTRAN 77 g77 <http://www.gnu.org/software/fortran/fortran.html>

Página del proyecto GNU <http://www.gnu.org>

Página de visual Tcl <http://vtcl.sourceforge.net/>

Página del equipo de desarrollo de Tcl/TK <http://wiki.tcl.tk/>

Página del proyecto <http://www.paginasprodigy.com/josets/>

APÉNDICES

A.1 Programa de Bishop simplificado, modificado y traducido del artículo: Una aplicación de la computación en los análisis de estabilidad de taludes. FORTRAN 77.

```
PROGRAM BISHOP
IMPLICIT NONE
```

```
C ANALISIS DE ESTABILIDAD DE TALUDES POR EL METODO DE BISHOP SIMPLIFICADO.
C COPYRIGHT 2012 JOSE E. TORRES, JOSENERNESTO@PRODIGY.NET.MX
C SE DISTRIBUYE BAJO LOS TERMINOS DE LA GENERAL PUBLIC LICENSE DE GNU
```

```
REAL P(100,2),L(100,3),S2(20,4),A(100),F(100,7),Z(100,5)
CHARACTER H*20, ARCHIVO*20
REAL PI,J6,S0,S6,S7,W0,E1,X,Y,R,J1,J2,X1,Y1,X2,Y2,S,C1,C2,C3,C4,C5,Q1,Q2,Q3,Q4,X4,X5
REAL F1,X6,X3,A5,Y3,I4,L5,L6,Y6,I6,E4,Z6,O,D,I7,F0,R4,R1,R2,R3,F2
REAL S9,S1,F9,U1,I1,W,E
INTEGER P1,L1,I,J,K1,SINO
S9 = 10
PI = 4 * ATAN(1.0)
J6 = 0
```

```
C LECTURA DE VARIABLES DEL PROGRAMA
```

```
WRITE(*,*)"NOMBRE DEL ARCHIVO"
READ(*,*) ARCHIVO
OPEN(UNIT=1,FILE=ARCHIVO,STATUS="OLD",ACCESS="DIRECT",FORM="FORMATTED")
WRITE(*,*)"ELEVACION N.A.F. (0 SI NO HAY)"
READ(1,*) S0
WRITE(*,*)"DE ORD-X"
READ(1,*) S6
WRITE(*,*)"A ORD-X"
READ(1,*) S7
WRITE(*,*)"PESO VOLUMETRICO DEL AGUA"
READ(1,*) W0
WRITE(*,*)"FACTOR SISMICO"
READ(1,*) E1
WRITE(*,*)"NUMERO DE PUNTOS"
READ(1,*) P1
DO I=1,P1
WRITE(*,*)"PUNTO----->",I
WRITE(*,*)"ORD-X"
READ(1,*) P(I,1)
WRITE(*,*)"ORD-Y"
READ(1,*) P(I,2)
END DO
WRITE(*,*)"NUMERO DE LINEAS"
READ(1,*) L1
DO I=1,L1
WRITE(*,*)"LINEA----->",I
WRITE(*,*)" PUNTO IZQUIERDO"
READ(1,*) L(I,1)
WRITE(*,*)"PUNTO DERECHO"
READ(1,*) L(I,2)
WRITE(*,*)"NUMERO DE SUELO"
READ(1,*) L(I,3)
END DO
WRITE(*,*)"NUMERO DE SUELOS"
READ(1,*) S1
DO I=1,S1
WRITE(*,*)"SUELO----->",I
```

```

WRITE(*,*)"PESO VOLUMETRICO"
READ(1,*) S2(I,1)
WRITE(*,*)"COHESION"
READ(1,*) S2(I,2)
WRITE(*,*)"ANGULO DE FRICCION INTERNA"
READ(1,*) S2(I,3)
WRITE(*,*)"SUELO SATURADO (0=SI,1=NO)"
READ(1,*) S2(I,4)
END DO
CLOSE(1)
370 F9=0
WRITE(*,*)"DEFINICION DEL CIRCULO DE FALLA"
WRITE(*,*)"CENTRO-X"
READ(*,*) X
WRITE(*,*)"CENTRO-Y"
READ(*,*) Y
WRITE(*,*)"RADIO"
READ(*,*) R

C REvisa si el circulo excede los limites
430 U1=P1
DO 490 I=2,P1
IF (P(I,1) .LT. P(I-1,1) .AND. U1 .EQ. P1) GOTO 480
GOTO 490
480 U1=I-1
490 CONTINUE
J1=R*R-(P(1,2)-Y)*(P(1,2)-Y)
J2=R*R-(P(U1,2)-Y)*(P(U1,2)-Y)
IF (J1 .LE. 0) GOTO 540
IF (J1 .GT. 0 .AND. P(1,1) .GT. X-SQRT(J1)) GOTO 570
IF (J2 .LE. 0) GOTO 560
IF (J2 .GT. 0 .AND. P(U1,1) .LT. X+SQRT(J2)) GOTO 570
GOTO 590
570 WRITE(*,*)"EL CIRCULO EXCEDE LOS LIMITES"
GOTO 4230

C CALCULA LAS INTERSECCIONES DEL CIRCULO CON LAS LINEAS
590 DO 1340 I=1,L1
X1=P(L(I,1),1)
Y1=P(L(I,1),2)
X2=P(L(I,2),1)
Y2=P(L(I,2),2)
IF (X2 .EQ. X1) GOTO 670
GOTO 680
670 S=9.99E+10
680 IF (X2 .NE. X1) GOTO 700
GOTO 710
700 S=(Y2-Y1)/(X2-X1)
710 IF (ABS(S) .LT. 0.00001) GOTO 860
C1=X1-Y1/S
C2=1/(S*S)+1
C3=2*C1/S-2*Y-2*X/S
C4=C1*C1-2*X*C1+X*X+Y*Y-R*R
C5=C3*C3-4*C2*C4
IF (C5 .LT. 0) GOTO 790
GOTO 800
790 Z(I,1)=0
800 IF (C5 .LT. 0) GOTO 1340
Q1=(-C3+SQRT(C5))/(2*C2)
Q2=(-C3-SQRT(C5))/(2*C2)
Q3=Q1/S+C1
Q4=Q2/S+C1
GOTO 950
860 C5=R*R-(Y-Y1)*(Y-Y1)
IF (C5 .LT. 0) GOTO 890
GOTO 900
890 Z(I,1)=0
900 IF (C5 .LT. 0) GOTO 1340
Q3=X+SQRT(C5)
Q4=X-SQRT(C5)
Q1=Y1
Q2=Y1
950 J1=0
J2=0
IF (ABS(S) .LE. 9.989999E+09 .AND. Q3 .GE. X1 .AND. Q3 .LE. X2) GOTO 990
GOTO 1000
990 J1=1
1000 IF (ABS(S) .LT. 9.989999E+09 .AND. Q4 .GE. X1 .AND. Q4 .LE. X2) GOTO 1020
GOTO 1030
1020 J2=1
1030 IF (S .LT. -9.989999E+09 .AND. Q1 .GE. Y2 .AND. Q1 .LE. Y1) GOTO 1050
GOTO 1060
1050 J1=1
1060 IF (S .LT. -9.989999E+09 .AND. Q2 .GE. Y2 .AND. Q2 .LE. Y1) GOTO 1080
GOTO 1090
1080 J2=1
1090 IF (S .GT. 9.989999E+09 .AND. Q1 .GE. Y1 .AND. Q1 .LE. Y2) GOTO 1110
GOTO 1120
1110 J1=1
1120 IF (S .GT. 9.989999E+09 .AND. Q2 .GE. Y1 .AND. Q2 .LE. Y2) GOTO 1140
GOTO 1150
1140 J2=1
1150 Z(I,1)=J1+J2
1160 IF (J1 .EQ. 1) GOTO 1180

```

```

GOTO 1190
1180 Z(I,2)=Q3
1190 IF (J1 .EQ. 1) GOTO 1210
GOTO 1220
1210 Z(I,3)=Q1
1220 IF (J1 .EQ. 0 .AND. J2 .EQ. 1) GOTO 1240
GOTO 1250
1240 Z(I,2)=Q4
1250 IF (J1 .EQ. 0 .AND. J2 .EQ. 1) GOTO 1270
GOTO 1280
1270 Z(I,3)=Q2
1280 IF (J1 .EQ. 1 .AND. J2 .EQ. 1) GOTO 1300
GOTO 1310
1300 Z(I,4)=Q4
1310 IF (J1 .EQ. 1 .AND. J2 .EQ. 1) GOTO 1330
GOTO 1340
1330 Z(I,5)=Q2
1340 CONTINUE
X4=0
X5=9.99E+20
I1=1
DO 1510 I=1,L1
IF (Z(I,1) .GE. 1) GOTO 1410
GOTO 1420
1410 A(I1)=Z(I,2)
1420 IF (Z(I,1) .GE. 1) GOTO 1440
GOTO 1450
1440 I1=I1+1
1450 IF (Z(I,1) .EQ. 2) GOTO 1470
GOTO 1480
1470 A(I1)=Z(I,4)
1480 IF (Z(I,1) .EQ. 2) GOTO 1500
GOTO 1510
1500 I1=I1+1
1510 CONTINUE
IF (I1 .EQ. 1) GOTO 1540
GOTO 1550
1540 WRITE(*,*)"EL CIRCULO DE FALLA NO INTERSECTA EL TALUD"
1550 IF (I1 .EQ. 1) GOTO 4230

```

C FORMA ARREGLO DE DOVELAS

```

DO 1640 I=1,I1-1
IF (A(I) .GT. X4) GOTO 1600
GOTO 1610
1600 X4=A(I)
1610 IF (A(I) .LT. X5) GOTO 1630
GOTO 1640
1630 X5=A(I)
1640 CONTINUE
DO 1720 I=1, P1
IF (P(I,1) .LT. X4 .AND. P(I,1) .GT. X5) GOTO 1680
GOTO 1690
1680 A(I1)=P(I,1)
1690 IF (P(I,1) .LT. X4 .AND. P(I,1) .GT. X5) GOTO 1710
GOTO 1720
1710 I1=I1+1
1720 CONTINUE
I1=I1-1
DO 1810 I=1,I1
DO 1800 J=1,I1-1
IF (A(J+1) .GT. A(J)) GOTO 1800
J1=A(J+1)
A(J+1)=A(J)
A(J)=J1
1800 CONTINUE
1810 CONTINUE
U1=0
DO 1900 I=1,I1-1
IF (A(I) .LT. A(I+1)) GOTO 1860
GOTO 1870
1860 U1=U1+1
1870 IF (A(I) .LT. A(I+1)) GOTO 1890
GOTO 1900
1890 A(U1)=A(I)
1900 CONTINUE
U1=U1+1
A(U1)=A(I)
I1=U1

```

C DEFINE LAS FRONTERAS DE LAS DOVELAS

```

Q1=A(I1)-A(1)
Q2=Q1/S9
U1=I1
DO 2140 I=1,U1-1
Q3=A(I+1)-A(I)
Q4=INT(Q3/Q2)+1
C1=Q3/Q4
C2=A(I)
DO 2130 J=1,Q4
IF (J .LT. Q4) GOTO 2060
GOTO 2070
2060 I1=I1+1

```

```

2070     IF (J .LT. Q4) GOTO 2090
        GOTO 2100
2090     A(I1)=C2+C1
2100     IF (J .LT. Q4) GOTO 2120
        GOTO 2130
2120     C2=C2+C1
2130     CONTINUE
2140     CONTINUE
        DO 2220 I=1,I1
        DO 2210 J=1,I1-1
        IF (A(J+1) .GT. A(J)) GOTO 2210
        J1=A(J+1)
        A(J+1)=A(J)
        A(J)=J1
2210     CONTINUE
2220     CONTINUE

```

C DEFINE LOS PARAMETROS DEL SUELO PARA CADA DOVELA

```

        F1=1-1
        DO 2930 I=1,F1
        F(I,4)=A(I+1)-A(I)
        X6=F(I,4)
        F(I,7)=(A(I+1)+A(I))/2
        X3=F(I,7)
        Y1=Y-SQRT(R*R-(A(I)-X)*(A(I)-X))
        Y2=Y-SQRT(R*R-(A(I+1)-X)*(A(I+1)-X))
        A5=ATAN(ABS(Y2-Y)/F(I,4))
        IF (Y2 .LT. Y1) GOTO 2350
        GOTO 2360
2350     A5=-A5
2360     F(I,2)=A5
        IF (A5 .EQ. 0) GOTO 2390
        GOTO 2400
2390     F(I,2)=0.00001
2400     Y3=Y-SQRT(R*R-(X3-X)*(X3-X))
        I4=0
        DO 2550 J=1,L1
        L5=L(J,1)
        L6=L(J,2)
        IF (P(L5,2) .LE. Y3 .AND. P(L6,2) .LE. Y3) GOTO 2550
        IF (P(L5,1) .LT. X3 .AND. P(L6,1) .LT. X3) GOTO 2550
        IF (P(L5,1) .GT. X3 .AND. P(L6,1) .GT. X3) GOTO 2550
        Y6=P(L5,2) + (P(L5,2)-P(L6,2))/(P(L5,1)-P(L6,1))*(X3-P(L5,1))
        IF (Y6 .LE. Y3) GOTO 2550
        I4=I4+1
        Z(I4,1)=Y6
        Z(I4,2)=L(J,3)
        W=0
        E=0
2550     CONTINUE
        IF (I4 .EQ. 1) GOTO 2680
        DO 2670 J=1,I4
        DO 2660 K1=1,I4-1
        IF (Z(K1,1) .GE. Z(K1+1,1)) GOTO 2660
        L5=Z(K1,1)
        L6=Z(K1,2)
        Z(K1,1)=Z(K1+1,1)
        Z(K1,2)=Z(K1+1,2)
        Z(K1+1,1)=L5
        Z(K1+1,2)=L6
2660     CONTINUE
2670     CONTINUE
2680     I4=I4+1
        Z(I4,1)=Y3
        DO 2880 K1=1,I4-1
        IF (I .EQ. 1 .AND. K1 .EQ. 1 .AND. X3 .GE. S6) GOTO 2730
        GOTO 2740
2730     I6=S0-Y1
2740     IF (I .EQ. F1 .AND. K1 .EQ. 1 .AND. X3 .GE. S6 .AND. X3 .LE. S7) GOTO 2760
        GOTO 2770
2760     J6=S0-Y2
2770     W=W+(Z(K1,1)-Z(K1+1,1))*X6*S2(Z(K1,2),1)
        IF (Z(K1,1) .LT. S0 .AND. X3 .LE. S7 .AND. K1 .EQ. 1) GOTO 2800
        GOTO 2810
2800     W=W+(S0-Z(K1,1))*X6*W0
2810     IF (S2(Z(K1,2),4) .GT. 0.95) GOTO 2830
        GOTO 2840
2830     E4=S2(Z(K1,2),1)
2840     IF (S2(Z(K1,2),4) .LT. 0.95) GOTO 2860
        GOTO 2870
2860     E4=S2(Z(K1,2),1)-W0
2870     E=E+(Z(K1,1)-Z(K1+1,1))*X6*E4
2880     CONTINUE
        F(I,1)=W
        F(I,5)=E
        F(I,3)=S2(Z(I4-1,2),2)
        F(I,6)=2*PI*(S2(Z(I4-1,2),3)/360)
2930     CONTINUE
        IF (F9 .EQ. 0) GOTO 3130
        WRITE(*,*) "DOVELA ANCHO PESO PESO EFECTIVO"
        O=360/(2*PI)
        DO 3020 I=1,F1
        WRITE(*,*) I,F(I,4)*100+0.5/100,F(I,1)*100+0.5/100,F(I,5)*100+0.5/100

```

```

3020 CONTINUE
PAUSE

WRITE(*,*),"DOVELA INCL. COHESION PHI X"
DO 3120 I=1,F1
3120 WRITE(*,*)I,(F(I,2)*O*100+0.5)/100,(F(I,3)*100+0.5)/100,(F(I,6)*O*100+0.5)/100,(F(I,7)*100+0.5)/100
CONTINUE
PAUSE

3130 D=0
DO 3170 I=1,F1
D=D+F(I,1)*SIN(ABS(F(I,2)))*(F(I,2)/ABS(F(I,2)))
D=D+E*I*F(I,1)*COS(ABS(F(I,2)))
3170 CONTINUE
IF (I6 .GT. 0) GOTO 3200
GOTO 3210
3200 I7=W0*I6*I6*(R-I6/3)/(2*R)
3210 IF (I6 .GT. 0) GOTO 3230
GOTO 3240
3230 D=D-SIGN(1,D)*I7
3240 IF (I6 .GT. 0 .AND. F9 .EQ. 1) GOTO 3260
GOTO 3290
3260 WRITE(*,*),"FUERZA DE DESBALANCEO = ",I7
3290 IF (J6 .GT. 0) GOTO 3310
GOTO 3320
3310 I7=W0*J6*J6*(R-J6/3)/(2*R)
3320 IF (J6 .GT. 0) GOTO 3340
GOTO 3350
3340 D=D+SIGN(1,D)*I7
3350 IF (J6 .GT. 0 .AND. F9 .EQ. 1) GOTO 3370
GOTO 3390
3370 WRITE(*,*),"FUERZA INCREMENTANTE = ",I7

C SOLUCION ITERATIVA PARA EL FACTOR DE SEGURIDAD

3390 F0=1
R4=0
3430 I6=0
DO 3480 I=1,F1
R1=F(I,3)*F(I,4)+F(I,5)*TAN(F(I,6))
R2=1/COS(ABS(F(I,2)))
R3=1+TAN(F(I,6))*TAN(F(I,2))/F0
R4=R4+R1*(R2/R3)
3480 CONTINUE
F2=R4/D
I6=I6+1
IF (F9 .EQ. 1) GOTO 3530
GOTO 3600
3530 IF (I6 .EQ. 1) GOTO 3550
GOTO 3580
3550 WRITE(*,*),"ITERACION VAL.INI. VAL.CALC."
3580 WRITE(*,*)I6,(F0*1000+0.5)/1000,(F2*1000+0.5)/1000

3600 IF (I6 .GT. 10) GOTO 3620
GOTO 3640
3620 WRITE(*,*),"NO CONVERGE"
3640 IF (I6 .GT. 10) GOTO 3780
IF (ABS(ABS(F0)-ABS(F2)) .LT. 0.005) GOTO 3690
F0=ABS(F2)
R4=0
GOTO 3430

3690 WRITE(*,*),"ARCHIVO", H
WRITE(*,*),"FACTOR DE SEGURIDAD =", (F2*1000+0.5)/1000 , " AT X = ", (X*100+0.5)/100," Y= ",(Y*100+0.5)/100 , " R= ",(R*100+0.5)/100
WRITE(*,*),"MOMENTO ACTUANTE", (D*R*100+0.5)/100
WRITE(*,*),"MOMENTO RESISTENTE", (R4*R*100+0.5)/100
WRITE(*,*),"FACTOR SISMICO", (E1*1000+0.5)/1000
PAUSE

3780 IF (F9 .NE. 0) GOTO 4150
3820 WRITE(*,*),"DESEA UNA IMPRESION FORMAL (YES=1,NO=0)"
READ(*,*)SINO
IF (SINO .EQ. 0) GOTO 4090
IF (SINO .NE. 1) GOTO 3820

WRITE(*,*),"ARCHIVO".H
WRITE(*,*),"PESO VOLUMETRICO AGUA", (W0*100+0.5)/100
WRITE(*,*),"COEFICIENTE SISMICO", (E1*100+0.5)/100
WRITE(*,*),"SUMERGIDO EN", S0 , " DESDE ", S6 , " HASTA " ,S7
PAUSE

WRITE(*,*),"PUNTO X-ORD Y-ORD"
DO 3920 I=1,P1
3920 WRITE(*,*)I,(P(I,1)*100+0.5)/100,(P(I,2)*100+0.5)/100
CONTINUE
PAUSE

WRITE(*,*),"LINEA IZQ. DER. SUELO"
DO 3970 I=1,L1
3970 WRITE(*,*)I,(L(I,1)*100+0.5)/100,(L(I,2)*100+0.5)/100,(L(I,3)*100+0.5)/100
CONTINUE
PAUSE

```

```

WRITE(*,*)"SUELO PESO VOL. COHESION PHI SATURADO"
DO 4020 I=1,S1
WRITE(*,*)1,(S2(I,1)*100+0.5)/100,(S2(I,2)*100+0.5)/100,(S2(I,3)*100+0.5)/100,(S2(I,4)*100+0.5)/100
4020 CONTINUE
PAUSE

WRITE(*,*)"X-ORD Y-ORD RADIO FACTOR SEGURIDAD"
WRITE(*,*)(X*100+0.5)/100,(Y*100+0.5)/100,(R*100+0.5)/100,(F2*100+0.5)/100

4090 WRITE(*,*)"DESEA UN DIAGNOSTICO (YES=1,NO=0)"
READ(*,*)SINO
IF (SINO .EQ. 0) GOTO 4230
IF (SINO .NE. 1) GOTO 4090
F9=1
GOTO 430
4150 IF (F9 .EQ. 2) GOTO 4230
4230 WRITE(*,*)"QUIERE CONTINUAR (SI=1,NO=0)"
READ(*,*)SINO
IF (SINO .EQ. 1) GOTO 370
IF (SINO .NE. 0) GOTO 4230
WRITE(*,*)"END"

END PROGRAM BISHOP

```

B.1 Programa para analizar una falla plana. FORTRAN 90.

! ANALISIS DE FALLA PLANA
! COPYRIGHT 2012 JOSE E. TORRES, JOSENERNESTO@PRODIGY.NET.MX
! SE DISTRIBUYE BAJO LOS TERMINOS DE LA GENERAL PUBLIC LICENSE DE GNU

```

Program fallaplana
implicit none
real::H,beta,alfa,psi,o,gama,pi,gamaw,Zw,T,teta
real::Bx,By,Cx,Cy,N,M,L,A,W,U,Q,Dx,Dy
real::E,Ex,Ey,delta,S,Wy,alfas,omega,Sx,Sy,J,Jx,Jy,DELTAa,K,Kx,Ky,DELTAp,V,Vx,Vy
real::Fy,Fx,Normal,Cortante,cohesion,phi,tau,FS
integer::grieta,pasg,pacg,fe,ancla,actpas,sism
real::Zt,Zf,L1,L2,L3,L4,L5,L6,L7,P1,P2,P3,P4,P5,P6,P7

pi=4*atan(1.0)

call seleccion

! SELECCION TIPO TALUD

contains

recursive subroutine seleccion
print *, "Seleccione el tipo de talud."
print *, "Sin grieta de tension (1)"
print *, "Con grieta de tension (2)"
read *, grieta

if (grieta .EQ. 1) then
call sin_grieta
else if (grieta .EQ. 2) then
call con_grieta
else
call seleccion
end if
end subroutine

!DIMENSIONES TALUD SIN GRIETA

subroutine sin_grieta
print *, 'Altura talud'
read *, H
print *, 'Angulo talud'
read *, beta
print *, 'Angulo discontinuidad'
read *, alfa
print *, 'Angulo corona talud'
read *, psi
print *, 'Peso especifico roca'
read *, gama
print *, 'Cohesion del macizo rocoso'
read *, cohesion
print *, 'Angulo de friccion interna del macizo rocoso'
read *, phi
print *, 'Peso especifico agua'
read *, gamaw
teta=0
N=H/(sin(beta*(pi/180)))
Bx=H*(1/tan(beta*(pi/180)))
By=H
L=(H*(1-((1/tan(beta*(pi/180)))*(tan(psi*(pi/180)))))/(sin(alfa*(pi/180))-((cos(alfa*(pi/180)))*(tan(psi*(pi/180))))))
M=(L*cos(alfa*(pi/180))-H*((1/tan(beta*(pi/180)))))/(cos(psi*(pi/180)))
Cx=L*cos(alfa*(pi/180))
Cy=L*sin(alfa*(pi/180))
A=0.5*abs(Bx*Cy-By*Cx)

```

```

W=A*gama
call presion_agua_sin_grieta
end subroutine

```

!SELECCION DEL TIPO DE PRESION DE AGUA, TALUD SIN GRIETA

```

recursive subroutine presion_agua_sin_grieta
print *, 'Seleccione el tipo de presion de agua'
print *, 'Presion maxima en la mitad de la discontinuidad (1)'
print *, 'Presion maxima al pie de la discontinuidad (2)'
print *, 'Sin presion de agua (3)'
read *, pasg
if (pasg .EQ. 1) then
call presion_max_med_altura_sin_grieta
else if (pasg .EQ. 2) then
call presion_max_pie_sin_grieta
else if (pasg .EQ. 3) then
U=0
V=0
call fuerzas_externas
else
call presion_agua_sin_grieta
end if
end subroutine

```

! PRESION DE AGUA SIN GRIETA
! PRESION MAXIMA A MEDIA ALTURA

```

subroutine presion_max_med_altura_sin_grieta
V=0
Zw=L*sin(alfa*(pi/180))
U=((Zw*Zw)*(gamaw))/(4*sin(alfa*(pi/180)))
call fuerzas_externas
end subroutine

```

!MAXIMA PRESION AL PIE DEL TALUD

```

subroutine presion_max_pie_sin_grieta
V=0
Zw=L*sin(alfa*(pi/180))
U=((Zw*Zw)*gamaw)/(2*sin(alfa*(pi/180)))
call fuerzas_externas
end subroutine

```

!DIMENSION TALUD CON GRIETA DE TENSION

```

subroutine con_grieta
print *, 'Altura talud'
read *, H
print *, 'Angulo talud'
read *, beta
print *, 'Angulo discontinuidad'
read *, alfa
print *, 'Angulo corona talud'
read *, psi
print *, 'Distancia de la grieta de tensión (partiendo de la corona)'
read *, T
print *, 'Angulo de la grieta de tensión'
read *, teta
print *, 'Peso especifico roca'
read *, gama
print *, 'Cohesion del macizo rocoso'
read *, cohesion
print *, 'Angulo de friccion interna del macizo rocoso'
read *, phi
print *, 'Peso especifico agua'
read *, gamaw

N=H/(sin(beta*(pi/180)))
Bx=H*(1/(tan(beta*(pi/180))))
By=H
Cx=Bx+T
Cy=By+(T*tan(psi*(pi/180)))
M=T/(cos(psi*(pi/180)))
Q=((Cy*(1/tan(alfa*(pi/180)))-Cx)/((sin(teta*(pi/180))*(1/tan(alfa*(pi/180)))-cos(teta*(pi/180))))
L=(Cx-(Q*cos(teta*(pi/180))))/(cos(alfa*(pi/180)))
Dx=L*cos(alfa*(pi/180))
Dy=L*sin(alfa*(pi/180))
A=(0.5*(abs((Bx*Dy)-(By*Dx)))+(0.5*abs(((Dx-Bx)*(Cy-By))-((Dy-By)*(Cx-Bx))))
W=A*gama
call presion_agua_con_grieta
end subroutine

```

!SELECCION TIPO PRESION DE AGUA, TALUD CON GRIETA

```

recursive subroutine presion_agua_con_grieta
print *, 'Seleccione el tipo de presion de agua'
print *, 'Presion maxima en la mitad de la discontinuidad (1)'
print *, 'Presion maxima al pie de la discontinuidad y grieta (2)'
print *, 'Presion maxima en la base de la grieta de tension (3)'
print *, 'Sin presion de agua (4)'
read *, pacg

if (pacg .EQ. 1) then
call presion_max_med_altura_con_grieta

```

```

else if (pacg .EQ. 2) then
call presion_max_pie_con_grieta
else if (pacg .EQ. 3) then
call presion_max_base_grieta_tension
else if (pacg .EQ. 4) then
U=0
V=0
Zw=0
call fuerzas_externas
else
call presion_agua_con_grieta
end if
end subroutine

```

!PRESION MAXIMA A MEDIA ALTURA CON GRIETA

```

subroutine presion_max_med_altura_con_grieta
Zw=Cy
Zf=Dy
Zt=Zw-Zf
if(Zw <= Zf) then
V=0
U=((Zw*Zw)*gamaw)/(4*sin(alfa*(pi/180)))
call fuerzas_externas
else if (Zw > Zf .AND. (Zw*0.5) < Zf) then
L1=Zw/(2*sin(alfa*(pi/180)))
L2=L-L1
P1=0.5*Zw*gamaw
P2=gamaw*Zt
U=(0.5*P1*L1)+(0.5*(P1+P2)*L2)
V=((Zt*Zt)*gamaw)/(2*sin(teta*(pi/180)))
call fuerzas_externas
else if (Zw > Zf .AND. (Zw*0.5) >= Zf) then
P3=gamaw*Zf
P4=0.5*gamaw*Zw
L3=((Zw/2)-Zf)/(sin(teta*(pi/180)))
L4=Zw/(2*sin(teta*(pi/180)))
U=0.5*L*P3
V=(0.5*(P3+P4)*L3)+(0.5*P4*L4)
call fuerzas_externas
end if
end subroutine presion_max_med_altura_con_grieta

```

!PRESION MAXIMA EN EL PIE DEL TALUD CON GRIETA

```

subroutine presion_max_pie_con_grieta
Zf=Dy
Zw=Cy
Zt=Zw-Zf
P5=gamaw*Zt
P6=gamaw*Zw
L5=Zt/sin(teta*(pi/180))
U=(0.5*(P5+P6))*L
V=0.5*P5*L5
call fuerzas_externas
end subroutine

```

!PRESION MAXIMA EN LA BASE DE LA GRIETA DE TENSION

```

subroutine presion_max_base_grieta_tension
Zw=Cy
Zf=Dy
Zt=Zw-Zf
P7=gamaw*Zt
L7=Zt/(sin(teta*(pi/180)))
U=0.5*P7*L
V=0.5*P7*L7
call fuerzas_externas
end subroutine

```

```

subroutine fuerzas_externas

```

!FUERZAS EXTERNAS

!FUERZA ACTIVA DEL AGUA EN LA GRIETA DE TENSION

```

Vx=-V*sin(teta*(pi/180))
Vy=V*cos(teta*(pi/180))
call fuerza_externa_fe
end subroutine fuerzas_externas

```

```

recursive subroutine fuerza_externa_fe
print*, 'Fuerza Externa'
print*, 'Si (1)'
print*, 'No (2)'
read*, fe
if (fe .EQ. 1) then
call fuerza_externa
else if (fe .EQ. 2) then
E=0
delta=0
call anclaje
else
call fuerza_externa_fe
end if
end subroutine fuerza_externa_fe

```

!FUERZA EXTERNA

```
subroutine fuerza_externa
print *,'Valor de la fuerza externa'
read *,E
print *,'Angulo de la fuerza externa'
read *,delta
Ex=E*cos(delta*(pi/180))
Ey=E*sin(delta*(pi/180))
call anclaje
end subroutine
```

!ANCLAJE recursive subroutine anclaje

```
print *, 'Anclaje'
print *, 'Si (1)'
print *, 'No (2)'
read*,ancla
if (ancla .EQ. 1) then
call anclaje_act_pas
else if (ancla .EQ. 2) then
J=0
DELTAa=0
K=0
DELTAp=0
call sismo
else
call anclaje
end if
end subroutine
```

recursive subroutine anclaje_act_pas

```
print *,'Tipo de anclaje'
print *,'Activo (1)'
print *,'Pasivo (2)'
read*,actpas
if (actpas .EQ. 1) then
call activo
else if (actpas .EQ. 2) then
call pasivo
else
call anclaje_act_pas
end if
end subroutine
```

subroutine activo

!ANCLAJE ACTIVO

```
print *, 'Resistencia del anclaje activo'
read *,J
print *, 'Angulo del anclaje activo'
read *,DELTAa
Jx=J*cos(DELTAa*(pi/180))
Jy=-J*sin(DELTAa*(pi/180))
call sismo
end subroutine
```

subroutine pasivo

!ANCLAJE PASIVO

```
print *, 'Resistencia del anclaje pasivo'
read *,K
print *, 'Angulo del anclaje pasivo'
read *,DELTAp
Kx=K*cos(DELTAp*(pi/180))
Ky=-K*sin(DELTAp*(pi/180))
call sismo
end subroutine
```

recursive subroutine sismo

```
print *,'Sismo'
print *,'Si (1)'
print *,'No (2)'
read*,sism
if (sism .EQ. 1) then
```

!SISMO

```
print *,'Aceleracion sismica'
read *,alfas
print *, 'Angulo actuante aceleracion sismica'
read *,omega
Wy=(W)
S=Wy*alfas
Sx=S*cos(omega*(pi/180))
Sy=S*sin(omega*(pi/180))
call esfuerzo_normal_cortante
else if (sism .EQ. 2) then
alfas=0
omega=0
call esfuerzo_normal_cortante
else
call sismo
end if
end subroutine
```

subroutine esfuerzo_normal_cortante

!ESFUERZO NORMAL EN EL PLANO DE FALLA
!ESFUERZO CORTANTE EN EL PLANO DE FALLA

```
Fy=(-(A)*gama)-(E*sin(delta*(pi/180)))-(S*sin(omega*(pi/180)))-(J*sin(DELTAa*(pi/180)))+(V*cos(teta*(pi/180)))  
Fx=(E*cos(delta*(pi/180)))+(S*cos(omega*(pi/180)))+(J*cos(DELTAa*(pi/180)))-(V*sin(teta*(pi/180)))  
Normal=-(Fy+Ky)*cos(alfa*(pi/180))+((Fx+Kx)*sin(alfa*(pi/180))-U)  
Cortante=-(Fy*sin(alfa*(pi/180)))-(Fx*cos(alfa*(pi/180)))
```

!RESISTENCIA AL ESFUERZO CORTANTE EN EL PLANO DE FALLA
!CRITERIO DE MOHR-COULOMB
!Kx y Ky SON LA RESISTENCIA DE LOS ANCLAJES PASIVOS

```
tau=(cohesion*L)+(Normal*tan(phi*(pi/180)))+(Kx*cos(alfa*(pi/180)))+(Ky*sin(alfa*(pi/180)))
```

!FACTOR DE SEGURIDAD

```
FS=tau/Cortante
```

!!IMPRIME LOS RESULTADOS

```
print *, 'Presion del agua'  
print *, U  
print *, V  
print *, 'Esfuerzo Normal en el plano de falla'  
print *, Normal  
print *, 'Esfuerzo Cortante en el plano de falla'  
print *, Cortante  
print *, 'Resistencia al esfuerzo cortante en el plano de falla'  
print *, tau  
print *, 'Peso cuña'  
print *, W  
print *, 'Volumen cuña'  
print *, A  
print *, 'Factor de Seguridad'  
print *, FS
```

```
end subroutine esfuerzo_normal_cortante
```

```
end program falla_plana
```