

# **DESENVOLVIMENTO DE FERRAMENTAS DE CÁLCULO PARA DIMENSIONAMENTO DE MUROS DE SUPORTE DE TERRAS**

**RUI RAMOS GOMES DA SILVA**

Projecto submetido para satisfação parcial dos requisitos do grau de  
**MESTRE EM ENGENHARIA CIVIL — ESPECIALIZAÇÃO EM GEOTECNIA**

---

Orientador: Professor Doutor José Couto Marques

ABRIL DE 2008

## **MESTRADO INTEGRADO EM ENGENHARIA CIVIL 2007/2008**

DEPARTAMENTO DE ENGENHARIA CIVIL

Tel. +351-22-508 1901

Fax +351-22-508 1446

✉ [miec@fe.up.pt](mailto:miec@fe.up.pt)

*Editado por*

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Rua Dr. Roberto Frias

4200-465 PORTO

Portugal

Tel. +351-22-508 1400

Fax +351-22-508 1440

✉ [feup@fe.up.pt](mailto:feup@fe.up.pt)

🌐 <http://www.fe.up.pt>

Reproduções parciais deste documento serão autorizadas na condição que seja mencionado o Autor e feita referência a *Mestrado Integrado em Engenharia Civil - 2007/2008 - Departamento de Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2008.*

As opiniões e informações incluídas neste documento representam unicamente o ponto de vista do respectivo Autor, não podendo o Editor aceitar qualquer responsabilidade legal ou outra em relação a erros ou omissões que possam existir.

Este documento foi produzido a partir de versão electrónica fornecida pelo respectivo Autor.

## **AGRADECIMENTOS**

O autor deste trabalho agradece a todos aqueles que auxiliaram, directa ou indirectamente, na elaboração deste trabalho, com especial realce:

- Ao Professor Doutor José Couto Marques, orientador deste trabalho, pelo apoio e disponibilidade demonstrados ao longo da execução do projecto;
- Aos meus familiares pelo apoio que me deram durante o período de elaboração deste projecto;
- A todos os colegas e amigos, principalmente aos de Geotecnia, pelos diversos contributos que deram e que em muito auxiliaram na execução deste trabalho;



## **RESUMO**

Com o presente trabalho, pretende-se elaborar uma ferramenta de cálculo automático que facilite o dimensionamento de estruturas de suporte de terras – muros de suporte, através de uma interface bastante intuitiva. Para tal, recorre-se ao Microsoft® Office Excel, visto ser um programa de fácil utilização, acessível a qualquer comum utilizador de computador e, acima de tudo, através do Microsoft® Office Excel, foi possível programar vários módulos em VBA – *Visual Basic for Applications*.

Mediante a inserção dos dados relativos ao muro e características de todo o terreno envolvente (fundação e terraplano), o programa gera as seguintes informações: segurança, escorregamento, derrube, capacidade de carga e estabilidade global.

**PALAVRAS-CHAVE:** Muros de suporte, escorregamento, derrube, gabiões, dimensionamento de muros.



**ABSTRACT**

With this project, the main goal is to create an automatic tool that assists in the design of gravity retaining walls through a very intuitive interface. This is achieved by using Microsoft® Office Excel, as this is powerful software tool, easily accessible to anyone who uses the computer on a regular basis. Several modules have been developed in VBA – *Visual Basic for Applications* – through Microsoft® Office Excel.

After introducing all the data regarding the wall, the backfill and the foundation, the program generates the following information: stability against sliding, stability against overturning, foundation bearing capacity and overall stability.

**KEYWORDS:** Retaining walls, sliding, overturning, gabion retaining wall, gravity retaining wall design.





## ÍNDICE GERAL

<b>1. INTRODUÇÃO</b>	1
<b>2. PRINCÍPIOS TEÓRICOS</b>	3
<b>2.1. GENERALIDADES</b>	3
<b>2.2. DETERMINAÇÃO DO IMPULSO</b>	3
2.2.1. IMPULSO DO SOLO	3
2.2.1.1. Ângulo $\lambda$	4
2.2.2. IMPULSO ACTIVO DEVIDO À SOBRECARGA	5
2.2.3. ACRÉSCIMO DE IMPULSO ACTIVO DEVIDO À ACÇÃO SÍSMICA	5
<b>2.3. VERIFICAÇÃO DA SEGURANÇA</b>	6
2.3.1. VERIFICAÇÃO AO DERRUBE	6
2.3.2. VERIFICAÇÃO AO ESCORREGAMENTO	7
2.3.3. VERIFICAÇÃO DA CAPACIDADE DE CARGA	8
2.3.4. VERIFICAÇÃO DA ESTABILIDADE GLOBAL	9
2.3.5. VERIFICAÇÃO DA ESTABILIDADE INTERNA	10
<b>3. DESCRIÇÃO DO PROGRAMA</b>	11
<b>3.1. GENERALIDADES</b>	11
<b>3.2. SELECÇÃO DO TIPO DE MURO</b>	11
<b>3.3. DEFINIÇÃO DO MURO</b>	11
<b>3.4. DEFINIÇÃO DA FUNDAÇÃO</b>	12
<b>3.5. DEFINIÇÃO DO TERRAPLENO</b>	13
<b>3.6. DEFINIÇÃO DAS ACÇÕES</b>	14
<b>3.7. CÁLCULO</b>	15
<b>4. EXEMPLO PRÁTICO</b>	17
<b>4.1. GENERALIDADES</b>	17
<b>4.2. EXEMPLO</b>	17
<b>5. FUTURAS ACTUALIZAÇÕES</b>	27
<b>6. BIBLIOGRAFIA</b>	29
<b>ANEXOS</b>	31



## SIMBOLOGIA

### ALFABETO LATINO

$c'$	coesão
$F$	coeficiente de segurança global
$h$	altura das terras
$I_a$	impulso activo devido à acção do peso próprio do terreno
$I_p$	impulso passivo devido à acção do peso próprio do terreno
$I_q$	impulso devido a sobrecarga
$K_a$	coeficiente de impulso activo
$K_{as}$	coeficiente de impulso activo sísmico
$k_h$	coeficiente sísmico horizontal
$K_p$	coeficiente de impulso passivo
$k_v$	coeficiente sísmico vertical
$l$	comprimento da secção circular
$q$	carga aplicada no terrapleno
$u_i$	pressão neutra na base da fatia
$W_i$	peso de cada fatia

### ALFABETO GREGO

$\beta$	inclinação das terras suportadas
$\delta$	ângulo de atrito da interface terras-muro
$\Delta I_{as}$	incremento de impulso associado à acção sísmica
$\Delta l_i$	comprimento da base de uma fatia
$\Delta x_i$	projecção horizontal do comprimento da base de uma fatia
$\phi'$	ângulo de atrito do solo
$\gamma$	peso volúmico
$\lambda$	inclinação da superfície de contacto terras muro
$\theta_i$	ângulo que o raio que passa no centro da base da fatia faz com a vertical



# 1

## INTRODUÇÃO

Os muros de suporte são estruturas concebidas para sustentar terras quando estas exibem um declive propício a instabilidades. De outro modo, estes muros sustentam o maciço terroso quando se pretende ter um desnível da superfície do terreno num comprimento reduzido como por exemplo na construção rodoviária, protecção de margens de cursos de água, estabilização de escarpas.

Estas estruturas funcionam maioritariamente à gravidade fazendo face aos impulsos resultantes do contacto com o maciço adjacente e outras acções, ou seja, funcionam praticamente com o peso do muro e o peso das terras acima do muro, a contrariar as solicitações por parte do terraplano (maioritariamente horizontais).

Estes muros têm duas tipologias principais. Podem ser construídos com caixas flexíveis de rede de aço galvanizado, permeáveis e monolíticas, preenchidas com material grosseiro estando-se assim perante uma solução de muro de gabiões. Podem também ser realizados a partir de betão recorrendo a cofragens para moldar a forma do muro. Em ambos os casos, é necessário proceder-se à escavação do talude, edificação do muro e conseqüente preenchimento da zona adjacente ao muro.

Este trabalho consistiu na criação de uma ferramenta de cálculo (uma folha em Microsoft® Office Excel) que possibilita o dimensionamento de estruturas deste género. Para tal, através de uma interface utilizador-computador bastante intuitiva e sempre com representação gráfica, solicitam-se os dados relativos ao muro, ao terraplano e à fundação. Para além dos impulsos provenientes das terras, atende-se a possíveis sobrecargas no terraplano, presença de água no solo, bem como a ocorrência de acção sísmica. Posteriormente calculam-se os impulsos das terras sobre a estrutura e verifica-se a segurança ao derrube e ao escorregamento, a capacidade de carga da fundação e a estabilidade global do talude em que o muro se insere. Estas verificações baseiam-se nas Teorias dos Estados de Equilíbrio Limite (Teorias de *Rankine* e de *Coulomb*), e nas Teorias de *Bishop* e de *Fellenius*.

Foi preparado um manual de utilização que proporciona auxílio na utilização do programa, no qual se apresenta um exemplo-tipo e se referem todos os passos e recomendações que o utilizador deve ter em conta.



# 2

## PRINCIPIOS TEÓRICOS

### 2.1. GENERALIDADES

Nos problemas de dimensionamento de muros de suporte verifica-se que o solo transmite impulsos ao muro e este terá que os suportar sem que a estrutura entre em colapso. Para tal a contribuição do peso do muro é fundamental para compensar os impulsos do terreno e possíveis sobrecargas.

A função deste tipo de estrutura é somente suportar o maciço terroso adjacente e não de o comprimir. Deste modo, o dimensionamento irá ser feito tendo em conta a iminência de colapso sendo considerados os impulsos provenientes do terraplano como activos.

Para o efeito, recorre-se às Teorias dos Estados de Equilíbrio Limite - Método de Coulomb, que assume os seguintes pressupostos:

O maciço é constituído por um solo emerso, não coesivo e homogéneo.

A totalidade da resistência ao corte do solo é mobilizada ao longo das superfícies de deslizamento e ao longo da superfície de contacto terras-muro.

A superfície de deslizamento é plana e passa pela base do muro.

O muro é suficientemente extenso para que sejam desprezáveis os efeitos tridimensionais.

Partindo das soluções analíticas do Método de Coulomb determinam-se os impulsos e avalia-se o comportamento da estrutura.

### 2.2. DETERMINAÇÃO DO IMPULSO

#### 2.2.1. IMPULSOS DO SOLO

A determinação dos impulsos, como se referiu anteriormente, é feita a partir das soluções analíticas obtendo-se os valores extremos de impulsos - impulsos activos e passivos.

Desta forma, os impulsos provocados pelo solo são calculados pelas seguintes expressões:

$$I_a = \frac{1}{2} \cdot K_a \cdot \gamma \cdot h^2 \quad (2.1.)$$

$$I_p = \frac{1}{2} \cdot K_p \cdot \gamma \cdot h^2 \quad (2.2.)$$

em que K é o coeficiente de impulso (activo ou passivo),  $\gamma$  é o peso volúmico das terras e h é a altura das mesmas.

Os coeficientes de impulso são calculados segundo a teoria acima citada através das seguintes expressões:

$$K_a = \frac{\cos^2(\phi' - \lambda)}{\cos^2(\lambda) \cdot \cos(\delta + \lambda) \left[ 1 + \sqrt{\frac{\sin(\phi' + \delta) \cdot \sin(\phi' - \beta)}{\cos(\beta - \lambda) \cdot \cos(\delta + \lambda)}} \right]^2} \quad (2.3.)$$

$$K_p = \frac{\cos^2(\phi' + \lambda)}{\cos^2(\lambda) \cdot \cos(\delta - \lambda) \left[ 1 - \sqrt{\frac{\sin(\phi' + \delta) \cdot \sin(\phi' + \beta)}{\cos(\beta - \lambda) \cdot \cos(\delta - \lambda)}} \right]^2} \quad (2.4.)$$

em que, tal como se representa na figura 2.1.,  $\beta$  representa a inclinação das terras suportadas,  $\lambda$  a inclinação da superfície de contacto terra-muro,  $\phi'$  o ângulo de atrito do solo e  $\delta$  ângulo de atrito da interface estrutura-solo.

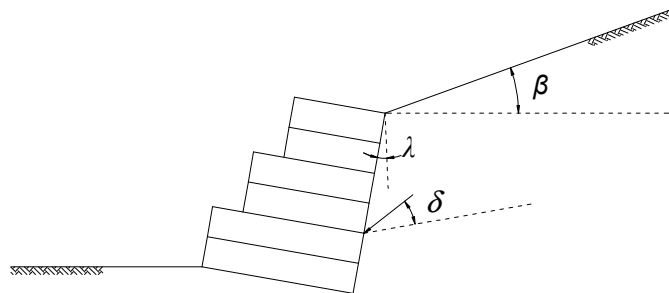


Fig. 2.1. – Representação dos factores que interferem no cálculo dos coeficiente de impulso

Este tipo de impulso tem como ponto de aplicação o terço inferior da altura h, dado o formato triangular do diagrama de tensões.

#### 2.2.1.1. Ângulo $\lambda$

O ângulo  $\lambda$  representa a inclinação da superfície de contacto terras-muro, tal como ilustrado na figura anterior, no entanto, no caso em que a geometria da estrutura não apresenta uma face de contacto com o solo tão regular, este factor não é de determinação imediata. Como simplificação neste trabalho, toma-se  $\lambda$  como o ângulo com a vertical formado pela recta que passa pelos vértices superior e inferior da face do muro, como representado na figura seguinte.



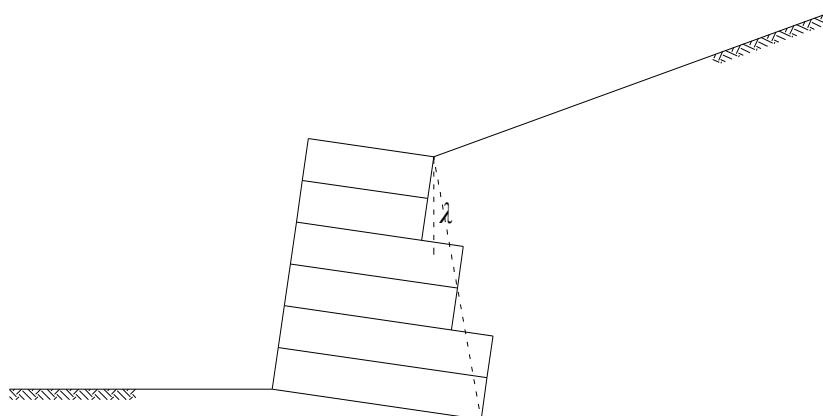


Fig. 2.2. – Definição do ângulo  $\lambda$

Para um maior rigor deveria ser calculada a recta que melhor aproxime as coordenadas dos vértices do tardo do muro, no entanto tal procedimento revela-se de programação complexa.

#### 2.2.2. IMPULSO ACTIVO DEVIDO À SOBRECARGA

Durante o período de utilização do muro este pode estar sujeito a acções de sobrecarga. Como tal, este factor tem que estar contemplado no dimensionamento de uma estrutura destas.

Como os impulsos devido à sobrecarga são somente desfavoráveis quando a sobrecarga é aplicada no terrapleno, não se contemplam os impulsos passivos devido a sobrecargas.

Quando é aplicada uma carga  $q$  no terrapleno, este vai solicitar o muro com um impulso dado por:

$$I_q = K_a \cdot q \cdot h \quad (2.5.)$$

O ponto de aplicação deste tipo de impulsos situa-se a meio da altura das terras –  $h$ , dado a geometria rectangular do diagrama de tensões

#### 2.2.3. ACRÉSCIMO DE IMPULSO ACTIVO DEVIDO À ACÇÃO SÍSMICA

A acção sísmica deve ser contemplada no dimensionamento de uma estrutura. Para tal recorre-se ao Método de Mononobe-Okabe utilizando-se a expressão que a seguir se apresenta para determinar o acréscimo de impulso devido a esta acção.

$$\Delta I_{as} = \frac{1}{2} \cdot \gamma \cdot h^2 \cdot [(1 \pm k_v) \cdot K_{as} - K_a] \quad (2.6.)$$

em que  $k_v$  representa o coeficiente sísmico vertical, e  $K_{as}$  o coeficiente de impulso activo sísmico definindo-se este último pela seguinte fórmula:

$$K_{as} = \frac{\cos^2(\phi' - \lambda - \theta)}{\cos(\theta) \cdot \cos^2(\lambda) \cdot \cos(\delta + \lambda + \theta) \left[ 1 + \sqrt{\frac{\sin(\phi' + \delta) \cdot \sin(\phi' - \beta - \theta)}{\cos(\beta - \lambda) \cdot \cos(\delta + \lambda + \theta)}} \right]^2} \quad (2.7.)$$

sendo:

$$\theta = \arctan\left(\frac{k_h}{1 \pm k_v}\right) \quad (2.8.)$$

Neste método pseudo-estático as forças de inércia horizontal e vertical associadas à massa do muro são obtidas multiplicando o peso deste pelos respectivos coeficientes sísmicos.

### 2.3. VERIFICAÇÃO DA SEGURANÇA

Após a determinação das acções que actuam na estrutura, é necessário verificar se esta não apresenta anomalias no seu bom funcionamento. Para tal estudam-se os diversos tipos de rotura típicos destas estruturas.

#### 2.3.1. VERIFICAÇÃO AO DERRUBE

A verificação da segurança do muro ao derrube consiste em comparar os momentos estabilizador e derrubador em relação à aresta esquerda da base provocados, respectivamente, pela acção do peso da estrutura e pelos impulsos aplicados no muro.

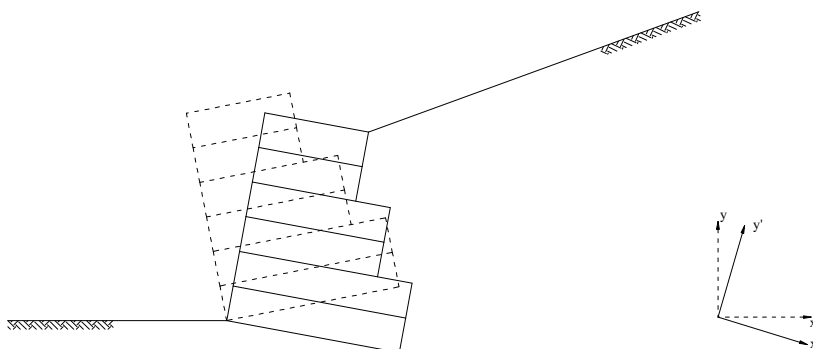


Fig. 2.3. – Representação do derrube e eixos auxiliares

O cálculo do momento estabilizador consiste no somatório, segundo um eixo  $x' y'$  tal como apresenta a figura 2.3., dos produtos das componentes do peso de cada bloco pelos respectivos braços, ou seja:

$$M_{est} = \sum_{i=1}^n W_{iy'} \cdot x'_i - W_i \cdot k_{hy'} \cdot x'_i \pm W_i \cdot k_{vy'} \cdot x'_i + W_{ix'} \cdot y'_i - W_i \cdot k_{hx'} \cdot y'_i \pm W_i \cdot k_{vx'} \cdot y'_i \quad (2.9.)$$

em que  $n$  representa o número de camadas de blocos presentes no muro,  $i$  representa o bloco em análise,  $W_i$  o peso do bloco  $i$ ,  $W_{iy'}$  a componente de  $W_i$  segundo o eixo  $y'$ ,  $W_{ix'}$  a componente de  $W_i$  segundo o eixo  $x'$ ,  $y'_i$  a distância entre o ponto de rotação e o centro geométrico do bloco  $i$  segundo o eixo do  $y'$ ,  $x'_i$  a distância entre o ponto de rotação e o centro geométrico do bloco  $i$  segundo o eixo do  $x'$  e  $k_{hy'}$ ,  $k_{hx'}$ ,  $k_{vy'}$  e  $k_{vx'}$  como o coeficiente sísmico (horizontal ou vertical) segundo um dos eixos ( $x'$  ou  $y'$ ).

O momento derrubador é determinado pelo somatório das componentes dos diversos impulsos, segundo os dois eixos, multiplicados pelo respectivo braço, ou seja:

$$M_{der} = I_{ax'} \cdot y'_i - I_{ay'} \cdot x'_i + I_{qx'} \cdot y'_i - I_{qy'} \cdot x'_i + \Delta I_{asx'} \cdot y'_i - \Delta I_{asy'} \cdot x'_i - I_{px'} \cdot y'_i - I_{py'} \cdot x'_i \quad (2.10.)$$

Em que  $I_{ax'}$  representa a componente do impulso activo no eixo  $x'$ ,  $I_{ay'}$  representa a componente do impulso activo no eixo  $y'$ ,  $y'_i$  a distância entre o ponto de rotação e ponto de aplicação do impulso segundo o eixo do  $y'$ ,  $x'_i$  a distância entre o ponto de rotação e ponto de aplicação do impulso segundo o eixo do  $x'$ ,  $I_{qx'}$  e  $I_{qy'}$  as componentes do impulso devido à acção de uma sobrecarga, e  $\Delta I_{as}$  as componentes do incremento de impulso devido à acção sísmica. A este somatório de impulsos ainda poderá ser adicionada a componente correspondente ao impulso devido à presença de água e o impulso passivo.

### 2.3.2. VERIFICAÇÃO AO ESCORREGAMENTO

A verificação da segurança do muro ao escorregamento consiste em comparar as forças estabilizadoras associadas ao peso do muro com as forças instabilizadoras derivadas dos impulsos.

O escorregamento ocorre pela base do muro, fazendo com que a estrutura deslize sobre a fundação, razão pela qual se consideram apenas as componentes das acções tangenciais à fundação. A figura que se segue ilustra o escorregamento de um muro.

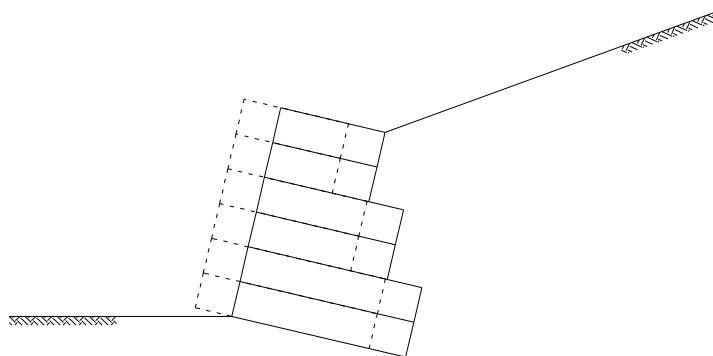


Fig. 2.4. – Representação de um escorregamento

Para o cálculo da força estabilizadora recorre-se à seguinte expressão:

$$F_{est} = \sum_{i=1}^n (W_{iy'} - W_i \cdot k_{hy'} \pm W_i \cdot k_{vy'}) \cdot \tan(\delta_b) + W_{ix'} - W_i \cdot k_{hx'} \pm W_i \cdot k_{vx'i} \quad (2.11.)$$

em que  $\delta_b$  representa o ângulo de atrito entre a base do muro e o maciço de fundação. Para o cálculo da força instabilizadora recorre-se à seguinte expressão:

$$F_{ins} = I_{ax'} + I_{qx'} + \Delta I_{asx'} - I_{px'} - (I_{ay'} + I_{qy'} + \Delta I_{asy'} + I_{py'}) \cdot \tan(\delta_b) \quad (2.12.)$$

Mais uma vez poderão ser adicionados os impulsos devido à existência de nível freático e o impulso passivo.

### 2.3.3. VERIFICAÇÃO DA CAPACIDADE DE CARGA

A verificação da capacidade de carga da fundação é efectuada de acordo com o EC7, Anexo D, a partir do qual se calcula a carga máxima que o solo pode suportar e se compara com a carga actuante do muro.

O cálculo da capacidade resistente do terreno assenta na seguinte expressão para o caso da fundação se encontrar em condições drenadas:

$$\frac{Q_{ult}}{A'} = c' \cdot N_c \cdot b_c \cdot s_c \cdot i_c + q' \cdot N_q \cdot b_q \cdot s_q \cdot i_q + 0,5 \cdot \gamma' \cdot B' \cdot N_\gamma \cdot b_\gamma \cdot s_\gamma \cdot i_\gamma \quad (2.13.)$$

em que,

$$N_q = e^{\pi \cdot \tan \varphi'} \cdot \tan^2 \left( 45 + \frac{\varphi'}{2} \right) \quad (2.14.)$$

$$N_c = (N_q - 1) \cdot \cot \varphi' \quad (2.15.)$$

$$N_\gamma = 2 \cdot (N_q - 1) \cdot \tan \varphi', \text{ com } \delta \geq \frac{\varphi'}{2} \text{ (base rugosa)} \quad (2.16.)$$

$$b_q = b_y = (1 - \alpha \cdot \tan \varphi')^2 \quad (2.17.)$$

$$b_c = b_q - \frac{1 - b_q}{N_c \cdot \tan \varphi'} \quad (2.18.)$$

$$s_q = 1 + \left( \frac{B'}{L'} \right) \cdot \sin \varphi' \text{ para uma forma rectangular} \quad (2.19.)$$

$$s_q = 1 + \sin \varphi' \text{ para uma forma quadrada ou circular} \quad (2.20.)$$

$$s_\gamma = 1 - 0,3 \frac{B'}{L'} \text{ para uma forma rectangular} \quad (2.21.)$$

$$s_\gamma = 0,7 \text{ para uma forma quadrada ou circular} \quad (2.22.)$$

$$s_c = \frac{S_q \cdot N_q - 1}{N_q - 1} \text{ para uma forma rectangular, quadrada ou circular} \quad (2.23.)$$

$$i_c = i_q - \frac{1 - i_q}{N_c \cdot \tan \varphi'} \quad (2.24.)$$

$$i_q = \left[ 1 - \frac{H}{V + A' \cdot c' \cdot \cot \varphi'} \right]^m \quad (2.25.)$$

$$i_\gamma = \left[ 1 - \frac{H}{V + A' \cdot c' \cdot \cot \varphi'} \right]^{m+1} \quad (2.26.)$$

$$m = m_B = \frac{2 + \frac{B'}{L'}}{1 + \frac{B'}{L'}} \text{ quando H actua na direcção de B'} \quad (2.27.)$$

$$m = m_L = \frac{2 + \frac{B'}{L'}}{1 + \frac{B'}{L'}} \text{ quando H actua na direcção de L'} \quad (2.28.)$$

ou para o caso de condições não drenadas, a seguinte expressão:

$$\frac{Q_{ult}}{A'} = (\pi + 2) \cdot c_u \cdot b_c \cdot s_c \cdot i_c + q \quad (2.29.)$$

em que,

$$b_c = 1 - \frac{2\alpha}{\pi + 2} \quad (2.30.)$$

$$s_c = 1 + 0,2 \frac{B'}{L'} \text{ para uma forma rectangular} \quad (2.31.)$$

$$s_c = 1,2 \text{ para uma forma quadrada ou circular} \quad (2.32.)$$

$$i_c = 0,5 \left[ 1 + \sqrt{1 - \frac{H}{A' \cdot c_u}} \right] \text{ com } H \leq A' \cdot c_u \quad (2.33.)$$

sendo  $B'$  a largura efectiva da fundação,  $b$  coeficientes de inclinação da base,  $i$  coeficientes de inclinação da carga,  $L$  comprimento efectivo da fundação,  $N$  factores de capacidade de carga,  $q'$  o valor de cálculo da tensão efectiva vertical ao nível da base da fundação,  $s$  coeficientes de forma da base de fundação,  $V$  a carga vertical e  $\alpha$  inclinação da base de fundação.

Após o cálculo da capacidade resistente do terreno, compara-se este com a componente do peso do muro normal à base do mesmo ( $N$ ), obtendo-se desta forma um factor de segurança.

$$FS_{cc} = \frac{Q_{ult}}{N} \quad (2.34.)$$

#### 2.3.4. VERIFICAÇÃO DA ESTABILIDADE GLOBAL

A verificação da estabilidade global envolve a consideração de várias superfícies circulares de escorregamento no maciço envolvente da estrutura, avaliando-se a mobilização da resistência ao corte ao longo dessas superfícies, tal como exemplifica a figura 2.5.

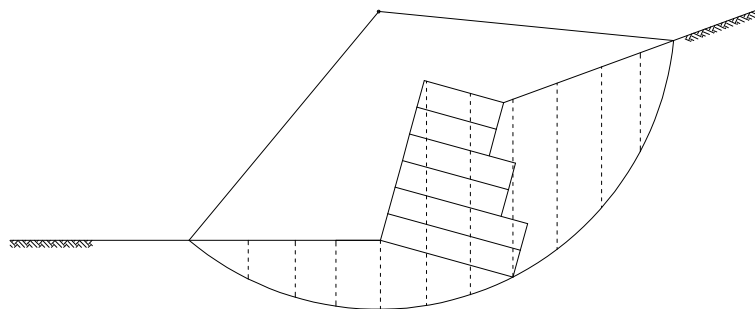


Fig. 2.5. – Representação de uma superfície de escorregamento

Deste modo é necessário identificar a superfície de escorregamento crítica, ou seja, a que apresente um menor coeficiente de segurança. Para tal, recorre-se aos métodos de *Fellenius* e de *Bishop*.

Na sua globalidade, estes métodos consistem em dividir o volume acima da superfície circular de escorregamento em várias fatias verticais e analisar os momentos do peso próprio de cada uma em relação ao centro da superfície.

O método de *Fellenius* é traduzido pela seguinte expressão:

$$F = \frac{c' \cdot l + \tan \phi' \sum_{i=1}^n (W_i \cdot \cos \theta_i - u_i \cdot \Delta l_i)}{\sum_{i=1}^n W_i \cdot \sin \theta_i} \quad (2.35.)$$

em que  $F$  é o coeficiente de segurança global,  $c'$  a coesão,  $l$  o comprimento da superfície circular,  $W_i$  o peso de cada fatia,  $\theta_i$  o ângulo que o centro da fatia faz com a vertical em relação ao centro de rotação,  $\Delta l_i$  o comprimento da base de cada fatia e  $u_i$  a pressão neutra na base da fatia.

Enquanto que o método anterior considera as forças de interação entre fatias paralelas à base destas, o método de *Bishop* admite-as como horizontais. Este segundo método é traduzido pela seguinte expressão:

$$F = \frac{\sum_{i=1}^n [c' \cdot \Delta x_i + (W_i - u_i \cdot \Delta x_i) \cdot \tan \phi'] [1 / M_i(\theta)]}{\sum_{i=1}^n W_i \cdot \sin \theta_i} \quad (2.36.)$$

na qual,

$$M_i(\theta) = \cos \theta_i \left( 1 + \frac{\tan \theta_i \cdot \tan \phi'}{F} \right) \quad (2.37.)$$

em que  $\Delta x_i$  é a projecção horizontal do comprimento da base de cada fatia.

### 2.3.5. VERIFICAÇÃO DA ESTABILIDADE INTERNA

Esta verificação consiste na análise ao derrube e ao deslizamento, no caso de muros de gabiões, de todos os conjuntos de blocos possíveis, tal como ilustra a figura 2.6. Esta análise é efectuada dado que o muro não é um elemento sólido uniforme, mas sim um conjunto de blocos sobrepostos interligados que podem não responder adequadamente quando solicitados individualmente.

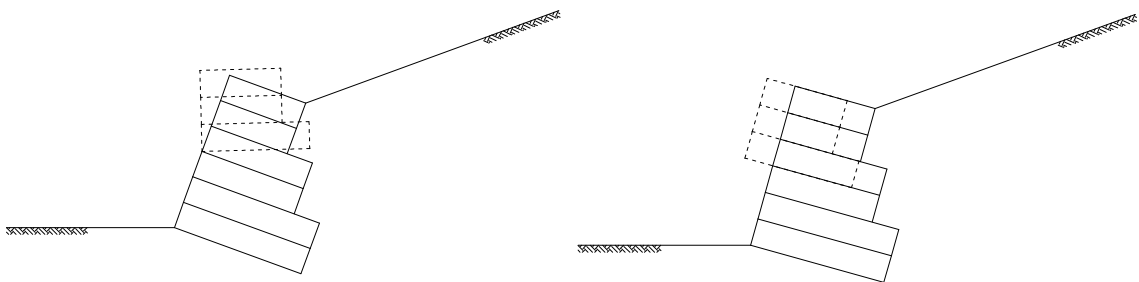


Fig. 2.6. – Representação da verificação da estabilidade interna.





# 3

## DESCRIÇÃO DO PROGRAMA

### 3.1. GENERALIDADES

No corrente capítulo será descrito o funcionamento do programa, incluindo o modo de introdução de dados, assim como a explicação de todos os cálculos subsequentes. Para isso serão apresentados e devidamente comentados os diversos menus do programa.

### 3.2. SELECÇÃO DO TIPO DE MURO

Este menu, apresentado na figura 3.1., permite a selecção pelo utilizador do tipo de muro a estudar, podendo escolher-se entre muros de gabiões e muros de gravidade. Após a selecção, será necessário carregar no botão “*Próximo*” para passar para o menu seguinte.

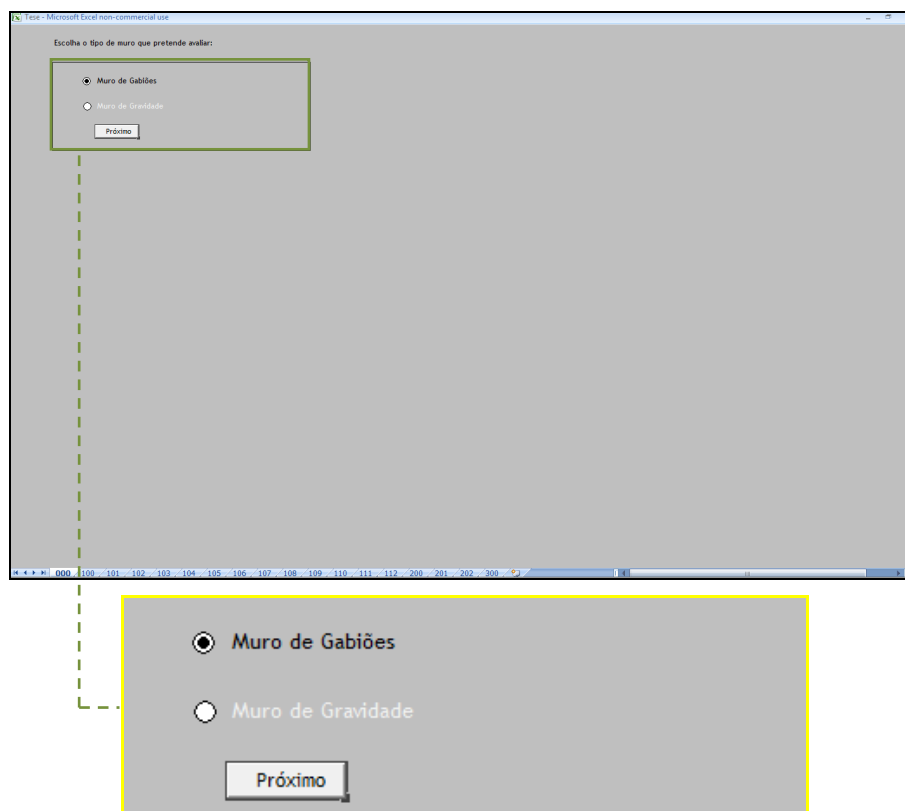


Fig. 3.1. – Representação do menu inicial.

### 3.3. DEFINIÇÃO DO MURO

O menu apresentado na figura 3.2. é o responsável pela definição das características e geometria do muro. Assim, como nos restantes menus, os dados deverão ser inseridos nos espaços em branco. Como tal, este menu permite definir directamente a inclinação, o peso volúmico do material do muro e, no caso dos muros de gabiões, a porosidade destes.

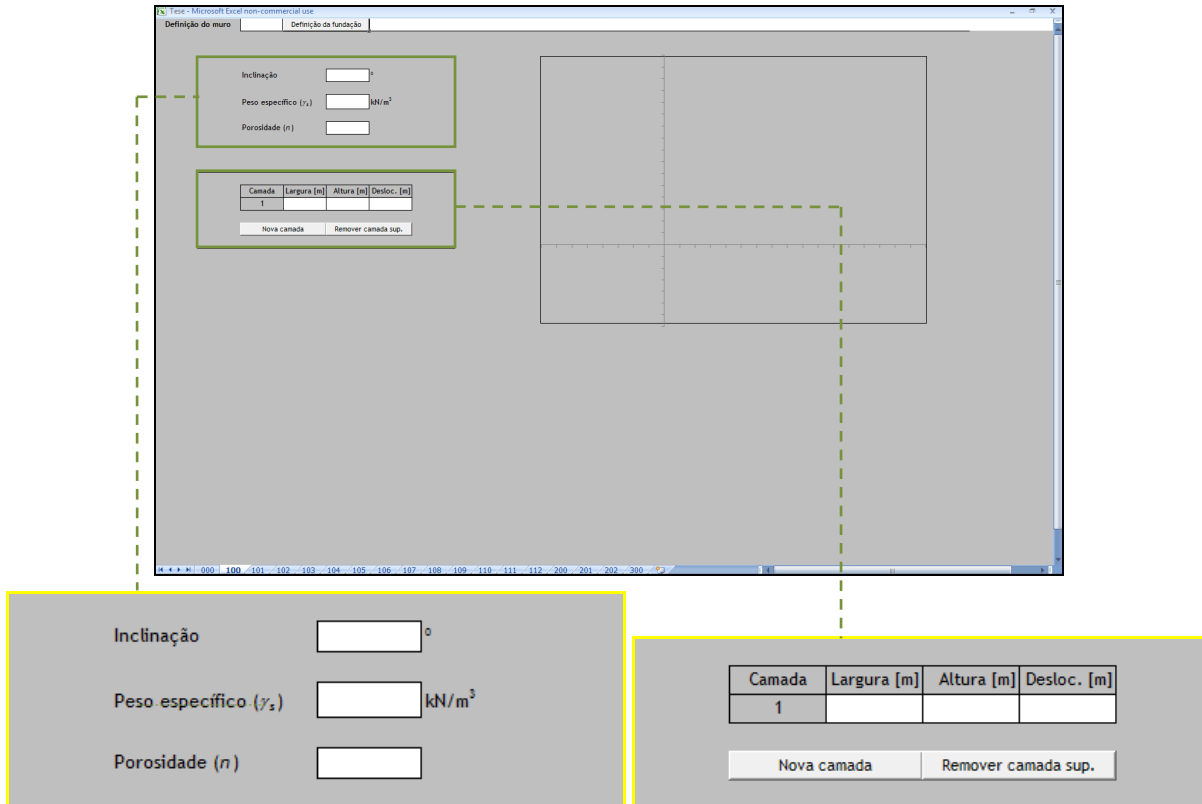


Fig. 3.2. – Representação do menu “Definição do muro”.

No que diz respeito às dimensões, estas podem ser definidas no segundo quadro deste menu. Para tal, é necessário preencher obrigatoriamente as colunas da “Largura” e “Altura”. No caso da primeira camada de blocos, a aresta inferior da esquerda funcionará como origem do sistema de eixos. No caso dos blocos subsequentes, se adicionar algum valor na coluna de “Desloc.”, o bloco vai mover-se essa distância em relação à aresta superior esquerda do bloco anterior.

Se o utilizador pretender adicionar uma nova camada de blocos, terá que carregar no botão “Nova camada” e automaticamente aparecerá uma nova linha para preenchimento.

De forma a evitar uma incorrecta introdução de dados, este menu disponibiliza a visualização imediata da geometria do muro. Como tal, se o utilizador detectar alguma falha na introdução dos dados, poderá alterá-los através da introdução dos valores correctos ou da remoção de camadas em excesso através do botão “Remover camada sup.”.

Após a introdução destes dados, o programa gera numa folha anexa as coordenadas de todos os vértices dos blocos

### 3.4. DEFINIÇÃO DA FUNDAÇÃO

A figura 3.3. representa o menu no qual se define a fundação. Este menu possibilita a introdução da geometria das terras à direita do muro e a existência ou não de nível freático e respectiva cota. Para além de dados referentes à geometria, é também possível definir as características mecânicas do solo - peso volúmico, ângulo de atrito e coesão.

Para efeitos de verificação da capacidade de carga, o utilizador poderá optar por uma fundação em condições drenadas ou não drenadas sendo neste caso possível definir o valor da resistência não drenada.

No segundo quadro poderão ser adicionados novos estratos à fundação. Por uma questão de lógica os estratos deverão ser inseridos por ordem (do mais superficial para o mais profundo). Sempre que adicionar um novo horizonte, o utilizador terá que introduzir a respectiva cota, peso volúmico, ângulo de atrito e coesão, de forma a tornar possível o cálculo posterior do coeficiente de estabilidade global.

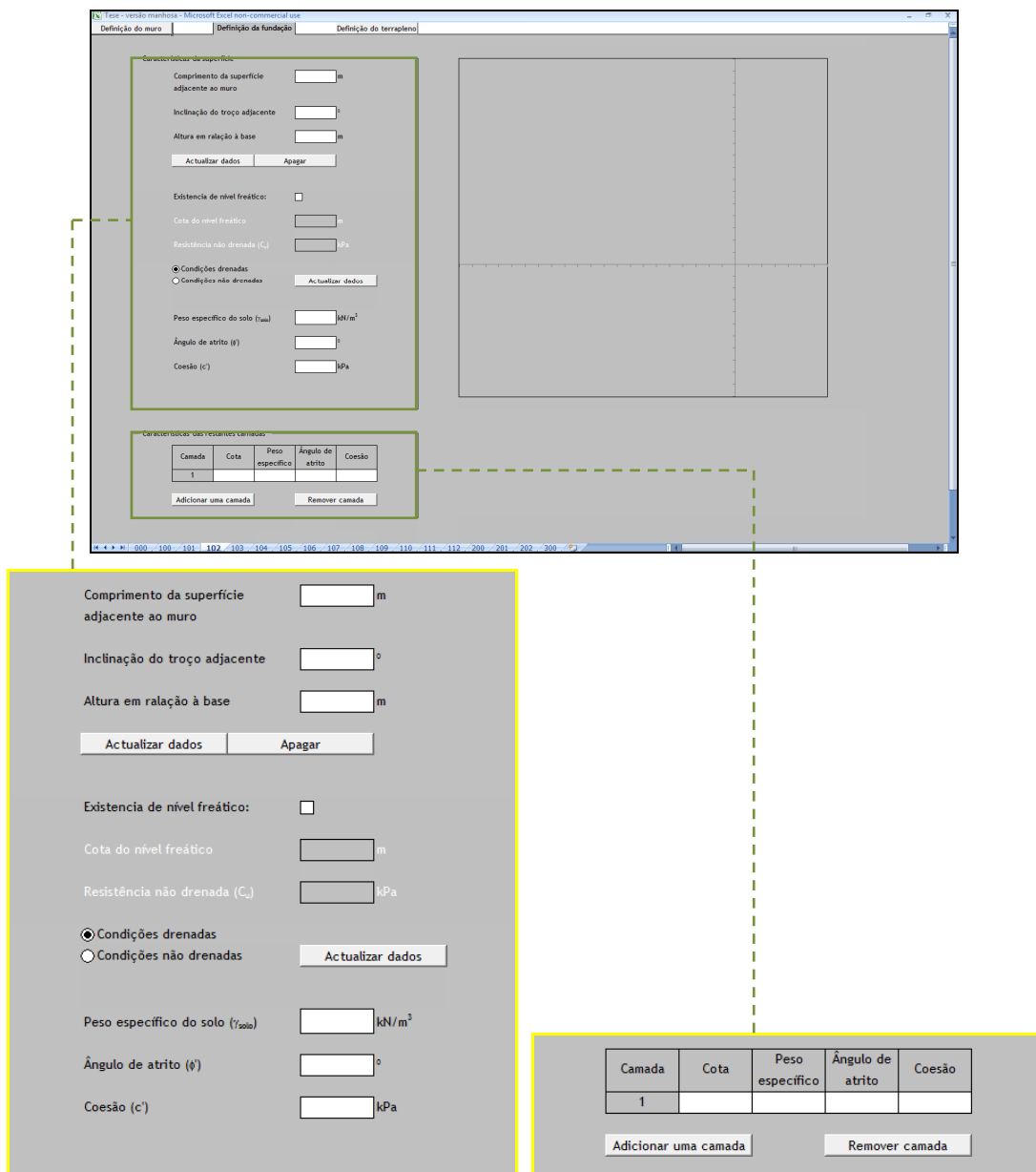


Fig. 3.3. – Representação do menu “Definição da fundação”.

Por defeito, ao serem adicionados, os novos estratos admitem automaticamente uma cota 1 metro inferior ao estrato anterior.

Tal como no menu anterior, este possui um gráfico que representa a estrutura, bem como a geometria da fundação. Para uma correcta visualização da geometria, após a inserção dos dados é necessário pressionar o botão “*Actualizar dados*”.

### 3.5. DEFINIÇÃO DO TERRAPLENO

O terrapleno é definido pelo menu ilustrado na figura seguinte.

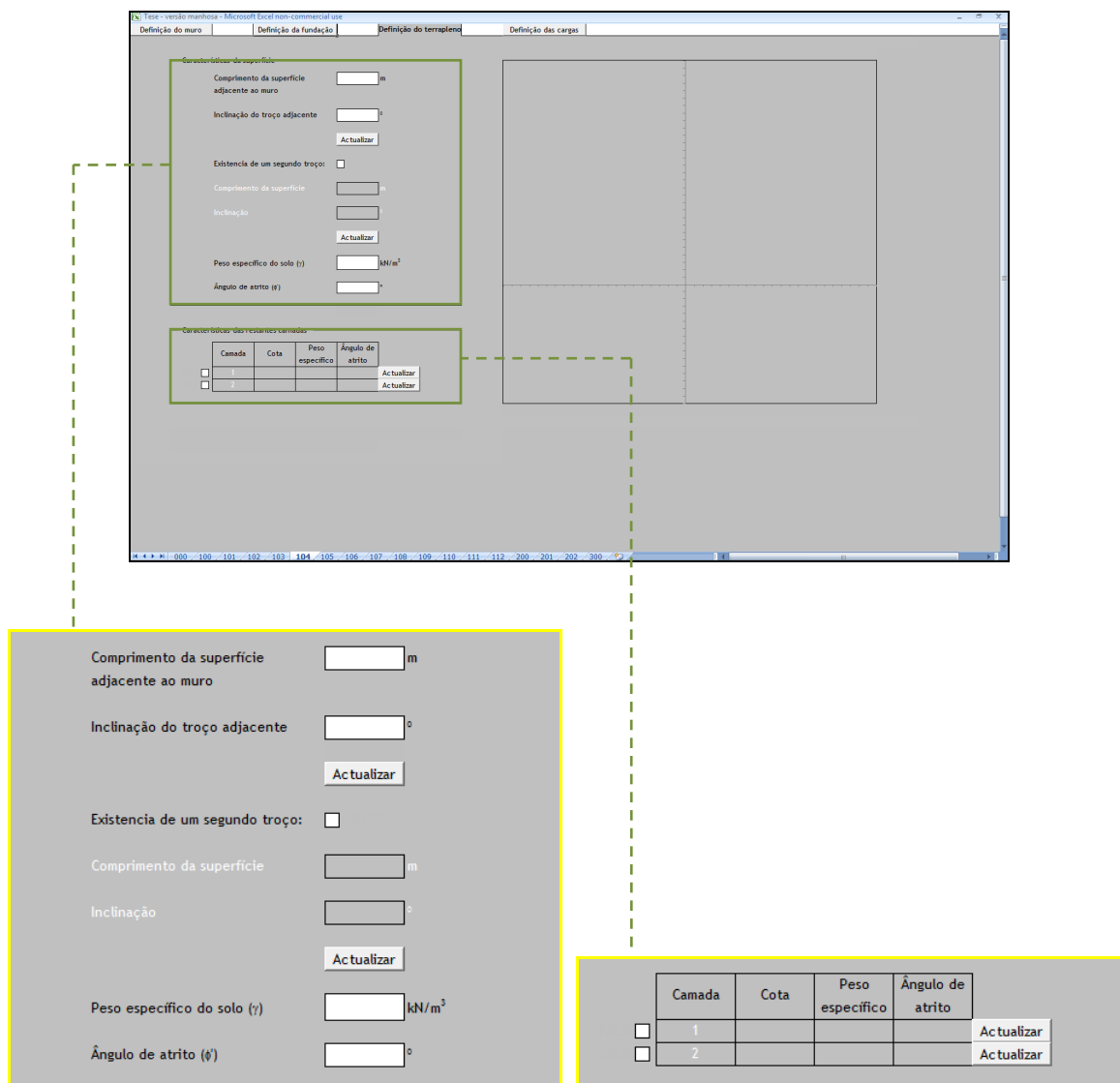


Fig. 3.4. – Representação do menu “*Definição do terrapleno*”.

A partir deste menu, é possível definir a geometria do terrapleno através da introdução do comprimento do terrapleno, da inclinação, da adição de um segundo tramo à superfície e respectivos dados.

Ainda neste menu é possível dividir o terrapleno em 3 sub-camadas. Após a activação dessas sub-camadas é necessário preencher os dados referentes à cota, peso volúmico e ângulo de atrito. A não inclusão de coesão e inclinação nestes novos estratos deve-se ao facto de não serem horizontes naturais. Como tal, a granulometria e a inclinação são escolhidas para um melhor desempenho e maior facilidade de construção respectivamente.

### 3.6. DEFINIÇÃO DAS ACÇÕES

Este quinto menu permite a introdução de acções que actuam sobre a estrutura.

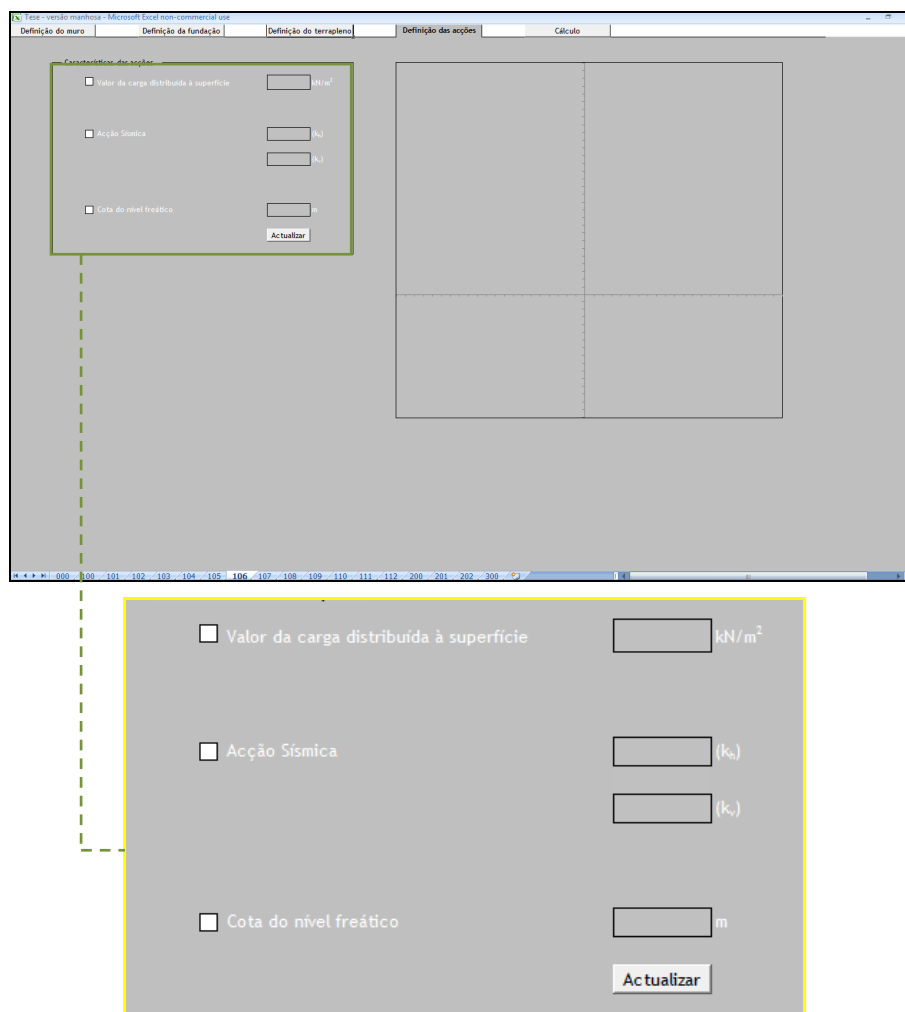


Fig. 3.5. – Representação do menu “Definição das acções”.

Neste menu, pode-se activar a aplicação de uma carga distribuída vertical ao longo da superfície. Esta carga vai gerar um impulso extra ao longo de toda a estrutura.

É possível também adicionar uma componente sísmica aos impulsos através da activação do segundo parâmetro deste menu, sendo neste caso obrigatório o preenchimento dos espaços referentes aos coeficientes sísmicos.

A última opção deste menu consiste em adicionar a acção hidrostática no terrapleno.

### 3.7. CÁLCULO

Este último menu, ilustrado na figura 3.6., permite seleccionar o caso do Eurocódigo 7 que se pretende adoptar para a verificação do dimensionamento, bem como a redução do ângulo de atrito na interacção terras-muro. Deve também ser seleccionado o método pelo qual se pretende fazer o estudo da estabilidade global.

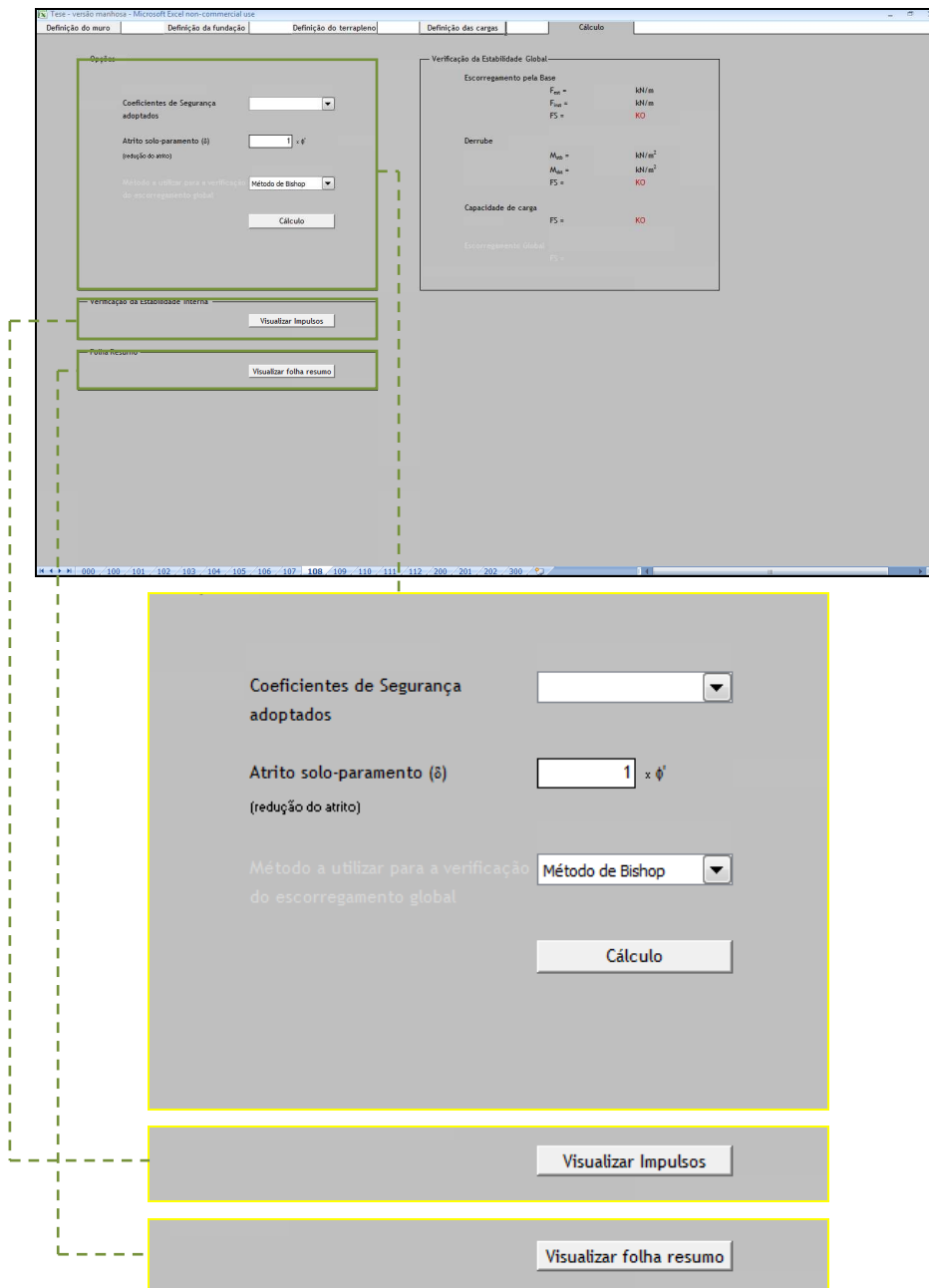


Fig. 3.6. – Representação do menu "Cálculo".

Após a introdução destes três parâmetros o utilizador deverá pressionar o botão "Calcular" de modo a que o programa inicie o cálculo das diversas solicitações (impulsos).

O programa executa uma série de tarefas de modo a identificar quais as camadas presentes e quais os impulsos gerados por essas camadas. É também calculada a componente do peso do muro a partir da geometria e características do mesmo. Após ter corrido essas tarefas, o programa apresenta os resultados da verificação da segurança no quadro da direita.

No caso de estruturas como os muros de gabiões é necessário saber se ocorre escorregamento ou derrube entre alguma das camadas. Como tal, o utilizador poderá carregar no botão “Visualizar impulsos” para saber o comportamento da estrutura em cada uma das camadas e a resultante dos impulsos aplicadas na base de cada camada de blocos.

No caso de os resultados serem satisfatórios para o utilizador, este poderá aceder facilmente aos menus anteriores e alterar as características que julgar mais adequadas para solucionar o problema. No entanto, quando o utilizador recuar nas etapas, não deverá esquecer de carregar em todos os botões obrigatórios de actualização de dados (tal como fez na primeira vez que correu o programa).

Após a obtenção dos resultados pretendidos, o utilizador poderá imprimir uma folha resumo que incluirá todos os dados fornecidos assim como os resultados da verificação da segurança.





# 4

## EXEMPLO PRÁTICO

### 4.1. GENERALIDADES

Este capítulo aborda um exemplo prático de aplicação do programa. O capítulo abrange não só o preenchimento dos diversos campos dos menus, mas também descreve as folhas de cálculo auxiliares que não são visualizadas durante a utilização normal do programa.

Com este capítulo pretende-se facilitar a compreensão dos procedimentos a levar a cabo para uma correcta utilização do programa.

### 4.2. EXEMPLO

Ao abrir o ficheiro, o utilizador terá que alterar as definições do Microsoft® Office Excel de modo a permitir o uso de macros, bem como da função “Solve”, tanto nas folhas de cálculo como na aplicação de VBA.

O exemplo a apresentar consiste num muro de gabiões, como tal, no menu inicial selecciona-se a opção “Muro de Gabiões”.

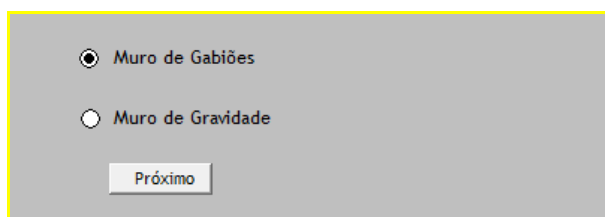


Fig. 4.1. – Selecção do tipo de muro.

No menu “Definição do muro” tem que se preencher os dados referentes à geometria do muro. Neste caso o muro vai ter uma inclinação de 6° e um peso específico de 17 kN/m<sup>3</sup>.

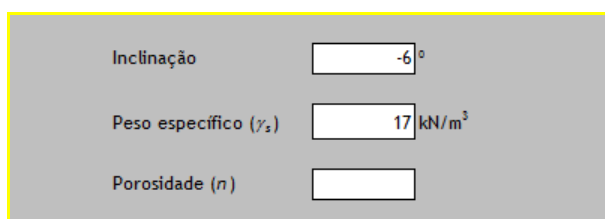


Fig. 4.2. – Introdução das características do muro.

Seguidamente, preenchem-se os campos das camadas, adicionando uma camada de cada vez.

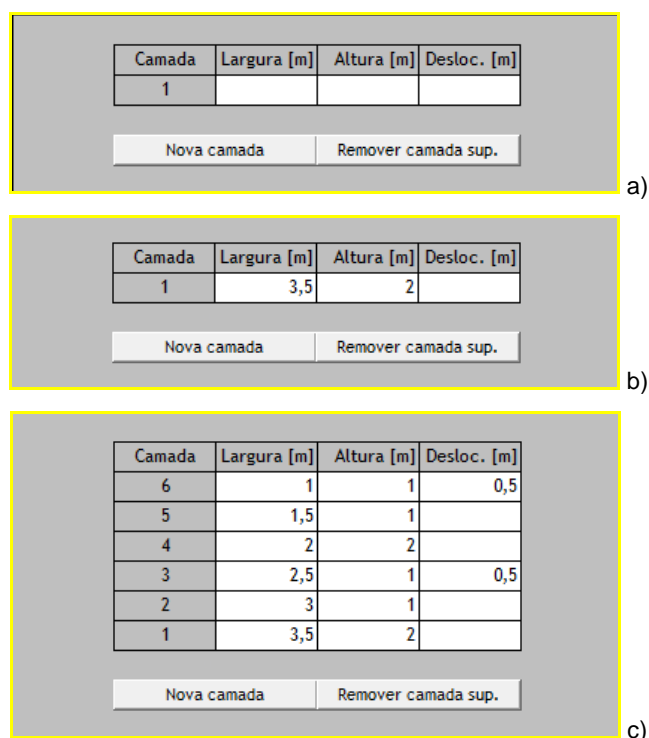


Fig. 4.3. – Etapas de introdução da geometria do muro.

A geometria final do muro é apresentada no gráfico que acompanha os quadros de introdução de dados. Durante a inserção dos valores, é criada numa folha anexa uma lista com as coordenadas dos vértices de cada camada de blocos. É a partir dessa listagem de coordenadas que se gera o gráfico do muro.

Ainda nessa folha auxiliar existe uma tabela que servirá para definir os limites do gráfico no menu seguinte.

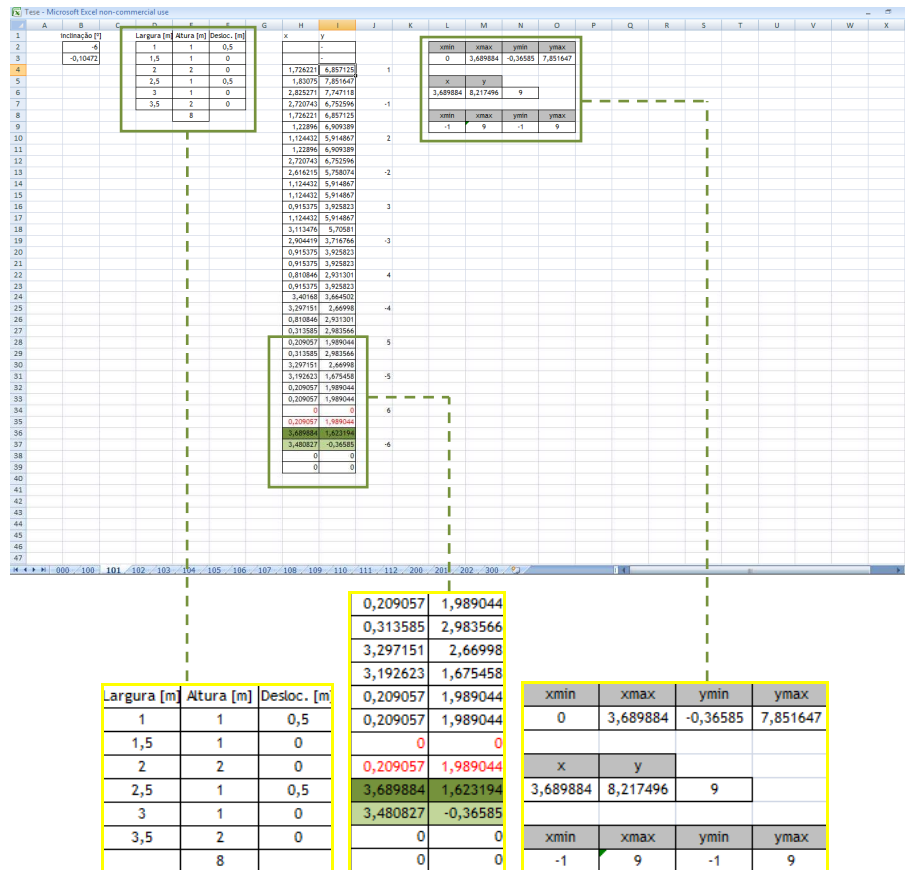


Fig. 4.4. – Folha anexa para definição das características do muro.

Após a introdução dos dados do muro, passa-se para o menu “Definição da fundação” através da barra de menus. Nesta nova folha introduzem-se os dados que caracterizam a fundação. No corrente exemplo, adopta-se um comprimento da superfície à frente do muro de 20 metros, uma inclinação de 0° e uma cota em relação à base de 2 metros. Para visualizar se os dados foram inseridos correctamente e de forma que o programa calcule o ponto de intersecção do plano do terreno com o muro, é fundamental que se pressione o botão “Actualizar dados”.

— Características da superfície —

Comprimento da superfície adjacente ao muro  m

Inclinação do troço adjacente  °

Altura em relação à base  m

Fig. 4.5. – Introdução de dados da superfície.

Ainda nesta folha, existe a possibilidade de se adicionar a presença de um nível freático no terreno a suster através da activação dessa opção. No caso de se activar é necessário inserir a cota a que se pretende ter o nível freático. Como a esta nova recta não é dada inclinação, o utilizador terá que

verificar se a recta que corresponde à superfície intersecta esta última. Existe também a possibilidade de optar por uma fundação em condições drenadas ou não drenadas e quantificar a resistência não drenada. Novamente, é fundamental que se pressione o botão “*Actualizar dados*”.

a)

b)

Fig. 4.6. – Introdução de dados do nível freático.

A introdução das características mecânicas do solo protegido é efectuada de seguida admitindo-se neste caso um peso específico de  $20 \text{ kN/m}^3$ , um ângulo de atrito de  $33^\circ$  e uma coesão de  $10 \text{ kPa}$ .

Fig. 4.7. – Introdução das características mecânicas da fundação.

Fig. 4.7. – Introdução das características mecânicas da fundação.

A última opção deste menu diz respeito à introdução de novos horizontes na fundação. Para tal, basta inserir os valores da cota, peso específico, ângulo de atrito e coesão e pressionar o botão “*Adicionar uma camada*”, caso se pretenda mais algum estrato.

Camada	Cota	Peso específico	Ângulo de atrito	Coesão
1	-5	25	30	5

Adicionar uma camada      Remover camada

Fig. 4.8. – Introdução dos dados de um novo estrato.

Após a introdução de todos estes dados, e antes de avançar para o menu seguinte, o gráfico apresenta-se com o seguinte aspecto:

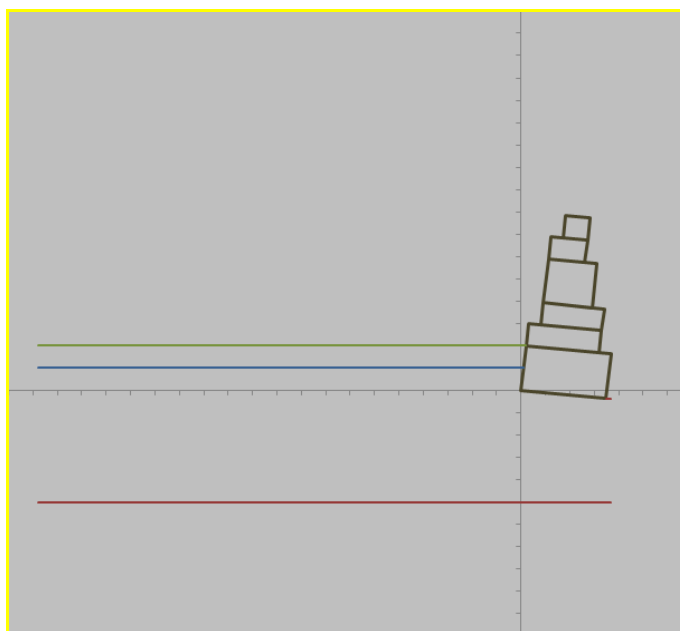


Fig. 4.9. – Aspecto do gráfico após introdução dos dados da fundação.

Já no menu “*Definição do terrapleno*” introduzem-se as características do solo a proteger. Para o exemplo actual, admite-se um comprimento da superfície de 20 metros e uma inclinação de 20°. Como peso específico e ângulo de atrito adoptam-se os valores de 20 kN/m<sup>3</sup> e 20° respectivamente.

Características da superfície	
Comprimento da superfície adjacente ao muro	<input type="text" value="20"/> m
Inclinação do troço adjacente	<input type="text" value="20"/> °
	<input type="button" value="Actualizar"/>
Existencia de um segundo troço:	<input type="checkbox"/>
Comprimento da superfície	<input type="text"/> m
Inclinação	<input type="text"/> °
	<input type="button" value="Actualizar"/>
Peso específico do solo ( $\gamma$ )	<input type="text" value="20"/> kN/m <sup>3</sup>
Ângulo de atrito ( $\phi$ )	<input type="text" value="30"/> °

Fig. 4.10. – Introdução das características da superfície do terrapleno.

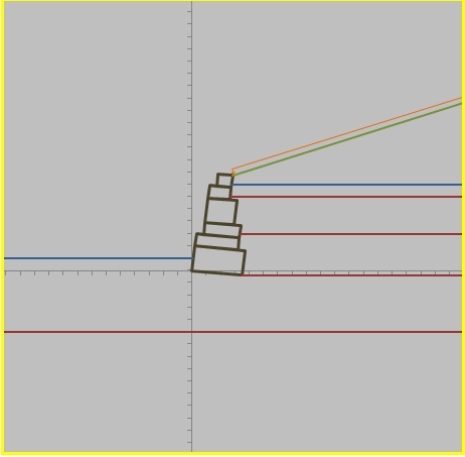
Para tornar este exemplo bastante abrangente, divide-se o terrapleno e acrescentam-se mais duas camadas. Para tal, basta seleccionar as duas camadas disponíveis e introduzir as características geométricas e mecânicas, tal como apresenta a figura 4.11.

Características das restantes camadas				
	Camada	Cota	Peso específico	Ângulo de atrito
<input checked="" type="checkbox"/>	1	6	20	25 Actualizar
<input checked="" type="checkbox"/>	2	3	20	30 Actualizar

Fig. 4.11. – Introdução das características da superfície do terrapleno.

Passando para o menu seguinte, o utilizador tem a oportunidade de adicionar acções que actuam sobre a estrutura. Desta forma, adopta-se uma carga distribuída de  $20\text{kN/m}^2$ , presença de acção sísmica com adopção de coeficientes sísmicos horizontal e vertical de valor 0,08 e 0,04 respectivamente e ainda a presença de nível freático a uma cota de 7 metros no terrapleno.

Características das acções	
<input checked="" type="checkbox"/> Valor da carga distribuída à superfície	<input type="text" value="20"/> $\text{kN/m}^2$
<input checked="" type="checkbox"/> Acção Sísmica	<input type="text" value="0,08"/> ( $k_h$ )
	<input type="text" value="0,04"/> ( $k_v$ )
<input checked="" type="checkbox"/> Cota do nível freático	<input type="text" value="7"/> m
	<input type="button" value="Actualizar"/>



O gráfico auxiliar à direita mostra uma representação técnica de um muro de suporte de terras. O muro é representado por uma linha vertical com uma inclinação para o lado da terra. Acima do muro, há uma superfície inclinada representada por uma linha verde e laranja. Abaixo do muro, há uma linha horizontal vermelha que representa o nível freático. O fundo do gráfico é dividido em camadas horizontais, representando as camadas de terrapleno.

Fig. 4.12. – Introdução das acções e aspecto do gráfico auxiliar.

No menu “Cálculo”, último menu do programa, escolhe-se o tipo de coeficientes a adoptar segundo o Eurocódigo7, a redução do atrito solo-paramento e o tipo de metodologia a utilizar para cálculo do coeficiente de segurança global. No caso em estudo adopta-se o caso 2 do EC7, inexistência de redução de atrito.

Opções

Coeficientes de Segurança adoptados

EC7 Caso 2

EC7 Caso 1

EC7 Caso 2

Atrito solo-paramento ( $\delta$ )  
(redução do atrito)

1 x  $\phi$

Método a utilizar para a verificação do escorregamento global

Método de Bishop

Cálculo

Fig. 4.13. – Introdução das opções de cálculo.

Após a correcta introdução destes valores, pressiona-se o botão cálculo e é executada uma rotina que calcula os impulsos e as diversas componentes do peso do muro. Todos esses valores calculados estão registados numa folha anexa que se apresenta nas imagens seguintes:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
1																									
2		$\gamma_s$	17	kN/m <sup>3</sup>		$\phi$														$Z_{se}$					
3		Canadas	6		$\alpha$	0,10472	rad	$\beta$	0,349066		$\lambda$			$\theta$					$H_{max}$						
4																									
5		Canada i	W (kN)	x	y	$W_{cy}$	$x \cdot W_{cy}$	$W_{ix}$	$y \cdot W_{ix}$	$(W \cdot k_x)$	$(W \cdot k_y)$	$(W \cdot k_x)$	$(W \cdot k_y)$	$(W \cdot k_x)$	$(W \cdot k_y)$	$(W \cdot k_x)$	$(W \cdot k_y)$	$k_{\phi}$							
6																									
7		6	17	2,275746	7,302122	16,90687	38,47575	1,776984	12,97575	0,142159	1,03806	1,35255	3,07806	0,071079	0,51903	0,676275	1,53903								
8		5	25,5	1,922587	6,33732	25,36031	48,75741	2,665476	16,88241	0,213238	1,350993	2,028825	3,900593	0,106619	0,675296	1,014412	1,950296								
9		4	68	2,014425	4,815817	67,62749	136,2305	7,107936	34,23052	0,568635	2,738441	5,410199	10,89844	0,284317	1,369221	2,7051	5,449221								
10		3	42,5	2,106263	3,297902	42,26718	89,0258	4,44246	14,6508	0,355397	1,172064	3,381374	7,122064	0,177698	0,886032	1,690687	3,561032								
11		2	51	1,753104	2,329512	50,72062	88,91852	5,330952	12,41852	0,426476	0,993481	4,057649	7,113481	0,213238	0,496741	2,028825	3,556741								
12		1	119	1,844942	0,811597	118,3481	218,3454	12,43889	10,09536	0,995111	0,807629	9,467848	17,46763	0,497555	0,403815	4,733924	8,733815								
13		0																							
14		$\Sigma$	323	1,92931	2,998971	321,2306	619,7534	33,76269	101,2534	2,701015	8,100268	25,69845	49,58027	1,350508	4,050134	12,84922	24,79013								
15																									
16																									
17																									
18																									
19																									
20																									
21																									
22																									
23																									
24																									
25																									
26																									
27																									
28																									
29																									
30																									
31																									
32																									
33																									
34																									
35																									
36																									
37																									
38																									
39																									
40																									
41																									
42																									
43																									
44																									
45																									
46																									

Fig. 4.14. – Folha anexa com os cálculos das diversas componentes do peso.



	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
1						390,9733																		
2			x	y																				
3		cam sup	2,83	7,75																				
4		1ª cam	2,64	6,00	TRUE																			
5		2ª cam	2,64	6,00	TRUE																			
6		NF	2,75	7,00	TRUE																			
7		fund	3,48	-0,37																				
8		x	0,21	2,00																				
9		y	0,11	1,00	TRUE																			
10			3,480827	-0,36585																				
11			2,825271	7,747118																				
12																								
13																								
14		SQ																						
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22																								
23																								
24																								
25																								
26																								
27																								
28																								
29																								
30																								
31																								
32																								
33																								
34																								
35																								
36																								
37																								
38																								
39																								
40																								
41																								
42																								
43																								
44																								
45																								
46																								

Fig. 4.15. – Folha anexa com o cálculo dos diversos impulsos.

Os resultados são apresentados no quadro “Verificação da segurança”, no qual estão expostos os factores de segurança ao escorregamento, derrube e capacidade de carga. Neste caso, o muro não verifica a segurança ao escorregamento pela base nem à capacidade de carga.

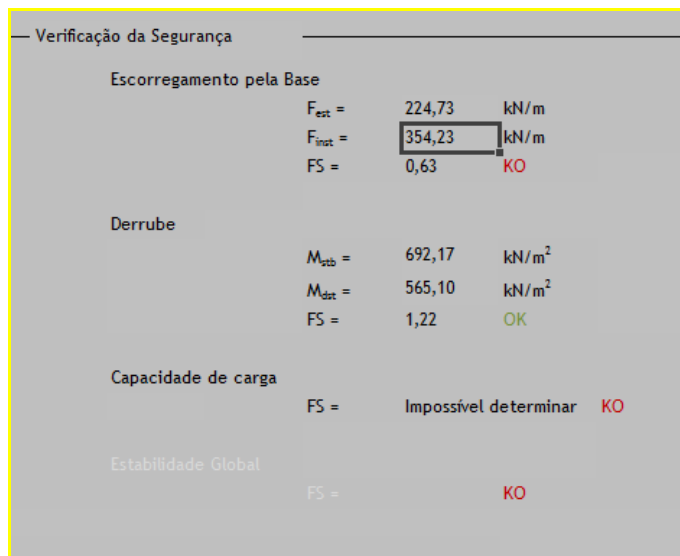


Fig. 4.16. – Quadro de resultados.

No caso de se pretender analisar a estabilidade em cada uma das camadas de blocos, terá que se pressionar o botão “Visualizar Impulsos” e o programa irá executar uma rotina que calcula a estabilidade para cada um dos conjuntos de blocos. Esta rotina só acaba quando for apresentado ao utilizador um gráfico com a estrutura, a resultante em cada uma das camadas e ainda a verificação de segurança ao derrube e escorregamento para as mesmas.

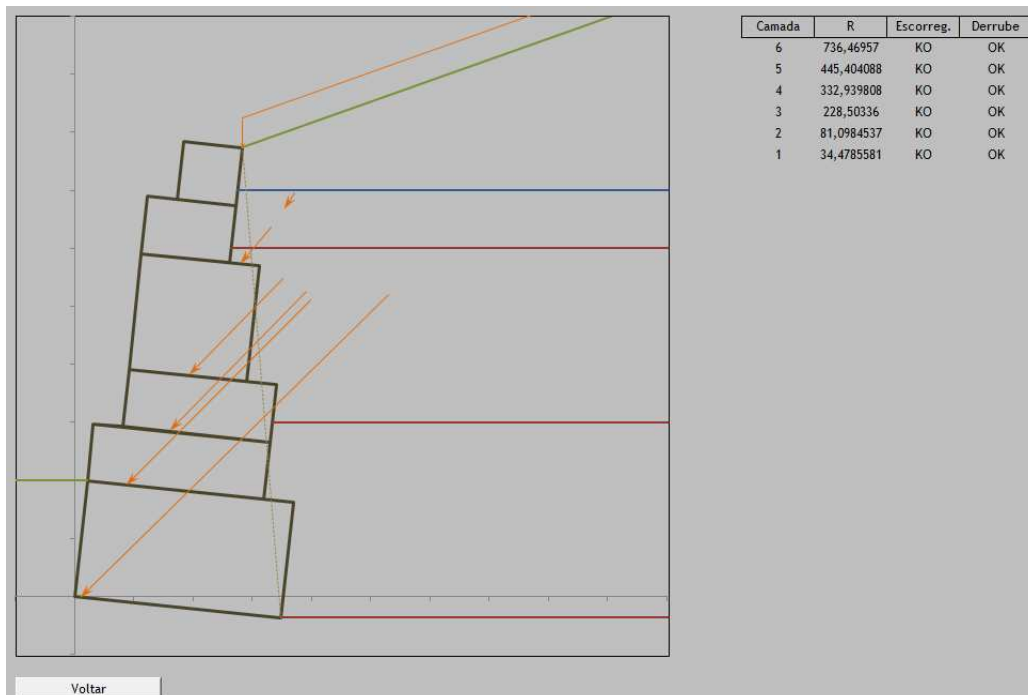


Fig. 4.17. – Apresentação dos resultados por camada.

Para finalizar, o programa fornece a possibilidade de imprimir um resumo da verificação da segurança efectuada. Para tal, basta seleccionar o botão “Visualizar folha resumo” e será apresentada uma folha com todos os valores inseridos pelo utilizador no programa e os resultados obtidos através de cálculo.



# 5

## FUTURAS ACTIALIZAÇÕES

No início da elaboração do presente trabalho, o objectivo era elaborar uma ferramenta de cálculo automático que dimensionasse todo o tipo de muros de suporte. Durante a elaboração do mesmo, concluiu-se que o curto tempo disponível não era suficiente para a elaboração de um trabalho tão vasto. Como tal, o programa ficou limitado ao dimensionamento de muros de gabiões. Uma das actualizações prioritárias deste programa é a inclusão de um módulo para o dimensionamento de muros de gravidade. Esse novo módulo não vai diferir muito do módulo já elaborado para o muro de gabiões, o único menu que irá ser elaborado de raiz é o menu “*Definição do Muro*”, visto que o novo tipo de muro compreenderá blocos triangulares e não só rectangulares. Esta actualização irá tornar o campo de aplicação do programa mais vasto.

Uma outra actualização a elaborar será a conclusão da rotina de cálculo do coeficiente de estabilidade global. O programa já integra os espaços de introdução dos dados e saída de resultados, no entanto, novamente por questões técnicas e de tempo, não foi possível terminar esta rotina em tempo útil. Como se trata de uma rotina já trabalhada, será de breve conclusão e será também uma mais-valia para o programa.

Apesar de se ter testado o programa para várias situações e já se terem efectuado algumas correcções, não é garantido que o programa não tenha falhas., Como tal, se forem detectadas novas falhas, o programa será sujeito a uma actualização.



## BIBLIOGRAFIA

Eurocódigo 7 (1994). *Projecto Geotécnico – Parte 1. Regras Gerais*. Pré-norma europeia, ENV 1997-1:1999 PT. Comité Europeu de Normalização, Bruxelas

Matos Fernandes, M. (1990). *Estruturas de Suporte de Terras*, Textos de apoio, FEUP.

Matos Fernandes, M. (1990). *Mecânica dos Solos – II Volume*, Textos de apoio, FEUP.

Maccaferri. *Gawac R.2.0 User's Manual*.

Look, Burt (1994). *Spreadsheet Geomechanics*.

Peres, Paula. *Excel Avançado*, Edições Sílabo, 2005, 1ª edição.





## **ANEXOS**

Sub Inicio()

```
'=====
' 000 - escolher o tipo de muro
'=====
'
  If Range("C9").Value = 1 Then
    Range("C9").Select
    Selection.ClearContents
    Sheets("100").Select
  End If
  If Range("C9").Value = 2 Then
    Range("C9").Select
    Selection.ClearContents
    Sheets("200").Select
  End If
End Sub
```

---

Sub muro()

```
'=====
' 100 - ir para a folha 100
'=====
  Sheets("100").Select
End Sub
```

---

Sub camadas()

```
'=====
' 100 - adicionar camadas
'=====
  altura = Range("e15").Value
  largura = Range("d15").Value
  If altura <> 0 And largura <> 0 Then
' introdução de linha na folha 100
    ActiveWindow.SmallScroll Down:=-12
    Rows("15:15").Select
    Selection.Insert Shift:=xlDown
    Range("C16").Select
    Selection.AutoFill Destination:=Range("C15:C16"), Type:=xlFillDefault
    Range("C15:C16").Select
    Range("D15:F15").Select
    Selection.Interior.ColorIndex = 2
    Selection.Locked = False
    Selection.FormulaHidden = False
    Range("A1:B1").Select
```

```

' dados do gráfico
Sheets("101").Select
Range("D2:F2").Select
Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
Range("D2").Select
ActiveCell.FormulaR1C1 = "="&100!R[13]C"
Range("E2").Select
ActiveCell.FormulaR1C1 = "="&100!R[13]C"
Range("F2").Select
ActiveCell.FormulaR1C1 = "="&100!R[13]C"
Range("F3").Select
Range("H4:I9").Select
Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
Range("H9").Select
ActiveCell.FormulaR1C1 = "=R[2]C"
Range("I9").Select
ActiveCell.FormulaR1C1 = "=R[2]C"
Range("H8").Select
ActiveCell.FormulaR1C1 = "=R[1]C+R[-6]C[-2]*COS(RADIANS(R[-6]C[-6]))"
Range("I8").Select
ActiveCell.FormulaR1C1 = "=R[1]C+R[-6]C[-3]*SIN(RADIANS(R[-6]C[-7]))"
Range("H7").Select
ActiveCell.FormulaR1C1 = "=R[2]C+(R[-5]C[-2]+R[-5]C[-4])*COS(RADIANS(R[-5]C[-6]))"
Range("I7").Select
ActiveCell.FormulaR1C1 = "=R[2]C+(R[-5]C[-3]+R[-5]C[-5])*SIN(RADIANS(R[-5]C[-7]))"
Range("H6").Select
ActiveCell.FormulaR1C1 = "=R[1]C-R[-4]C[-3]*SIN(RADIANS(R[-4]C[-6]))"
Range("I6").Select
ActiveCell.FormulaR1C1 = "=R[1]C+R[-4]C[-4]*COS(RADIANS(R[-4]C[-7]))"
Range("H5").Select
ActiveCell.FormulaR1C1 = "=R[-1]C-R[-3]C[-3]*SIN(RADIANS(R[-3]C[-6]))"
Range("I5").Select
ActiveCell.FormulaR1C1 = "=R[-1]C+R[-3]C[-4]*COS(RADIANS(R[-3]C[-7]))"
Range("H4").Select
ActiveCell.FormulaR1C1 = "=R[4]C"
Range("I4").Select
ActiveCell.FormulaR1C1 = "=R[4]C"
Range("105!F4").Value = Range("101!H6").Value
Range("105!G4").Value = Range("101!I6").Value
Sheets("100").Select
Else
MsgBox "Tem que introduzir obrigatoriamente os dados referentes à altura e à largura.", vbOKOnly, "Erro"
End If
End Sub

```

Sub rcamada()

'=====

' 100 - remover camadas

'=====

camada = Range("c15").Value

If camada <> 1 Then

' remoção da linha de introdução de dados

Rows("15:15").Select

Selection.Delete Shift:=xlUp

Range("A1:B1").Select

' remoção das linhas de dados do gráfico

Sheets("101").Select

Range("D2:F2").Select

Selection.Delete Shift:=xlUp

Range("H4:I9").Select

Selection.Delete Shift:=xlUp

Range("'105!F4").Value = Range("'101!H6").Value

Range("'105!G4").Value = Range("'101!I6").Value

Sheets("100").Select

Else

MsgBox "O muro deverá ter obrigatoriamente uma camada.", vbOKOnly, "Erro"

End If

End Sub

---

Sub fundação()

'=====

' 102 - ir para a folha 102

'=====

If Range("'100!E5").Value = "" Or Range("'100!E7").Value = "" Then

MsgBox "Terá que preencher obrigatoriamente todos os campos.", vbOKOnly, "Erro"

Else

Sheets("102").Select

End If

End Sub

Sub fund1()

```
'=====
' 102 - condições drenadas ou n drenadas?
'=====
```

```
If Range("F15").Value = False Then
    Range("C17:G19").Select
    Selection.Font.ColorIndex = 2
    Range("F17:F19").Select
    Selection.Interior.Pattern = x1Solid
    Selection.Interior.ColorIndex = 15
    Range("F17").Select
    Selection.ClearContents
    Range("F19").Select
    Selection.ClearContents
    Range("'103!F5").Value = ""
    Range("'103!F6").Value = ""
    Range("'110!H7").Value = 1
    Range("'110!D7").Value = 0
    Range("D1:E1").Select
```

Else

```
    Range("C17:G19").Select
    Selection.Font.ColorIndex = 0
    Range("F17").Select
    Selection.Interior.Pattern = x1Solid
    Selection.Interior.ColorIndex = 2
    Range("F19").Select
    Selection.Interior.Pattern = x1Solid
    Selection.Interior.ColorIndex = 2
    Range("'110!H7").Value = 0
    Range("'110!D7").Value = 1
    Range("D1:E1").Select
```

End If

End Sub

---

Sub fund\_apagar()

```
'=====
' 102 - apagar dados - características da superfície
'=====
```

```
Range("F5").Select
Selection.ClearContents
Range("F8").Select
Selection.ClearContents
```

```

Range("F10").Select
Selection.ClearContents
Range("D1:E1").Select
Sheets("103").Select
Range("F14:G15").Select
Selection.ClearContents
Sheets("102").Select
End Sub

```

---

```

Sub fund_act()

```

```

'=====

```

```

' 102 - actualizações dos dados das características da superfície

```

```

'=====

```

```

Sheets("103").Select
alt_fund = Range("D2").Value
alt_max = Range("D5").Value
alt_1 = Range("'101!I5").Value
linha = 5
Click = Range("'100!C15").Value

```

```

If Range("'100!E5").Value < 0 Then

```

```

    If alt_fund > alt_max Then

```

```

        MsgBox "A cota do terreno tem que ser inferior à cota máxima do muro.", vbOKOnly, "Erro"

```

```

    Else

```

```

        While alt_1 > alt_fund

```

```

            linha = linha + 6

```

```

            alt_1 = Range("'101!I" & linha).Value

```

```

        Wend

```

```

        linha = linha - 6

```

```

        alt_1 = Range("'101!I" & linha).Value

```

```

        coord = Range("'101!H" & linha).Value + (alt_1 - alt_fund) * Tan(Range("B8").Value)

```

```

        Range("F15").Value = coord

```

```

        linha = linha + 6

```

```

        passo = 1

```

```

        dist1 = 0

```

```

        Range("F14").Value = coord

```

```

        Range("G14").Value = alt_fund

```

```

    While passo < Click

```

```

        linhaa = linha - 3

```

```

        dist = Range("'101!h" & linhaa).Value - Range("'101!h" & linha).Value

```

```
dist = dist / Cos(Range("B8").Value)
dist1 = dist1 + dist

h = dist1 / Cos(Range("C8").Value)
xint = coord - h * Cos(Range("C3").Value)
yint = alt_fund - h * Sin(Range("C3").Value)

If yint < Range("'101'!" & linha).Value Then
    Range("F14").Value = xint
    Range("G14").Value = yint
End If

linha = linha + 6
passo = passo + 1
Wend

End If

Else

If alt_fund > alt_max Then
    MsgBox "A cota do terreno tem que ser inferior à cota máxima do muro.", vbOKOnly, "Erro"
Else

While alt_1 > alt_fund
    linha = linha + 6
    alt_1 = Range("'101'!" & linha).Value
Wend
linha = linha - 6
linha2 = linha - 1
linha3 = linha + 6
alt_1 = Range("'101'!" & linha).Value
alt_2 = Range("'101'!" & linha2).Value
alt_3 = Range("'101'!" & linha3).Value

If alt_fund > alt_2 Then

coord = Range("'101'H" & linha).Value + (alt_1 - alt_fund) * Tan(Range("B8").Value)
Range("F15").Value = coord
Range("F14").Value = coord
Range("G14").Value = alt_fund

Else
```

```

coord = Range("'101!H" & linha2).Value - (alt_2 - alt_fund) / Tan(Range("B8").Value)
Range("F15").Value = coord
Range("F14").Value = coord
Range("G14").Value = alt_fund
End If

passo = 2
passo2 = 1
dist = 0
dist1 = 0
dist2 = 0
xint = coord
yint = alt_fund
While passo < Click
    dist = Range("'101!h" & linha3).Value - coord
    haval = alt_fund + dist * Tan(Range("C3").Value)
    If haval < Range("'101!l" & linha3).Value Then
        While passo2 = 1
            If alt_fund > alt_2 Then
                dist2 = (Range("'101!h" & linha2).Value - Range("'101!h" & linha3).Value) / Cos(Range("B8").Value)
                hip = dist2 / Cos(Range("C8").Value)
                xint = coord - hip * Cos(Range("C3").Value)
                yint = alt_fund - hip * Sin(Range("C3").Value)
            Else
                dist2 = (Range("'101!h" & linha3).Value - coord) / Cos(Range("B8").Value)
                hip = dist2 / Cos(Range("C8").Value)
                xint = coord + hip * Cos(Range("C3").Value)
                yint = alt_fund + hip * Sin(Range("C3").Value)
            End If
            Range("F14").Value = xint
            Range("G14").Value = yint
            passo2 = passo2 + 1
        Wend
    End If
    linha = linha + 6
    linha2 = linha - 1
    linha3 = linha + 6
    alt_1 = Range("'101!l" & linha).Value
    alt_2 = Range("'101!l" & linha2).Value
    alt_3 = Range("'101!l" & linha3).Value
    passo = passo + 1
    Range("G16").Value = passo

```



```

dist = Range("'101!h" & linha).Value - xint
haval = yint + dist * Tan(Range("C3").Value)

If haval < Range("'101!!" & linha3).Value And (coord - xint) / Sin(Range("C3").Value) < Range("B2").Value Then
dist2 = (Range("'101!h" & linha2).Value - Range("'101!h" & linha3).Value) / Cos(Range("B8").Value)
hip = dist2 / Cos(Range("C8").Value)
xint = xint - hip * Cos(Range("C3").Value)
yint = yint - hip * Sin(Range("C3").Value)
Range("F14").Value = xint
Range("G14").Value = yint
End If

Wend
End If
End If
' aranjando o gráfico
Sheets("102").Select
ActiveSheet.ChartObjects("Chart 6").Activate
ActiveChart.Axes(xlValue).MinimumScale = Range("'103!D30").Value
ActiveChart.Axes(xlValue).MaximumScale = Range("'103!E30").Value
ActiveChart.Axes(xlCategory).MinimumScale = Range("'103!B30").Value
ActiveChart.Axes(xlCategory).MaximumScale = Range("'103!C30").Value
End Sub

```

---

```

Sub cam_rem()
'=====
' 102 - remover camada
'=====

If Range("C38").Value > 1 Then
Rows("38:38").Select
Selection.Delete Shift:=xlUp
Range("D38").Select

Sheets("103").Select
Range("I3:J5").Select
Selection.Delete Shift:=xlUp
Sheets("102").Select
Else
MsgBox "Não existe camada de possível remoção.", vbOKOnly, "Erro"
End If
Range("'200!D11").Value = "=IF('102!D38=""",0,'102!C38)"
End Sub

```

```
Sub cam_add()
```

```
'=====
```

```
' 102 - adicionar uma camada
```

```
'=====
```

```
If Range("D38").Value = 0 Or Range("E38").Value = 0 Then
```

```
    MsgBox "É obrigatório introduzir valores para a cota da camada e para o peso específico.", vbOKOnly, "Erro"
```

```
Else
```

```
    If Range("C38").Value < 5 Then
```

```
        Rows("38:38").Select
```

```
        Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
```

```
        Range("D38:G38").Select
```

```
        Selection.Interior.Pattern = xlNone
```

```
        Range("C38").Select
```

```
        ActiveCell.FormulaR1C1 = "=R[1]C+1"
```

```
        Range("D38").Select
```

```
        With Selection.Validation
```

```
            .Delete
```

```
            .Add Type:=xlValidateDecimal, AlertStyle:=xlValidAlertStop, Operator _
```

```
            :=xlLess, Formula1:="=D39"
```

```
            .IgnoreBlank = True
```

```
            .InCellDropdown = True
```

```
            .InputTitle = ""
```

```
            .ErrorTitle = "Erro"
```

```
            .InputMessage = ""
```

```
            .ErrorMessage = "A cota tem que ser inferior à camada anterior"
```

```
            .ShowInput = True
```

```
            .ShowError = True
```

```
        End With
```

```
        'dados do gráfico
```

```
        Sheets("103").Select
```

```
        Range("I3:J5").Select
```

```
        Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
```

```
        ActiveWindow.SmallScroll Down:=-12
```

```
        Range("I3").Value = "=IF(J3=0,0,I6)"
```

```
        Range("I4").Value = "=IF(J4=0,0,I7)"
```

```
        Range("J3").Value = "'102!D38"
```

```
        Range("J4").Value = "=J3"
```

```
        Sheets("102").Select
```

```
        Range("D38").Value = Range("D39").Value - 1
```

```
        Range("D38").Select
```

```

Else
    MsgBox "Não poderá ter mais de 5 camadas.", vbOKOnly, "Erro"
End If
End If
Range("200!D11").Value = "=IF('102!D38=""",0,'102!C38)"
End Sub

```

---

```

Sub Fund_NF()

```

```

'=====

```

```

' 102 - Adicionar o NF no gráfico

```

```

'=====

```

```

If Range("F15").Value = True Then

```

```

    cotaNF = Range("103!C13").Value

```

```

    ncamadas = Range("100!C15").Value

```

```

    linha = 5 + (ncamadas - 1) * 6

```

```

    inc = Range("101!B3").Value

```

```

    While cotaNF > Range("101!I" & linha).Value

```

```

        linha = linha - 6

```

```

    Wend

```

```

    difcota = Range("101!I" & linha).Value - cotaNF

```

```

    x = Range("101!H" & linha).Value + Tan(inc) * difcota

```

```

    Range("103!F6").Value = "=B24"

```

```

    Range("103!F5").Value = x

```

```

End If

```

```

End Sub

```

---

```

Sub aterro()

```

```

'=====

```

```

' 104 - ir para a folha 104

```

```

'=====

```

```

If Range("102!F5").Value = "" Or Range("102!F8").Value = "" Or Range("102!F10").Value = "" Or Range("102!F25").Value = "" Or Range("102!F27").Value = "" Or Range("102!F29").Value = "" Then

```

```

    MsgBox "Terá que preencher obrigatoriamente todos os campos.", vbOKOnly, "Erro"

```

```

Else

```

```

    Sheets("104").Select

```

```

End If

```

```

End Sub

```

```

Sub ate_act()
'=====
' 104 - atualizar o troço superior
'=====

Sheets("105").Select
Range("F4").Value = "'101!H6"
Range("G4").Value = "'101!I6"
Range("F5").Value = "=F4+D3*COS(RADIANS(D4))"
Sheets("106").Select
ActiveSheet.ChartObjects("Chart 2").Activate
ActiveChart.Axes(xlValue).MinimumScale = Range("'105!D30").Value
ActiveChart.Axes(xlValue).MaximumScale = Range("'105!E30").Value
ActiveChart.Axes(xlCategory).MinimumScale = Range("'105!B30").Value
ActiveChart.Axes(xlCategory).MaximumScale = Range("'105!C30").Value
Sheets("104").Select
Range("F10").Select
ActiveSheet.ChartObjects("Chart 6").Activate
ActiveChart.Axes(xlValue).MinimumScale = Range("'105!D30").Value
ActiveChart.Axes(xlValue).MaximumScale = Range("'105!E30").Value
ActiveChart.Axes(xlCategory).MinimumScale = Range("'105!B30").Value
ActiveChart.Axes(xlCategory).MaximumScale = Range("'105!C30").Value
End Sub

```

---

```

Sub aterr_tro()
'=====
' 104 - adicionar/remover um troço adicional
'=====

If Range("F12").Value = False Then
    Range("C14:G16").Select
    Selection.Font.ColorIndex = 2
    Range("F14:F16").Select
    Selection.Interior.Pattern = xlSolid
    Selection.Interior.ColorIndex = 15
    Range("F14").Select
    Selection.ClearContents
    Range("F16").Select
    Selection.ClearContents
    Sheets("105").Select
    Range("F6").Select
    Selection.ClearContents
    Sheets("104").Select
    Range("G1:H1").Select

```

```
' apagar dados

Else
  Range("C14:G16").Select
  Selection.Font.ColorIndex = 0
  Range("F14").Select
  Selection.Interior.Pattern = x1Solid
  Selection.Interior.ColorIndex = 2
  Range("F16").Select
  Selection.Interior.Pattern = x1Solid
  Selection.Interior.ColorIndex = 2
  Sheets("105").Select
  Range("F6").Value = "=F5+D7*COS(RADIANS(D8))"
  Sheets("104").Select
  Range("G1:H1").Select
End If

End Sub
```

---

```
Sub aterr_1()
'=====
' 104 - adicionar/retirar uma primeira camada no terrapleno
'=====

If Range("B29").Value = False Then
  Range("C29").Select
  Selection.Font.ColorIndex = 2
  Range("D29:F29").Select
  Selection.Interior.Pattern = x1Solid
  Selection.Interior.ColorIndex = 15
  Range("D29").Value = ""
  Range("E29").Value = ""
  Range("F29").Value = ""
  Range("'105!I3").Value = ""
  Range("'105!I4").Value = ""
  Range("'105!J3").Value = ""
  Range("'105!J4").Value = ""
  Range("G1:H1").Select

' apagar dados

Else
  Range("C29").Select
  Selection.Font.ColorIndex = 0
```

```
Range("D29:F29").Select
Selection.Interior.Pattern = x1Solid
Selection.Interior.ColorIndex = 2
Range("D29").Select
Range("'"105!J4").Value = "=C14"
Range("'"105!J5").Value = "=J4"
End If
```

End Sub

---

Sub aterr\_2()

'=====

' 104 - adicionar/retirar uma segunda camada no terrapleno

'=====

```
If Range("B30").Value = True And Range("B29").Value = True Then
```

```
Range("C30").Select
Selection.Font.ColorIndex = 0
Range("D30:F30").Select
Selection.Interior.Pattern = x1Solid
Selection.Interior.ColorIndex = 2
Range("'"105!J7").Value = "=C15"
Range("D30").Select
```

```
Else
```

```
Range("C30").Select
Selection.Font.ColorIndex = 2
Range("D30:F30").Select
Selection.Interior.Pattern = x1Solid
Selection.Interior.ColorIndex = 15
Range("D30").Value = ""
Range("E30").Value = ""
Range("F30").Value = ""
Range("'"105!I7").Value = ""
Range("'"105!J7").Value = ""
Selection.ClearContents
Range("G1:H1").Select
```

```
End If
```

End Sub

---

Sub ate\_act1()

'=====

' 104 - adicionar os dados de uma camada no terrapleno

'=====

```

Sheets("105").Select
alt_cam = Range("C14").Value
linha = 6
linha2 = 7
linha3 = 12

If Range("B18").Value >= 0 Then
    alt_mur = Range("'101!I6").Value
    alt_mur2 = Range("'101!I7").Value
    Click = Range("'100!C15").Value

    While alt_mur > alt_cam
        linha = linha + 6
        linha2 = linha + 1
        alt_mur = Range("'101!I" & linha).Value
        alt_mur2 = Range("'101!I" & linha2).Value
    Wend

    linha = linha - 6
    linha2 = linha + 1
    alt_mur = Range("'101!I" & linha).Value
    coord = Range("'101!H" & linha).Value + (alt_mur - alt_cam) * Tan(Range("B18").Value)
    Range("I3").Value = coord
    Range("I4").Value = coord
Else
    alt_mur = Range("'101!I" & linha).Value
    alt_mur2 = Range("'101!I" & linha2).Value
    alt_mur3 = Range("'101!I" & linha3).Value

    While alt_cam < alt_mur And alt_cam < alt_mur3
        linha = linha + 6
        linha2 = linha2 + 6
        linha3 = linha3 + 6
        alt_mur = Range("'101!I" & linha).Value
        alt_mur2 = Range("'101!I" & linha2).Value
        alt_mur3 = Range("'101!I" & linha3).Value
    Wend

    If alt_cam < alt_mur2 Then
        coord = Range("'101!H" & linha2).Value - (alt_mur2 - alt_cam) / Tan(Range("B18").Value)
    Else
        coord = Range("'101!H" & linha2).Value - (alt_cam - alt_mur2) * Tan(Range("B18").Value)
    End If

```

```
Range("I4").Value = coord
```

```
End If
```

```
Sheets("104").Select
```

```
End Sub
```

---

```
Sub ate_act2()
```

```
'=====
```

```
' 104 - adicionar os dados de uma segunda camada no terraplano
```

```
'=====
```

```
Sheets("105").Select
```

```
alt_cam = Range("C15").Value
```

```
linha = 6
```

```
linha2 = 7
```

```
linha3 = 12
```

```
If Range("B18").Value >= 0 Then
```

```
alt_mur = Range("'101!I6").Value
```

```
alt_mur2 = Range("'101!I7").Value
```

```
Click = Range("'100!C15").Value
```

```
While alt_mur > alt_cam
```

```
linha = linha + 6
```

```
linha2 = linha + 1
```

```
alt_mur = Range("'101!I" & linha).Value
```

```
alt_mur2 = Range("'101!I" & linha2).Value
```

```
Wend
```

```
linha = linha - 6
```

```
linha2 = linha + 1
```

```
alt_mur = Range("'101!I" & linha).Value
```

```
coord = Range("'101!H" & linha).Value + (alt_mur - alt_cam) * Tan(Range("B18").Value)
```

```
Range("I7").Value = coord
```

```
Else
```

```
alt_mur = Range("'101!I" & linha).Value
```

```
alt_mur2 = Range("'101!I" & linha2).Value
```

```
alt_mur3 = Range("'101!I" & linha3).Value
```

```
While alt_cam < alt_mur And alt_cam < alt_mur3
```

```
linha = linha + 6
```

```
linha2 = linha2 + 6
```

```
linha3 = linha3 + 6
```

```
alt_mur = Range("'101!I" & linha).Value
```



```
alt_mur2 = Range("'101!I" & linha2).Value  
alt_mur3 = Range("'101!I" & linha3).Value  
Wend
```

```
If alt_cam < alt_mur2 Then  
    coord = Range("'101!H" & linha2).Value - (alt_mur2 - alt_cam) / Tan(Range("B18").Value)  
Else  
    coord = Range("'101!H" & linha2).Value - (alt_cam - alt_mur2) * Tan(Range("B18").Value)  
End If  
Range("I7").Value = coord  
End If
```

```
Sheets("104").Select  
End Sub
```

---

```
Sub cargas()  
'=====
```

```
' 106 - ir para a folha 106
```

```
'=====
```

```
    Sheets("106").Select  
End Sub
```

---

```
Sub cargasd()  
'=====
```

```
' 106 - activar cardas distribuidas
```

```
'=====
```

```
If Range("B5").Value = False Then  
    Range("C5:H5").Select  
    Selection.Font.ColorIndex = 2  
    Range("G5").Select  
    Selection.Interior.Pattern = x1Solid  
    Selection.Interior.ColorIndex = 15  
    Range("G5").Select  
    Range("'107!K4").Value = 0  
    Selection.ClearContents
```

```
Else  
    Range("C5:H5").Select  
    Selection.Font.ColorIndex = 0  
    Range("G5").Select  
    Selection.Interior.Pattern = x1Solid  
    Selection.Interior.ColorIndex = 2
```

```
Range("107!K4").Value = 0.5
```

```
Range("G5").Select
```

```
End If
```

```
End Sub
```

---

```
Sub cargadas()
```

```
'=====
```

```
' 106 - acção sísmica
```

```
'=====
```

```
If Range("B9").Value = False Then
```

```
Range("C9:H11").Select
```

```
Selection.Font.ColorIndex = 2
```

```
Range("G9:G11").Select
```

```
Selection.Interior.Pattern = x1Solid
```

```
Selection.Interior.ColorIndex = 15
```

```
Range("G9:G11").Select
```

```
Selection.ClearContents
```

```
Else
```

```
Range("C9:H11").Select
```

```
Selection.Font.ColorIndex = 0
```

```
Range("G9").Select
```

```
Selection.Interior.Pattern = x1Solid
```

```
Selection.Interior.ColorIndex = 2
```

```
Range("G11").Select
```

```
Selection.Interior.Pattern = x1Solid
```

```
Selection.Interior.ColorIndex = 2
```

```
Range("G9").Select
```

```
End If
```

```
End Sub
```

---

```
Sub carganf()
```

```
'=====
```

```
' 106 - activar nível freático
```

```
'=====
```

```
If Range("B15").Value = False Then
```

```
Range("C15:H15").Select
```

```
Selection.Font.ColorIndex = 2
```

```
Range("G15").Select
```

```
Selection.Interior.Pattern = x1Solid
```

```
Selection.Interior.ColorIndex = 15
```

```
Range("G15").Select
```

```

Selection.ClearContents
Sheets("107").Select
Range("L16:L17").Select
Selection.ClearContents
Sheets("106").Select
Range("200!D9").Value = 0
Else
Range("C15:H15").Select
Selection.Font.ColorIndex = 0
Range("G15").Select
Selection.Interior.Pattern = x1Solid
Selection.Interior.ColorIndex = 2
Range("G15").Select
End If
End Sub

```

---

```

Sub carga_actnf()

```

```

'=====
' 106 - adicionar o nível freático no gráfico
'=====

```

```

Sheets("107").Select
alt_cam = Range("J14").Value
linha = 6
linha2 = 7
linha3 = 12

```

```

If Range("I18").Value >= 0 Then
alt_mur = Range("101!I6").Value
alt_mur2 = Range("101!I7").Value
Click = Range("100!C15").Value

```

```

While alt_mur > alt_cam
linha = linha + 6
linha2 = linha + 1
alt_mur = Range("101!" & linha).Value
alt_mur2 = Range("101!" & linha2).Value

```

```

Wend

```

```

linha = linha - 6
linha2 = linha + 1
alt_mur = Range("101!" & linha).Value
coord = Range("101!H" & linha).Value + (alt_mur - alt_cam) * Tan(Range("I18").Value)
Range("L16").Value = coord

```

```
Range("L17").Value = Range("103!F9").Value
Range("M16").Value = Range("107!J14").Value
Range("M17").Value = Range("107!J14").Value
Else
alt_mur = Range("101!I" & linha).Value
alt_mur2 = Range("101!I" & linha2).Value
alt_mur3 = Range("101!I" & linha3).Value

While alt_cam < alt_mur And alt_cam < alt_mur3
linha = linha + 6
linha2 = linha2 + 6
linha3 = linha3 + 6
alt_mur = Range("101!I" & linha).Value
alt_mur2 = Range("101!I" & linha2).Value
alt_mur3 = Range("101!I" & linha3).Value
Wend

If alt_cam < alt_mur2 Then
coord = Range("101!H" & linha2).Value - (alt_mur2 - alt_cam) / Tan(Range("I18").Value)
Else
coord = Range("101!H" & linha2).Value - (alt_cam - alt_mur2) * Tan(Range("I18").Value)
End If
Range("L16").Value = coord
Range("L17").Value = Range("103!F9").Value
Range("M16").Value = Range("107!J14").Value
Range("M17").Value = Range("107!J14").Value
End If

'verificar se a cota coincide com a cota de outra camada
If Range("105!C14").Value = Range("107!J14").Value Or Range("105!C15").Value = Range("107!J14").Value Then
Range("200!D9").Value = 0
Else
Range("200!D9").Value = 1
End If
Sheets("106").Select
End Sub

Sub calculo()
'=====
' 108 - ir para a folha 108
'=====

Sheets("108").Select
End Sub
```

---

Sub calculoEG()

'=====

' 108 - Cálculo da estabilidade Global

'=====

Sheets("109").Select

While Range("B7").Value > 1

Range("B7:Q7").Select

Selection.Delete Shift:=xlUp

Wend

Range("C3").Value = "=" & Range("C15").Value

camada = Range("C3").Value

kv = Range("R2").Value

kh = Range("R3").Value

clique = 1

linha1 = camada + 1

linha2 = 4 + 6 \* (camada - 1)

linha3 = 6 + 6 \* (camada - 1)

XG = 0

YG = 0

' inserir a 1ª camada

Wi = Range("D" & linha1).Value \* Range("E" & linha1).Value \* Range("C2").Value

Range("C7").Value = Wi

xx = (Range("H" & linha2).Value + Range("H" & linha3).Value) / 2

Range("D7").Value = xx

yy = (Range("I" & linha2).Value + Range("I" & linha3).Value) / 2

Range("E7").Value = yy

Wiy = Range("C7").Value \* Cos(Range("F3").Value)

Range("F7").Value = Wiy

xxWiy = xx \* Wiy

Range("G7").Value = xxWiy

Wix = Range("C7").Value \* Sin(Range("F3").Value)

Range("H7").Value = Wix

yyWix = yy \* Wix

Range("I7").Value = yyWix

WiKhx = Wi \* kh \* Sin(Range("F3").Value)

Range("J7").Value = WiKhx

yyWiKhx = yy \* WiKhx

Range("K7").Value = yyWiKhx

WiKhy =  $W_i * k_h * \cos(\text{Range}("F3").\text{Value})$

Range("L7").Value = WiKhy

xxWiKhy =  $xx * \text{WiKhy}$

Range("M7").Value = xxWiKhy

WiKvx =  $W_i * k_v * \sin(\text{Range}("F3").\text{Value})$

Range("N7").Value = WiKvx

yyWiKvx =  $yy * \text{WiKvx}$

Range("O7").Value = yyWiKvx

WiKvy =  $W_i * k_v * \cos(\text{Range}("F3").\text{Value})$

Range("P7").Value = WiKvy

xxWiKvy =  $xx * \text{WiKvy}$

Range("Q7").Value = xxWiKvy

XG = XG +  $xx * W_i$

YG = YG +  $yy * W_i$

'inserir as restantes camadas

While clique < camada

Range("B7:Q7").Select

Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

Range("B8").Select

Selection.AutoFill Destination:=Range("B7:B8"), Type:=xlFillDefault

linha1 = linha1 - 1

linha2 = linha2 - 6

linha3 = linha3 - 6

Wi = Range("'" & linha1 & "!" & "D" & linha1).Value \* Range("'" & linha1 & "!" & "E" & linha1).Value \* Range("C2").Value

Range("C7").Value = Wi

xx = (Range("'" & linha2 & "!" & "H" & linha2).Value + Range("'" & linha3 & "!" & "H" & linha3).Value) / 2

Range("D7").Value = xx

yy = (Range("'" & linha2 & "!" & "I" & linha2).Value + Range("'" & linha3 & "!" & "I" & linha3).Value) / 2

Range("E7").Value = yy

Wiy = Range("C7").Value \*  $\cos(\text{Range}("F3").\text{Value})$

Range("F7").Value = Wiy

xxWiy =  $xx * \text{Wiy}$

Range("G7").Value = xxWiy

Wix = Range("C7").Value \*  $\sin(\text{Range}("F3").\text{Value})$

Range("H7").Value = Wix

yyWix =  $yy * \text{Wix}$

Range("I7").Value = yyWix

WiKhx =  $W_i * k_h * \sin(\text{Range}("F3").\text{Value})$

Range("J7").Value = WiKhx

yyWiKhx =  $yy * \text{WiKhx}$

Range("K7").Value = yyWiKhx

```
WiKhy = Wi * kh * Cos(Range("F3").Value)
Range("L7").Value = WiKhy
xxWiKhy = xx * WiKhy
Range("M7").Value = xxWiKhy
WiKvx = Wi * kv * Sin(Range("F3").Value)
Range("N7").Value = WiKvx
yyWiKvx = yy * WiKvx
Range("O7").Value = yyWiKvx
WiKvy = Wi * kv * Cos(Range("F3").Value)
Range("P7").Value = WiKvy
xxWiKvy = xx * WiKvy
Range("Q7").Value = xxWiKvy
XG = XG + xx * Wi
YG = YG + yy * Wi
```

```
clique = clique + 1
Wend
```

```
Range("D4").Value = XG
Range("E4").Value = YG
```

```
Sheets("108").Select
```

```
End Sub
```

```
Sub limpaintern()
```

```
Sheets("112").Select
While Range("B16").Value <> 6
    Range("B16:Z16").Select
    Selection.Delete Shift:=xlUp
```

```
Wend
```

```
End Sub
```

---

```
Sub visuacalc()
```

```
'=====
' 111 - calcular os impulsos para a folha 111
'=====
```

```
Sheets("112").Select
camada = Range("C3").Value
alpha = Range("C5").Value
lameda = Range("C6").Value
teta = Range("C8").Value
kv = Range("C9").Value
kh = Range("C10").Value
```

ncamada = camada

peso = Range("C4").Value

clique = 1

linha1 = 2

linha2 = 4

linha3 = 6

linha4 = 7

XG = 0

YG = 0

fi = Range("I4").Value

gama = Range("I3").Value

$ka = \frac{(\cos(\text{fi} - \text{lameda})^2)}{(\cos(\text{lameda})^2 * \cos(\text{Delta} + \text{lameda}) * (1 + ((\sin(\text{fi} + \text{Delta}) * \sin(\text{fi} - \text{beta})) / (\cos(\text{beta} - \text{lameda}) * \cos(\text{Delta} + \text{lameda))))}^{0.5})^2}$

$kas = \frac{(\cos(\text{fi} - \text{lameda} - \text{teta})^2)}{(\cos(\text{teta}) * \cos(\text{lameda})^2 * \cos(\text{Delta} + \text{lameda} + \text{teta}) * (1 + ((\sin(\text{fi} + \text{Delta}) * \sin(\text{fi} - \text{beta} - \text{teta})) / (\cos(\text{beta} - \text{lameda}) * \cos(\text{Delta} + \text{lameda} + \text{teta))))}^{0.5})^2}$

clique = 1

linha1 = 2

linha2 = 4

linha3 = 6

linha4 = 7

XG = 0

YG = 0

' inserir a 1ª camada

Wi = Range("'101!D" & linha1).Value \* Range("'101!E" & linha1).Value \* Range("C3").Value

WiT = Wi

Range("C11").Value = WiT

Wiy = Range("C11").Value \* Cos(Range("C5").Value)

Range("D11").Value = Wiy

Wix = Range("C11").Value \* Sin(Range("C5").Value)

Range("E11").Value = Wix

xx = (Range("'101!H" & linha2).Value + Range("'101!H" & linha3).Value) / 2

XG = xx \* Wi

yy = (Range("'101!I" & linha2).Value + Range("'101!I" & linha3).Value) / 2

YG = yy \* Wi

h = Range("'105!G4").Value - Range("'101!I" & linha4).Value

Range("F11").Value = h

Range("G11").Value = ka



```

Range("H11").Value = kas
deltalas = 0.5 * gama * h ^ 2 * ((1 + kv) * kas - ka)
Range("I11").Value = deltalas
la = 0.5 * gama * h ^ 2 * ka
Range("J11").Value = la
las = la + deltalas
Range("K11").Value = la
lax = las * Cos(lamedada + alpha + Delta)
Range("L11").Value = lax
lay = las * Sin(lamedada + alpha + Delta)
Range("M11").Value = lay
N = Wiy + lay
Range("N11").Value = N
T = -Wix + lax
Range("O11").Value = T
x = Range("'105!F4").Value + (Range("'101!H" & linha4).Value - Range("'105!F4").Value) * 2 / 3
Range("P11").Value = x
y = Range("'105!G4").Value - h * 2 / 3
Range("Q11").Value = x
mstb = Wix * (YG / WiT - Range("'101!I" & linha2).Value) + Wiy * (XG / WiT - Range("'101!H" & linha2).Value)
Range("R11").Value = mstb
mdst = lax * (y - Range("'101!I" & linha2).Value) - lay * (x - Range("'101!H" & linha2).Value)
Range("S11").Value = mdst
Mo = mstb - mdst
Range("T11").Value = Mo
eo = Mo / N
Range("U11").Value = eo
xb = Range("'101!H" & linha2).Value + eo * Cos(alpha)
Range("V11").Value = xb
Range("Y4").Value = xb
yb = Range("'101!I" & linha2).Value - eo * Sin(alpha)
Range("W11").Value = yb
Range("Z4").Value = yb

Nx = xb + N / 100 * Sin(alpha)
Ny = yb + N / 100 * Cos(alpha)
Tx = xb + T / 100 * Cos(alpha)
Ty = yb - T / 100 * Sin(alpha)
Rx = Tx + N / 100 * Sin(alpha)
Ry = Ny - T / 100 * Sin(alpha)
Range("Y5").Value = Rx
Range("Z5").Value = Ry

Range("'11!M4").Value = clique

```

Range("111!N4").Value =  $(N^2 + T^2)^{0.5}$

'inserir as restantes camadas

While clique < camada

Range("B11:W11").Select

Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

Range("B12").Select

Selection.AutoFill Destination:=Range("B11:B12"), Type:=xlFillDefault

Range("Y4:AC6").Select

Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

linha1 = linha1 + 1

linha2 = linha2 + 6

linha3 = linha3 + 6

linha4 = linha4 + 6

Wi = Range("101!D" & linha1).Value \* Range("101!E" & linha1).Value \* Range("C3").Value

WiT = WiT + Wi

Range("C11").Value = WiT

Wiy = WiT \* Cos(Range("C5").Value)

Range("D11").Value = Wiy

Wix = WiT \* Sin(Range("C5").Value)

Range("E11").Value = Wix

xx = (Range("101!H" & linha2).Value + Range("101!H" & linha3).Value) / 2

XG = XG + xx \* Wi

yy = (Range("101!I" & linha2).Value + Range("101!I" & linha3).Value) / 2

YG = YG + yy \* Wi

h = Range("105!G4").Value - Range("101!I" & linha4).Value

Range("F11").Value = h

Range("G11").Value = ka

Range("H11").Value = kas

deltalas =  $0.5 * \text{gama} * h^2 * ((1 + kv) * kas - ka)$

Range("I11").Value = deltalas

la =  $0.5 * \text{gama} * h^2 * ka$

Range("J11").Value = la

las = la + deltalas

Range("K11").Value = las

lax = las \* Cos(lamedada + alpha + Delta)

Range("L11").Value = lax

lay = las \* Sin(lamedada + alpha + Delta)

Range("M11").Value = lay

```
N = Wiy + lay
Range("N11").Value = N
T = -Wix + lax
Range("O11").Value = T
x = Range("'105!F4").Value + (Range("'101!H" & linha4).Value - Range("'105!F4").Value) * 2 / 3
Range("P11").Value = x
y = Range("'105!G4").Value - h * 2 / 3
Range("Q11").Value = y
mstb = Wix * (YG / WiT - Range("'101!H" & linha2).Value) + Wiy * (XG / WiT - Range("'101!H" & linha2).Value)
Range("R11").Value = mstb
mdst = lax * (y - Range("'101!H" & linha2).Value) - lay * (x - Range("'101!H" & linha2).Value)
Range("S11").Value = mdst
Mo = mstb - mdst
Range("T11").Value = Mo
eo = Mo / N
Range("U11").Value = eo
xb = Range("'101!H" & linha2).Value + eo * Cos(alpha)
Range("V11").Value = xb
Range("Y4").Value = xb
yb = Range("'101!H" & linha2).Value - eo * Sin(alpha)
Range("W11").Value = yb
Range("Z4").Value = yb

Nx = xb + N / 100 * Sin(alpha)
Ny = yb + N / 100 * Cos(alpha)
Tx = xb + T / 100 * Cos(alpha)
Ty = yb - T / 100 * Sin(alpha)
Rx = Tx + N / 100 * Sin(alpha)
Ry = Ny - T / 100 * Sin(alpha)
Range("Y5").Value = Rx
Range("Z5").Value = Ry

clique = clique + 1
Sheets("111").Select
Range("M4:N4").Select
Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
Range("M4").Value = clique
Range("N4").Value = (N ^ 2 + T ^ 2) ^ 0.5
Sheets("112").Select

Wend
End Sub

Sub resumo()
'=====
```

' 300 - ir para a folha 300

'=====

Sheets("300").Select

camada = Range("'100!C15").Value

While Range("'300!B14").Value <> 1

    Rows("14:14").Select

    Selection.Delete Shift:=xlUp

Wend

While Range("'300!B14").Value <> camada

    Rows("14:14").Select

    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

    Range("B15:E15").Select

    Selection.AutoFill Destination:=Range("B14:E15"), Type:=xlFillDefault

    Range("B14:E15").Select

Wend

Line = 110 + camada - 1

ExecuteExcel4Macro "PRINT(1,,1,,TRUE,,,,,2,,TRUE,,FALSE)"

End Sub

```

'=====
'
' Módulo para programação da estabilidade / superfícies de deslizamento '
'
'=====

Sub segcoord()
'
' 200 - definição das coordenadas dos segmentos
'

'=====
'linha superior
'=====

'informações genéricas
Sheets("200").Select
ncamada = Range("'109!C3").Value

' linha da fundação
linha = 9
nlinha = 1
Range("E9").Value = nlinha
Range("F9:G9").Value = Range("'103!F12:'103!G12").Value
If Range("'103!F12").Value <> Range("'103!F13").Value Then
    linha = linha + 1
    nlinha = nlinha + 1
    Range("E" & linha).Value = nlinha
    Range("F" & linha).Value = Range("'103!F13").Value
    Range("G" & linha).Value = Range("'103!G13").Value
End If

' contacto entre a linha da fundação e o muro
linha = linha + 1
nlinha = nlinha + 1
Range("E" & linha).Value = nlinha
Range("F" & linha).Value = Range("'103!F14").Value
Range("G" & linha).Value = Range("'103!G14").Value

' linha do muro
camada = ncamada
linhamur = 2 + 6 * camada
linhastop = linha
While camada > 0

```

```

    If Range("'"101!'" & linhamur).Value > Range("G" & linhastop).Value And Range("'"101!'" & linhamur).Value <>
Range("'"101!'" & linhamur - 3).Value And Range("'"101!H" & linhamur).Value <> Range("'"101!H" & linhamur - 3).Value Then
        linha = linha + 1
        nlinha = nlinha + 1
        Range("E" & linha).Value = nlinha
        Range("F" & linha).Value = Range("'"101!H" & linhamur).Value
        Range("G" & linha).Value = Range("'"101!'" & linhamur).Value
    End If
    linhamur = linhamur - 3
    camada = camada - 0.5
Wend
linha = linha + 1
nlinha = nlinha + 1
Range("E" & linha).Value = nlinha
Range("F" & linha).Value = Range("'"101!H6").Value
Range("G" & linha).Value = Range("'"101!I6").Value

'linha do terrapleno
linha = linha + 1
nlinha = nlinha + 1
Range("E" & linha).Value = nlinha
Range("F" & linha).Value = Range("'"105!F5").Value
Range("G" & linha).Value = Range("'"105!G5").Value
If Range("'"104!F12").Value = True Then
    linha = linha + 1
    nlinha = nlinha + 1
    Range("E" & linha).Value = nlinha
    Range("F" & linha).Value = Range("'"105!F6").Value
    Range("G" & linha).Value = Range("'"105!G6").Value
End If

'=====
'linha do muro
'=====

' linha da fundação
linha = 9
nlinha = 1
Range("H9").Value = nlinha
Range("I9:J9").Value = Range("'"103!F12:'103!G12").Value
If Range("'"103!F12").Value <> Range("'"103!F13").Value Then
    linha = linha + 1
    nlinha = nlinha + 1

```

```

Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'103!F13").Value
Range("J" & linha).Value = Range("'103!G13").Value
End If

' contacto entre a linha da fundação e o muro
linha = linha + 1
nlinha = nlinha + 1
Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'103!F14").Value
Range("J" & linha).Value = Range("'103!G14").Value

' linha do muro
linhamur = 2 + 6 * ncamada
linhastop = linha
linha = linha + 1
nlinha = nlinha + 1
While Range("'101!I" & linhamur).Value < Range("J" & linhastop).Value
    Range("H" & linha).Value = nlinha
    Range("I" & linha).Value = Range("'101!H" & linhamur).Value
    Range("J" & linha).Value = Range("'101!I" & linhamur).Value
    linhamur = linhamur - 3
Wend
linhamur = linhamur + 6
While linhamur <> 2 + 6 * ncamada
    If Range("'101!I" & linhamur).Value <> Range("'101!I" & linhamur - 3).Value And Range("'101!H" & linhamur).Value <>
Range("'101!H" & linhamur - 3).Value Then
        linha = linha + 1
        nlinha = nlinha + 1
        Range("H" & linha).Value = nlinha
        Range("I" & linha).Value = Range("'101!H" & linhamur).Value
        Range("J" & linha).Value = Range("'101!I" & linhamur).Value
    End If
    linhamur = linhamur + 3
Wend
linha = linha + 1
nlinha = nlinha + 1
Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'101!H" & 2 + 6 * ncamada).Value
Range("J" & linha).Value = Range("'101!I" & 2 + 6 * ncamada).Value
linha = linha + 1
nlinha = nlinha + 1
Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'101!H" & 1 + 6 * ncamada).Value

```

```

Range("J" & linha).Value = Range("'101!I" & 1 + 6 * ncamada).Value
linhastop = 6 * ncamada
While linhastop > 6
  linha = linha + 1
  nlinha = nlinha + 1
  Range("H" & linha).Value = nlinha
  Range("I" & linha).Value = Range("'101!H" & linhastop).Value
  Range("J" & linha).Value = Range("'101!I" & linhastop).Value
  linhastop = linhastop - 5
  If Range("'101!I" & linhastop).Value <> Range("'101!I" & linhastop + 5).Value And Range("'101!H" & linhastop).Value <>
Range("'101!H" & linhastop + 5).Value Then
    linha = linha + 1
    nlinha = nlinha + 1
    Range("H" & linha).Value = nlinha
    Range("I" & linha).Value = Range("'101!H" & linhastop).Value
    Range("J" & linha).Value = Range("'101!I" & linhastop).Value
  End If
  linhastop = linhastop - 1
Wend
linha = linha + 1
nlinha = nlinha + 1
Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'101!H6").Value
Range("J" & linha).Value = Range("'101!I6").Value

'linha do terraplano
linha = linha + 1
nlinha = nlinha + 1
Range("H" & linha).Value = nlinha
Range("I" & linha).Value = Range("'105!F5").Value
Range("J" & linha).Value = Range("'105!G5").Value
If Range("'104!F12").Value = True Then
  linha = linha + 1
  nlinha = nlinha + 1
  Range("H" & linha).Value = nlinha
  Range("I" & linha).Value = Range("'105!F6").Value
  Range("J" & linha).Value = Range("'105!G6").Value
End If

'=====
'camada do terraplano 1
'=====

```



```

' informações genéricas
ctum = Range("104!B29").Value
cota = Range("104!D29").Value

If ctum = True Then
  ' linha até à fundação do muro
  linha = 9
  nlinha = 1
  While Range("J" & linha).Value <> Range("101!I" & 1 + 6 * ncamada).Value
    Range("K" & linha).Value = nlinha
    Range("L" & linha).Value = Range("I" & linha).Value
    Range("M" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend
  Range("K" & linha).Value = nlinha
  Range("L" & linha).Value = Range("I" & linha).Value
  Range("M" & linha).Value = Range("J" & linha).Value
  linha = linha + 1
  nlinha = nlinha + 1

  'linha de contacto muro/terrapleno
  While Range("J" & linha).Value < cota
    Range("K" & linha).Value = nlinha
    Range("L" & linha).Value = Range("I" & linha).Value
    Range("M" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend

  'linha da camada
  Range("K" & linha).Value = nlinha
  Range("L" & linha).Value = Range("105!I4").Value
  Range("M" & linha).Value = cota
  linha = linha + 1
  nlinha = nlinha + 1
  Range("K" & linha).Value = nlinha
  Range("L" & linha).Value = Range("105!I5").Value
  Range("M" & linha).Value = cota
End If

'=====
'camada do terrapleno 2

```

```
'=====
' informações genéricas
ctdois = Range("'104!B30").Value
cota = Range("'104!D30").Value

If ctdois = True Then
  ' linha até à fundação do muro
  linha = 9
  nlinha = 1
  While Range("J" & linha).Value <> Range("'101!I" & 1 + 6 * ncamada).Value
    Range("N" & linha).Value = nlinha
    Range("O" & linha).Value = Range("I" & linha).Value
    Range("P" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend
  Range("N" & linha).Value = nlinha
  Range("O" & linha).Value = Range("I" & linha).Value
  Range("P" & linha).Value = Range("J" & linha).Value
  linha = linha + 1
  nlinha = nlinha + 1

  'linha de contacto muro/terrapleno
  While Range("J" & linha).Value < cota
    Range("N" & linha).Value = nlinha
    Range("O" & linha).Value = Range("I" & linha).Value
    Range("P" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend

  'linha da camada
  Range("N" & linha).Value = nlinha
  Range("O" & linha).Value = Range("'105!I7").Value
  Range("P" & linha).Value = cota
  linha = linha + 1
  nlinha = nlinha + 1
  Range("N" & linha).Value = nlinha
  Range("O" & linha).Value = Range("'105!I8").Value
  Range("P" & linha).Value = cota
End If
```

```

'=====
'camada do Nivel freático
'=====

' informações genéricas
NF = Range("106!B15").Value
cota = Range("106!G15").Value

If NF = True Then
  ' linha até à fundação do muro
  linha = 9
  nlinha = 1
  While Range("J" & linha).Value <> Range("101!I" & 1 + 6 * ncamada).Value
    Range("Q" & linha).Value = nlinha
    Range("R" & linha).Value = Range("I" & linha).Value
    Range("S" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend
  Range("Q" & linha).Value = nlinha
  Range("R" & linha).Value = Range("I" & linha).Value
  Range("S" & linha).Value = Range("J" & linha).Value
  linha = linha + 1
  nlinha = nlinha + 1

  'linha de contacto muro/terrapleno
  While Range("J" & linha).Value < cota
    Range("Q" & linha).Value = nlinha
    Range("R" & linha).Value = Range("I" & linha).Value
    Range("S" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
  Wend

  'linha da camada
  Range("Q" & linha).Value = nlinha
  Range("R" & linha).Value = Range("107!L16").Value
  Range("S" & linha).Value = cota
  linha = linha + 1
  nlinha = nlinha + 1
  Range("Q" & linha).Value = nlinha
  Range("R" & linha).Value = Range("107!L17").Value
  Range("S" & linha).Value = cota
End If

```

```

'=====
'limite da fundação
'=====

' linha até à fundação do muro
linha = 9
nlinha = 1
While Range("J" & linha).Value <> Range("'"101"!I" & 1 + 6 * ncamada).Value
    Range("T" & linha).Value = nlinha
    Range("U" & linha).Value = Range("I" & linha).Value
    Range("V" & linha).Value = Range("J" & linha).Value
    linha = linha + 1
    nlinha = nlinha + 1
Wend
Range("T" & linha).Value = nlinha
Range("U" & linha).Value = Range("I" & linha).Value
Range("V" & linha).Value = Range("J" & linha).Value
linha = linha + 1
nlinha = nlinha + 1
Range("T" & linha).Value = nlinha
Range("U" & linha).Value = Range("'"103!F9").Value
Range("V" & linha).Value = Range("'"103!G9").Value

'=====
'restantes camadas da fundação
'=====

ncamada = Range("'"102!C38").Value

'caso de 5 camadas
If ncamada = 5 Then
    'linha 5
    zcinco = Range("'"102!D38").Value
    Range("AJ9").Value = Range("'"103!I4").Value
    Range("AJ10").Value = Range("'"103!I3").Value
    Range("AK9").Value = zcinco
    Range("AK10").Value = zcinco
    'linha 4
    zquatro = Range("'"102!D39").Value
    Range("AG9").Value = Range("'"103!I4").Value
    Range("AG10").Value = Range("'"103!I3").Value

```

```
Range("AH9").Value = zquatro
Range("AH10").Value = zquatro
'linha 3
ztres = Range("'102!D40").Value
Range("AD9").Value = Range("'103!I4").Value
Range("AD10").Value = Range("'103!I3").Value
Range("AE9").Value = ztres
Range("AE10").Value = ztres
'linha 2
zdois = Range("'102!D41").Value
Range("AA9").Value = Range("'103!I4").Value
Range("AA10").Value = Range("'103!I3").Value
Range("AB9").Value = zdois
Range("AB10").Value = zdois
'linha 1
zum = Range("'102!D42").Value
Range("X9").Value = Range("'103!I4").Value
Range("X10").Value = Range("'103!I3").Value
Range("Y9").Value = zum
Range("Y10").Value = zum
End If

'caso de 4 camadas
If ncamada = 4 Then
'linha 4
zquatro = Range("'102!D38").Value
Range("AG9").Value = Range("'103!I4").Value
Range("AG10").Value = Range("'103!I3").Value
Range("AH9").Value = zquatro
Range("AH10").Value = zquatro
'linha 3
ztres = Range("'102!D39").Value
Range("AD9").Value = Range("'103!I4").Value
Range("AD10").Value = Range("'103!I3").Value
Range("AE9").Value = ztres
Range("AE10").Value = ztres
'linha 2
zdois = Range("'102!D40").Value
Range("AA9").Value = Range("'103!I4").Value
Range("AA10").Value = Range("'103!I3").Value
Range("AB9").Value = zdois
Range("AB10").Value = zdois
'linha 1
zum = Range("'102!D41").Value
```

```
Range("X9").Value = Range("103!I4").Value
Range("X10").Value = Range("103!I3").Value
Range("Y9").Value = zum
Range("Y10").Value = zum
End If
```

'caso de 3 camadas

```
If ncamada = 3 Then
    'linha 3
    ztres = Range("102!D38").Value
    Range("AD9").Value = Range("103!I4").Value
    Range("AD10").Value = Range("103!I3").Value
    Range("AE9").Value = ztres
    Range("AE10").Value = ztres
    'linha 2
    zdois = Range("102!D39").Value
    Range("AA9").Value = Range("103!I4").Value
    Range("AA10").Value = Range("103!I3").Value
    Range("AB9").Value = zdois
    Range("AB10").Value = zdois
    'linha 1
    zum = Range("102!D40").Value
    Range("X9").Value = Range("103!I4").Value
    Range("X10").Value = Range("103!I3").Value
    Range("Y9").Value = zum
    Range("Y10").Value = zum
End If
```

'caso de 2 camadas

```
If ncamada = 2 Then
    'linha 2
    zdois = Range("102!D38").Value
    Range("AA9").Value = Range("103!I4").Value
    Range("AA10").Value = Range("103!I3").Value
    Range("AB9").Value = zdois
    Range("AB10").Value = zdois
    'linha 1
    zum = Range("102!D39").Value
    Range("X9").Value = Range("103!I4").Value
    Range("X10").Value = Range("103!I3").Value
    Range("Y9").Value = zum
    Range("Y10").Value = zum
End If
```

```
'caso de 1 camada
If ncamada = 1 And Range("'102!D38").Value <> 0 Then
    'linha 1
    zum = Range("'102!D38").Value
    Range("X9").Value = Range("'103!I4").Value
    Range("X10").Value = Range("'103!I3").Value
    Range("Y9").Value = zum
    Range("Y10").Value = zum
End If

End Sub
Sub segint()
,
' 200 - determinação de todas as intersecções das linhas de separação com os círculos de deslizamento
,

'=====
'verificação de todas as curvas possíveis começando por: xmin, ymin e rmin
'=====

iteração = 1
linha = 6 'linha de registo de dados na folha 201
nsegmentos = Range("'200!D12").Value

'While iteração <> 2 ' alterar para factor de segurança

'-----
'introdução dos dados das superfícies de escorregamento
'-----

Sheets("200").Select
If iteração = 1 Then
    'determinação da coordenada do centro
    SolverReset
    SolverAdd CellRef:="$B$63", Relation:=2, FormulaText:="0"
    SolverOk SetCell:="$B$62", MaxMinVal:=3, ValueOf:="0", ByChange:="$C$60"
    SolverSolve True
    'determinação de L2
    SolverReset
    SolverOk SetCell:="$B$65", MaxMinVal:=3, ValueOf:="0", ByChange:="$D$58"
    SolverSolve True
    Range("D26").Value = "(0-'103!B24)/2"
```

```
L11 = Range("D26").Value
Range("D27").Value = "="&101&"!I37-D58"
L21 = Range("D27").Value
Range("D28").Value = "="&(MAX("105"!F5:"105"!F6)-"105"!F4)/2"
L31 = Range("D28").Value
Range("C51").Value = L11
Range("C52").Value = L21
Range("C53").Value = L31
'determinação do raio e coordenadas do círculo
Range("B74").Value = "99"
SolverReset
SolverAdd CellRef:="$C$76", Relation:=2, FormulaText:="0"
SolverAdd CellRef:="$D$76", Relation:=2, FormulaText:="0"
SolverOk SetCell:="$B$76", MaxMinVal:=3, ValueOf:="0", ByChange:="$B$74:$D$74"
SolverSolve True
R1 = Range("B74").Value
Xc1 = Range("C74").Value
yc1 = Range("D74").Value
End If
```

```
If iteração > 1 Then
  L11 = Range("D26").Value
  L21 = Range("D27").Value
  L31 = Range("D28").Value
  Range("B74").Value = "99"
  SolverReset
  SolverAdd CellRef:="$C$76", Relation:=2, FormulaText:="0"
  SolverAdd CellRef:="$D$76", Relation:=2, FormulaText:="0"
  SolverOk SetCell:="$B$76", MaxMinVal:=3, ValueOf:="0", ByChange:="$B$74:$D$74"
  SolverSolve True
  R1 = Range("B74").Value
  Xc1 = Range("C74").Value
  yc1 = Range("D74").Value
End If
```

'Determinação dos raios e centros das restantes 3 curvas

```
L12 = Range("D32").Value
Range("C51").Value = L12
L22 = Range("D33").Value
Range("C52").Value = L22
L32 = Range("D34").Value
Range("C53").Value = L32
Range("B74").Value = "99"
SolverReset
```



```

SolverAdd CellRef:="$C$76", Relation:="<=", FormulaText:="0"
SolverAdd CellRef:="$D$76", Relation:="<=", FormulaText:="0"
SolverOk SetCell:="$B$76", MaxMinVal:="3", ValueOf:="0", ByChange:="$B$74:$D$74"
SolverSolve True
R2 = Range("B74").Value
xc2 = Range("C74").Value
yc2 = Range("D74").Value

L13 = Range("D38").Value
Range("C51").Value = L13
L23 = Range("D39").Value
Range("C52").Value = L23
L33 = Range("D40").Value
Range("C53").Value = L33
Range("B74").Value = "99"
SolverReset
SolverAdd CellRef:="$C$76", Relation:="<=", FormulaText:="0"
SolverAdd CellRef:="$D$76", Relation:="<=", FormulaText:="0"
SolverOk SetCell:="$B$76", MaxMinVal:="3", ValueOf:="0", ByChange:="$B$74:$D$74"
SolverSolve True
R3 = Range("B74").Value
xc3 = Range("C74").Value
yc3 = Range("D74").Value

L14 = Range("D44").Value
Range("C51").Value = L14
L24 = Range("D45").Value
Range("C52").Value = L24
L34 = Range("D46").Value
Range("C53").Value = L34
Range("B74").Value = "99"
SolverReset
SolverAdd CellRef:="$C$76", Relation:="<=", FormulaText:="0"
SolverAdd CellRef:="$D$76", Relation:="<=", FormulaText:="0"
SolverOk SetCell:="$B$76", MaxMinVal:="3", ValueOf:="0", ByChange:="$B$74:$D$74"
SolverSolve True
R4 = Range("B74").Value
xc4 = Range("C74").Value
yc4 = Range("D74").Value

```

```

'-----
'início das verificações em todos as circunferências
'-----

```

```

circ = 1
While circ < 5
  If circ = 1 Then
    Xc = Xc1
    Yc = yc1
    R = R1
    linhas = 7
  Elself circ = 2 Then
    Xc = xc2
    Yc = yc2
    R = R2
    linhas = 8
  Elself circ = 3 Then
    Xc = xc3
    Yc = yc3
    R = R3
    linhas = 9
  Elself circ = 4 Then
    Xc = xc4
    Yc = yc4
    R = R4
    linhas = 10

End If
Range("201!F" & linhas).Value = R
Range("201!E" & linhas).Value = Yc
Range("201!D" & linhas).Value = Xc
'diferença entre as abcissas e as ordenadas dos pontos que definem o segmento
sup = 1 'número de superfície e respectivas colunas de dados na folha 200
raizcount = 0
colunax = 6
colunay = 7
colunaxx = 6
While sup <= 11
  linhac = 9 'linha das coordeandas
  linhacstop = linhac + Worksheets("200").Cells(6, colunay).Value

  If Worksheets("200").Cells(6, colunay).Value <> 0 Then 'se a linha não tiver dados, salta para a próxima
    Nsol = 0 'nº de soluções = 0
    While linhac <= linhacstop 'entrada das coordenadas do segmento
      x1 = Worksheets("200").Cells(linhac, colunax).Value
      y1 = Worksheets("200").Cells(linhac, colunay).Value
      x2 = Worksheets("200").Cells(linhac + 1, colunax).Value
      y2 = Worksheets("200").Cells(linhac + 1, colunay).Value
    
```

```

segmento      deltx = x2 - x1          'diferença entre as abscissas e as ordenadas dos pontos que definem o
              delty = y2 - y1

              If deltx <> 0 Then
                  Slope = delty / deltx          'cálculo do declive
                  B = (y1 * x2 - y2 * x1) / deltx  'cálculo da ordenada na origem y=m.x+b ; b - ordenada na origem
com a circunferência A.x2+B.x+C=0          'cálculo do parâmetro A da equação do 2º grau que define a intersecção da recta
              ' soluções da equação:  x = (-B+/- sqrt(B2-4.A.C))/2.A
              '          x = -B/2.A +- sqrt((B2-4.A.C)/4.A2)
              '          x = -B/2.A +- sqrt((B/2A)^2-C)/A

              a = 1 + Slope * Slope          ' Cálculo do termo B/2A
              B2A = (Slope * (B - Yc) - Xc) / a
              CA = ((B - Yc) ^ 2 + Xc ^ 2 - R ^ 2) / a  'cálculo do quociente C/A
              B24AC = B2A * B2A - CA          'cálculo de (B2-4AC)/4A2

              'determinação das soluções da equação
              If B24AC > 0 Then          'verificar a existência de soluções

                  Raiz = B24AC ^ 0.5
                  Raiz1 = -B2A - Raiz
                  Raiz2 = -B2A + Raiz

                  If Raiz1 >= x1 And Raiz1 <= x2 Then  'verificar raiz1
                      Nsol = Nsol + 1
                      Worksheets("201").Cells(linhas, colunaxx + 2 * sup).Value = Raiz1

                  End If

                  If Raiz2 >= x1 And Raiz2 <= x2 Then  'verificar raiz2
                      If Raiz1 <> Raiz2 Then          'verificar se a raiz não é a mesma
                          Nsol = Nsol + 1
                          If Nsol = 1 Then
                              Worksheets("201").Cells(linhas, colunaxx + 2 * sup).Value = Raiz2
                          Elseif Nsol = 2 Then
                              Worksheets("201").Cells(linhas, colunaxx + 2 * sup + 1).Value = Raiz2
                          End If
                      End If
                  End If

                  End If

              End If

              Elseif x1 <= (Xc + R) And x1 >= (Xc - R) Then  'troço vertical
                  Raiz1 = x2
                  Nsol = 1

```

```
End If

    linhac = linhac + 1
Wend
End If
sup = sup + 1 'mudar de superfície
raizcount = 0
colunax = colunax + 3
colunay = colunay + 3

Wend

'call reorg

    circ = circ + 1
Wend
iteração = iteração + 1
'Wend
End Sub
Sub reorg()
'
' 200 - reordenar os pontos de intersecção
'
'=====
'verificar se as curvas estão completamente dentro da área definida
'=====

'copiar a lista para uma coluna
Sheets("201").Select
Range("H7:AC10").Select
Application.CutCopyMode = False
Selection.Copy
Sheets("202").Select
Range("B3").Select
Selection.PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:= _
    False, Transpose:=True
ActiveWindow.SmallScroll Down:=9

'ordenar por ordem crescente
Range("B2:E2").Select
Selection.AutoFilter
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Clear
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Add Key:=Range( _
```

```
"B2"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
xlSortNormal
With ActiveWorkbook.Worksheets("202").AutoFilter.Sort
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Clear
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Add Key:=Range( _
"C2"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
xlSortNormal
With ActiveWorkbook.Worksheets("202").AutoFilter.Sort
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Clear
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Add Key:=Range( _
"D2"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
xlSortNormal
With ActiveWorkbook.Worksheets("202").AutoFilter.Sort
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Clear
ActiveWorkbook.Worksheets("202").AutoFilter.Sort.SortFields.Add Key:=Range( _
"E2"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
xlSortNormal
With ActiveWorkbook.Worksheets("202").AutoFilter.Sort
.Header = xlYes
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
Selection.AutoFilter
End Sub
```

Sub intern()

```
'=====
' 111 - calcular os impulsos para a folha 111
'=====
```

```
Sheets("112").Select
camada = Range("C3").Value
```

```
While Range("B16").Value <> camada
```

```
    Range("B16:Z16").Select
    Selection.Delete Shift:=xlUp
```

```
    Range("Y4:Z6").Select
    Selection.Delete Shift:=xlUp
```

```
Wend
```

```
Sheets("111").Select
```

```
While Range("Q4").Value <> 1
```

```
    Range("M4:Q4").Select
    Selection.Delete Shift:=xlUp
```

```
Wend
```

```
Sheets("112").Select
```

```
alpha = Range("C5").Value
lameda = Range("C6").Value
teta = Range("C8").Value
kv = Range("C9").Value
kh = Range("C10").Value
ncamada = camada
peso = Range("C4").Value
zmax = Range("O6").Value
xmax = Range("N6").Value
zfund = Range("O10").Value
zzzNF = Range("P9").Value
znf = Range("O9").Value
zzzprim = Range("P7").Value
zprim = Range("O7").Value
zzzseg = Range("P8").Value
zseg = Range("O8").Value
zpass = Range("O11").Value
zzNFF = Range("P12").Value
zNFF = Range("O12").Value
```

```

linha1 = 2
linha2 = 4
linha3 = 6
linha4 = 7
linha5 = 8
XG = 0
YG = 0
WiT = 0

While ncamada > 0
  If ncamada <> camada Then
    Range("B16:Z16").Select
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

    Range("Y4:Z6").Select
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

  End If
  Range("B16").Value = ncamada
  T = 0
  N = 0
  '=====
  ' cálculo da componente estabilizadora
  '=====

  Wi = Range("'"101!D" & linha1).Value * Range("'"101!E" & linha1).Value * peso
  WiT = WiT + Wi
  Range("C16").Value = WiT
  xx = (Range("'"101!H" & linha2).Value + Range("'"101!H" & linha3).Value) / 2
  XG = XG + xx * Wi
  Range("D16").Value = XG / WiT
  yy = (Range("'"101!I" & linha2).Value + Range("'"101!I" & linha3).Value) / 2
  YG = YG + yy * Wi
  Range("E16").Value = YG / WiT
  Wiy = Wi * Cos(alpha)
  WiyT = WiyT + Wiy
  Range("F16").Value = WiyT
  xxWiy = XG / WiT * WiyT
  Range("G16").Value = xxWiy
  Wix = Wi * Sin(alpha)
  WixT = WixT + Wix
  Range("H16").Value = WixT

```

```

yyWix = YG / WiT * WixT
Range("I16").Value = yyWix
WiKhx = Wix * kh
WiKhxT = WiKhxT + WiKhx
Range("J16").Value = WiKhxT
yyWiKhx = YG / WiT * WiKhxT
Range("K16").Value = yyWiKhx
WiKhy = Wiy * kh
WiKhyT = WiKhyT + WiKhy
Range("L16").Value = WiKhyT
xxWiKhy = XG / WiT * WiKhyT
Range("M16").Value = xxWiKhy
WiKvx = Wix * kv
WiKvxT = WiKvxT + WiKvx
Range("N16").Value = WiKvxT
yyWiKvx = XG / WiT * WiKvxT
Range("O16").Value = yyWiKvx
WiKvy = Wiy * kv
WiKvyT = WiKvyT + WiKvy
Range("P16").Value = WiKvy
xxWiKvy = XG / WiT * WiKvyT
Range("Q16").Value = xxWiKvy

mstb = xxWiy + yyWix - yyWiKhx - xxWiKhy + yyWiKvx + xxWiKvy
Range("T16").Value = mstb
T = -WixT - WiKhxT - WiKvxT
N = WiyT + WiKhyT + WiKvyT
escest = WiyT + WiKhyT + WiKvyT + Tan(Range("C11").Value) * (WiyT + WiKhyT + WiKvyT)

'=====
' cálculo da componente instabilizadora
'=====

zinf = Range("'101!!" & linha4).Value

If zzzNF = True And znf > zinf Then
    zznf = True
Else
    zznf = False
End If

If zzzprim = True And zprim > zinf Then
    zzprim = True

```



```

Else
  zzprim = False
End If

If zzzseg = True And zseg > zinf Then
  zzseg = True
Else
  zzseg = False
End If

escinst = 0
sq = Range("O3").Value
mdst = 0

If zzNF = True And znf > zinf Then

  If zzprim = True And zprim > zinf Then

    If zzseg = True And zseg > zinf Then

      If znf < zseg Then 'cam1 - cam2 - cam3 - NF
        '1ª camada
        h1 = zmax - Range("O7").Value
        ka1 = Range("F6").Value
        kas = Range("F7").Value
        gama1 = Range("F3").Value
        Delta1 = Range("F5").Value
        la = 0.5 * gama1 * h1 ^ 2 * ka1
        deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
        lax = la * Cos(lamedada + alpha + Delta1)
        y = zmax - 2 / 3 * h1 - Range("'101!!" & linha5).Value
        lay = la * Sin(lamedada + alpha + Delta1)
        x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("'101!H" & linha5).Value
        mdst = mdst + lax * y - lay * x
        lasx = deltalas * Cos(lamedada + alpha + Delta1)
        y = zmax - 0.4 * h1 - Range("'101!!" & linha5).Value
        lasy = deltalas * Sin(lamedada + alpha + Delta1)
        x = xmax + Tan(lamedada) * 0.4 * h1 - Range("'101!H" & linha5).Value
        mdst = mdst + lasx * y - lasy * x
        T = T + lax + lasx
        N = N + lay + lasy
        escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

      '2ª camada
    
```

---

```

h2 = Range("O7").Value - Range("O8").Value
ka2 = Range("I6").Value
kas = Range("I7").Value
gama2 = Range("I3").Value
Delta2 = Range("I5").Value
la = 0.5 * gama2 * h2 ^ 2 * ka2
deltalas = 0.5 * gama2 * h2 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 2 / 3 * h2 - Range("'101!'I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 2 / 3 * h2) - Range("'101!'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 0.4 * h2 - Range("'101!'I" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 0.4 * h2) - Range("'101!'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'3ª camada
h3 = Range("O8").Value - Range("O9").Value
ka3 = Range("L6").Value
kas = Range("L7").Value
gama3 = Range("L3").Value
Delta3 = Range("L5").Value
la = 0.5 * gama3 * h3 ^ 2 * ka3
deltalas = 0.5 * gama3 * h3 ^ 2 * ((1 + kv) * kas - ka3)
lax = la * Cos(lamedada + alpha + Delta3)
y = zmax - h1 - h2 - 2 / 3 * h3 - Range("'101!'I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta3)
x = xmax + Tan(lamedada) * (h1 + h2 + 2 / 3 * h3) - Range("'101!'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta3)
y = zmax - h1 - h2 - 0.4 * h3 - Range("'101!'I" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta3)
x = xmax + Tan(lamedada) * (h1 + h2 + 0.4 * h3) - Range("'101!'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'NF

```

```

h4 = Range("O9").Value - Range("'101!I" & linha4).Value
gama4 = Range("L3").Value - 9.81
la = 0.5 * gama4 * h4 ^ 2 * ka3
deltas = 0.5 * gama4 * h4 ^ 2 * ((1 + kv) * kas - ka3)
lax = la * Cos(lamedada + alpha + Delta3)
y = zmax - h1 - h2 - h3 - 2 / 3 * h4 - Range("'101!I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta3)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 + 2 / 3 * h4) - Range("'101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltas * Cos(lamedada + alpha + Delta3)
y = zmax - h1 - h2 - h3 - 0.4 * h4 - Range("'101!I" & linha5).Value
lasy = deltas * Sin(lamedada + alpha + Delta3)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 + 0.4 * h4) - Range("'101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("'101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 / 2) - Range("'101!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

lq2 = (sq + gama1 * h1) * h2 * ka2
lq2x = lq2 * Cos(Delta2 + lamedada + alpha)
lq2y = lq2 * Sin(Delta2 + lamedada + alpha)
y = zmax - h1 - h2 / 2 - Range("'101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 / 2) - Range("'101!H" & linha5).Value
mdst = mdst + lq2x * y - lq2y * x

lq3 = (sq + gama1 * h1 + gama2 * h2) * h3 * ka3
lq3x = lq3 * Cos(Delta3 + lamedada + alpha)
lq3y = lq3 * Sin(Delta3 + lamedada + alpha)
y = zmax - h1 - h2 - h3 / 2 - Range("'101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 + h3 / 2) - Range("'101!H" & linha5).Value
mdst = mdst + lq3x * y - lq3y * x

lq4 = (sq + gama1 * h1 + gama2 * h2 + gama3 * h3) * h4 * ka3
lq4x = lq4 * Cos(Delta3 + lamedada + alpha)
lq4y = lq4 * Sin(Delta3 + lamedada + alpha)
y = zmax - h1 - h2 - h3 - h4 / 2 - Range("'101!I" & linha5).Value

```

```
x = xmax + Tan(lamedada) * (h1 + h2 + h3 + h4 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq4x * y - lq4y * x
```

```
'água
```

```
lw = 0.5 * 9.81 * h4 ^ 2
```

```
lwx = lw * Cos(lamedada + alpha)
```

```
lwy = lw * Sin(lamedada + alpha)
```

```
y = zmax - h1 - h2 - h3 - 2 / 3 * h4 - Range("101!I" & linha5).Value
```

```
x = xmax + Tan(lamedada) * (h1 + h2 + h3 + 2 / 3 * h4) - Range("101!H" & linha5).Value
```

```
mdst = mdst + lwx * y - lwy * x
```

```
T = T + lq1x + lq2x + lq3x + lq4x + lwx
```

```
N = N + lq1y + lq2y + lq3y + lq4y + lwy
```

```
escinst = escinst + (lq1x + lq2x + lq3x + lq4x + lwx) - (lq1y + lq2y + lq3y + lq4y + lwy) *  
Tan(Range("C11").Value)
```

```
End If
```

```
If znf < zprim And znf > zseg Then 'cam1 - cam2 - NF - cam3
```

```
'1ª camada
```

```
h1 = zmax - Range("O7").Value
```

```
ka1 = Range("F6").Value
```

```
kas = Range("F7").Value
```

```
gama1 = Range("F3").Value
```

```
Delta1 = Range("F5").Value
```

```
la = 0.5 * gama1 * h1 ^ 2 * ka1
```

```
deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
```

```
lax = la * Cos(lamedada + alpha + Delta1)
```

```
y = zmax - 2 / 3 * h1 - Range("101!I" & linha5).Value
```

```
lay = la * Sin(lamedada + alpha + Delta1)
```

```
x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("101!H" & linha5).Value
```

```
mdst = mdst + lax * y - lay * x
```

```
lasx = deltalas * Cos(lamedada + alpha + Delta1)
```

```
y = zmax - 0.4 * h1 - Range("101!I" & linha5).Value
```

```
lasy = deltalas * Sin(lamedada + alpha + Delta1)
```

```
x = xmax + Tan(lamedada) * 0.4 * h1 - Range("101!H" & linha5).Value
```

```
mdst = mdst + lasx * y - lasy * x
```

```
T = T + lax + lasx
```

```
N = N + lay + lasy
```

```
escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)
```

```
'2ª camada
```

```
h2 = Range("O7").Value - Range("O9").Value
```

```

ka2 = Range("I6").Value
kas = Range("I7").Value
gama2 = Range("I3").Value
Delta2 = Range("I5").Value
la = 0.5 * gama2 * h2 ^ 2 * ka2
deltalas = 0.5 * gama2 * h2 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 2 / 3 * h2 - Range("'101'!" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 2 / 3 * h2) - Range("'101'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 0.4 * h2 - Range("'101'!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 0.4 * h2) - Range("'101'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'NF
h3 = Range("O9").Value - Range("O8").Value
gama3 = Range("I3").Value - 9.81
la = 0.5 * gama3 * h3 ^ 2 * ka2
deltalas = 0.5 * gama3 * h3 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - h2 - h3 * 2 / 3 - Range("'101'!" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 * 2 / 3) - Range("'101'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - h2 - h3 * 0.4 - Range("'101'!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 * 0.4) - Range("'101'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'3ª camada
h4 = Range("O8").Value - Range("'101'!" & linha4).Value
ka4 = Range("L6").Value
kas = Range("L7").Value
gama4 = Range("L3").Value - 9.81

```

Delta4 = Range("L5").Value  
la = 0.5 \* gama4 \* h4 ^ 2 \* ka4  
deltalas = 0.5 \* gama4 \* h4 ^ 2 \* ((1 + kv) \* kas - ka4)  
lax = la \* Cos(lamedada + alpha + Delta4)  
y = zmax - h1 - h2 - h3 - h4 \* 2 / 3 - Range("'101'!" & linha5).Value  
lay = la \* Sin(lamedada + alpha + Delta4)  
x = xmax + Tan(lamedada) \* (h1 + h2 + h3 + h4 \* 2 / 3) - Range("'101'H" & linha5).Value  
mdst = mdst + lax \* y - lay \* x  
lasx = deltalas \* Cos(lamedada + alpha + Delta4)  
y = zmax - h1 - h2 - h3 - h4 \* 0.4 - Range("'101'!" & linha5).Value  
lasy = deltalas \* Sin(lamedada + alpha + Delta4)  
x = xmax + Tan(lamedada) \* (h1 + h2 + h3 + h4 \* 0.4) - Range("'101'H" & linha5).Value  
mdst = mdst + lasx \* y - lasy \* x  
T = T + lax + lasx  
N = N + lay + lasy  
escinst = escinst + (lax + lasx) - (lay + lasy) \* Tan(Range("C11").Value)

'sobrecarga  
lq1 = sq \* h1 \* ka1  
lq1x = lq1 \* Cos(Delta1 + lamedada + alpha)  
lq1y = lq1 \* Sin(Delta1 + lamedada + alpha)  
y = zmax - h1 / 2 - Range("'101'!" & linha5).Value  
x = xmax + Tan(lamedada) \* (h1 / 2) - Range("'101'H" & linha5).Value  
mdst = mdst + lq1x \* y - lq1y \* x

lq2 = (sq + gama1 \* h1) \* h2 \* ka2  
lq2x = lq2 \* Cos(Delta2 + lamedada + alpha)  
lq2y = lq2 \* Sin(Delta2 + lamedada + alpha)  
y = zmax - h1 - h2 / 2 - Range("'101'!" & linha5).Value  
x = xmax + Tan(lamedada) \* (h1 + h2 / 2) - Range("'101'H" & linha5).Value  
mdst = mdst + lq2x \* y - lq2y \* x

lq3 = (sq + gama1 \* h1 + gama2 \* h2) \* h3 \* ka2  
lq3x = lq3 \* Cos(Delta2 + lamedada + alpha)  
lq3y = lq3 \* Sin(Delta2 + lamedada + alpha)  
y = zmax - h1 - h2 - h3 / 2 - Range("'101'!" & linha5).Value  
x = xmax + Tan(lamedada) \* (h1 + h2 + h3 / 2) - Range("'101'H" & linha5).Value  
mdst = mdst + lq3x \* y - lq3y \* x

lq4 = (sq + gama1 \* h1 + gama2 \* h2 + gama3 \* h3) \* h4 \* ka4  
lq4x = lq4 \* Cos(Delta4 + lamedada + alpha)  
lq4y = lq4 \* Sin(Delta4 + lamedada + alpha)  
y = zmax - h1 - h2 - h3 - h4 / 2 - Range("'101'!" & linha5).Value  
x = xmax + Tan(lamedada) \* (h1 + h2 + h3 + h4 / 2) - Range("'101'H" & linha5).Value

```

mdst = mdst + lq4x * y - lq4y * x

'água
lw = 0.5 * 9.81 * (znf - Range("'101'!" & linha4).Value) ^ 2
lwx = lw * Cos(lamedada + alpha)
lwy = lw * Sin(lamedada + alpha)
y = zmax - h1 - h2 - (h3 + h4) * 2 / 3 - Range("'101'!" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 + (h3 + h4) * 2 / 3) - Range("'101'H" & linha5).Value
mdst = mdst + lwx * y - lwy * x
T = T + lq1x + lq2x + lq3x + lq4x + lwx
N = N + lq1y + lq2y + lq3y + lq4y + lwy
escinst = escinst + (lq1x + lq2x + lq3x + lq4x + lwx) - (lq1y + lq2y + lq3y + lq4y + lwy) *
Tan(Range("C11").Value)

End If

If znf > zprim Then 'cam1 - NF - cam2 - cam3

'1ª camada
h1 = Range("O6").Value - Range("O9").Value
ka1 = Range("F6").Value
kas = Range("F7").Value
gama1 = Range("F3").Value
Delta1 = Range("F5").Value
la = 0.5 * gama1 * h1 ^ 2 * ka1
deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - 2 / 3 * h1 - Range("'101'!" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("'101'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta1)
y = zmax - 0.4 * h1 - Range("'101'!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 0.4 * h1 - Range("'101'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'NF
h2 = Range("O9").Value - Range("O7").Value
gama2 = Range("F3").Value - 9.81
la = 0.5 * gama2 * h2 ^ 2 * ka1

```

$$\begin{aligned} \text{deltalas} &= 0.5 * \text{gama2} * h2^2 * ((1 + kv) * \text{kas} - \text{ka1}) \\ \text{lax} &= \text{la} * \text{Cos}(\text{lameda} + \alpha + \text{Delta1}) \\ y &= z_{\text{max}} - h1 - h2 * 2 / 3 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lay} &= \text{la} * \text{Sin}(\text{lameda} + \alpha + \text{Delta1}) \\ x &= x_{\text{max}} + \text{Tan}(\text{lameda}) * (h1 + h2 * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lay} * x \\ \text{lasx} &= \text{deltalas} * \text{Cos}(\text{lameda} + \alpha + \text{Delta1}) \\ y &= z_{\text{max}} - h1 - h2 * 0.4 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lasy} &= \text{deltalas} * \text{Sin}(\text{lameda} + \alpha + \text{Delta1}) \\ x &= x_{\text{max}} + \text{Tan}(\text{lameda}) * (h1 + h2 * 0.4) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lasx} * y - \text{lasy} * x \\ T &= T + \text{lax} + \text{lasx} \\ N &= N + \text{lay} + \text{lasy} \\ \text{escinst} &= \text{escinst} + (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value}) \end{aligned}$$

'2ª camada

$$\begin{aligned} h3 &= \text{Range}("O7").\text{Value} - \text{Range}("O8").\text{Value} \\ \text{ka3} &= \text{Range}("I6").\text{Value} \\ \text{kas} &= \text{Range}("I7").\text{Value} \\ \text{gama3} &= \text{Range}("I3").\text{Value} - 9.81 \\ \text{Delta3} &= \text{Range}("I5").\text{Value} \\ \text{la} &= 0.5 * \text{gama3} * h3^2 * \text{ka3} \\ \text{deltalas} &= 0.5 * \text{gama3} * h3^2 * ((1 + kv) * \text{kas} - \text{ka3}) \\ \text{lax} &= \text{la} * \text{Cos}(\text{lameda} + \alpha + \text{Delta3}) \\ y &= z_{\text{max}} - h1 - h2 - h3 * 2 / 3 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lay} &= \text{la} * \text{Sin}(\text{lameda} + \alpha + \text{Delta3}) \\ x &= x_{\text{max}} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lay} * x \\ \text{lasx} &= \text{deltalas} * \text{Cos}(\text{lameda} + \alpha + \text{Delta3}) \\ y &= z_{\text{max}} - h1 - h2 - h3 * 0.4 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lasy} &= \text{deltalas} * \text{Sin}(\text{lameda} + \alpha + \text{Delta3}) \\ x &= x_{\text{max}} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 * 0.4) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lasx} * y - \text{lasy} * x \\ T &= T + \text{lax} + \text{lasx} \\ N &= N + \text{lay} + \text{lasy} \\ \text{escinst} &= \text{escinst} + (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value}) \end{aligned}$$

'3ª camada

$$\begin{aligned} h4 &= \text{Range}("O8").\text{Value} - \text{Range}("101!!" \& \text{linha4}).\text{Value} \\ \text{ka4} &= \text{Range}("L6").\text{Value} \\ \text{kas} &= \text{Range}("L7").\text{Value} \\ \text{gama4} &= \text{Range}("L3").\text{Value} - 9.81 \\ \text{Delta4} &= \text{Range}("L5").\text{Value} \\ \text{la} &= 0.5 * \text{gama4} * h4^2 * \text{ka4} \end{aligned}$$



$$\begin{aligned} \text{deltalas} &= 0.5 * \text{gama4} * h4^2 * ((1 + kv) * \text{kas} - \text{ka4}) \\ \text{lax} &= \text{la} * \text{Cos}(\text{lameda} + \text{alpha} + \text{Delta4}) \\ \text{y} &= \text{zmax} - h1 - h2 - h3 - h4 * 2 / 3 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lay} &= \text{la} * \text{Sin}(\text{lameda} + \text{alpha} + \text{Delta4}) \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 + h4 * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * \text{y} - \text{lay} * \text{x} \\ \text{lasx} &= \text{deltalas} * \text{Cos}(\text{lameda} + \text{alpha} + \text{Delta4}) \\ \text{y} &= \text{zmax} - h1 - h2 - h3 - h4 * 0.4 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{lasy} &= \text{deltalas} * \text{Sin}(\text{lameda} + \text{alpha} + \text{Delta4}) \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 + h4 * 0.4) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lasx} * \text{y} - \text{lasy} * \text{x} \\ \text{N} &= \text{N} + \text{lay} + \text{lasy} \\ \text{T} &= \text{T} + \text{lax} + \text{lasx} \\ \text{escinst} &= \text{escinst} + (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value}) \end{aligned}$$

'sobrecarga

$$\begin{aligned} \text{lq1} &= \text{sq} * h1 * \text{ka1} \\ \text{lq1x} &= \text{lq1} * \text{Cos}(\text{Delta1} + \text{lameda} + \text{alpha}) \\ \text{lq1y} &= \text{lq1} * \text{Sin}(\text{Delta1} + \text{lameda} + \text{alpha}) \\ \text{y} &= \text{zmax} - h1 / 2 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lq1x} * \text{y} - \text{lq1y} * \text{x} \end{aligned}$$

$$\begin{aligned} \text{lq2} &= (\text{sq} + \text{gama1} * h1) * h2 * \text{ka1} \\ \text{lq2x} &= \text{lq2} * \text{Cos}(\text{Delta1} + \text{lameda} + \text{alpha}) \\ \text{lq2y} &= \text{lq2} * \text{Sin}(\text{Delta1} + \text{lameda} + \text{alpha}) \\ \text{y} &= \text{zmax} - h1 - h2 / 2 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 + h2 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lq2x} * \text{y} - \text{lq2y} * \text{x} \end{aligned}$$

$$\begin{aligned} \text{lq3} &= (\text{sq} + \text{gama1} * h1 + \text{gama2} * h2) * h3 * \text{ka3} \\ \text{lq3x} &= \text{lq3} * \text{Cos}(\text{Delta3} + \text{lameda} + \text{alpha}) \\ \text{lq3y} &= \text{lq3} * \text{Sin}(\text{Delta3} + \text{lameda} + \text{alpha}) \\ \text{y} &= \text{zmax} - h1 - h2 - h3 / 2 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lq3x} * \text{y} - \text{lq3y} * \text{x} \end{aligned}$$

$$\begin{aligned} \text{lq4} &= (\text{sq} + \text{gama1} * h1 + \text{gama2} * h2 + \text{gama3} * h3) * h4 * \text{ka4} \\ \text{lq4x} &= \text{lq4} * \text{Cos}(\text{Delta4} + \text{lameda} + \text{alpha}) \\ \text{lq4y} &= \text{lq4} * \text{Sin}(\text{Delta4} + \text{lameda} + \text{alpha}) \\ \text{y} &= \text{zmax} - h1 - h2 - h3 - h4 / 2 - \text{Range}("101!!" \& \text{linha5}).\text{Value} \\ \text{x} &= \text{xmax} + \text{Tan}(\text{lameda}) * (h1 + h2 + h3 + h4 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lq4x} * \text{y} - \text{lq4y} * \text{x} \end{aligned}$$

'água

$$lw = 0.5 * 9.81 * (znf - \text{Range}("101!I" & \text{linha4}).\text{Value})^2$$

$$lwx = lw * \text{Cos}(\text{lamedada} + \text{alpha})$$

$$lwy = lw * \text{Sin}(\text{lamedada} + \text{alpha})$$

$$y = z_{\text{max}} - h_1 - (h_2 + h_3 + h_4) * 2 / 3 - \text{Range}("101!I" & \text{linha5}).\text{Value}$$

$$x = x_{\text{max}} + \text{Tan}(\text{lamedada}) * (h_1 + (h_2 + h_3 + h_4) * 2 / 3) - \text{Range}("101!H" & \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lwx * y - lwy * x$$

$$T = T + lq1x + lq2x + lq3x + lq4x + lwx$$

$$N = N + lq1y + lq2y + lq3y + lq4y + lwy$$

$$\text{escinst} = \text{escinst} + (lq1x + lq2x + lq3x + lq4x + lwx) - (lq1y + lq2y + lq3y + lq4y + lwy) * \text{Tan}(\text{Range}("C11").\text{Value})$$

End If

Elsif zzseg <> True Then

If znf < zprim Then 'cam1 - cam2 - NF

'1ª camada

$$h_1 = z_{\text{max}} - \text{Range}("O7").\text{Value}$$

$$ka_1 = \text{Range}("F6").\text{Value}$$

$$kas = \text{Range}("F7").\text{Value}$$

$$\text{gama}_1 = \text{Range}("F3").\text{Value}$$

$$\text{Delta}_1 = \text{Range}("F5").\text{Value}$$

$$la = 0.5 * \text{gama}_1 * h_1^2 * ka_1$$

$$\text{deltalas} = 0.5 * \text{gama}_1 * h_1^2 * ((1 + kv) * kas - ka_1)$$

$$lax = la * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta}_1)$$

$$y = z_{\text{max}} - 2 / 3 * h_1 - \text{Range}("101!I" & \text{linha5}).\text{Value}$$

$$lay = la * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta}_1)$$

$$x = x_{\text{max}} + \text{Tan}(\text{lamedada}) * 2 / 3 * h_1 - \text{Range}("101!H" & \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lax * y - lay * x$$

$$\text{lasx} = \text{deltalas} * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta}_1)$$

$$y = z_{\text{max}} - 0.4 * h_1 - \text{Range}("101!I" & \text{linha5}).\text{Value}$$

$$\text{lasy} = \text{deltalas} * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta}_1)$$

$$x = x_{\text{max}} + \text{Tan}(\text{lamedada}) * 0.4 * h_1 - \text{Range}("101!H" & \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + \text{lasx} * y - \text{lasy} * x$$

$$T = T + lax + \text{lasx}$$

$$N = N + lay + \text{lasy}$$

$$\text{escinst} = (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value})$$

'2ª camada

$$h_2 = \text{Range}("O7").\text{Value} - \text{Range}("O9").\text{Value}$$

$$ka_2 = \text{Range}("I6").\text{Value}$$

$$kas = \text{Range}("I7").\text{Value}$$

```

gama2 = Range("I3").Value
Delta2 = Range("I5").Value
la = 0.5 * gama2 * h2 ^ 2 * ka2
deltas = 0.5 * gama2 * h2 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 2 / 3 * h2 - Range("'101!I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 2 / 3 * h2) - Range("'101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 0.4 * h2 - Range("'101!I" & linha5).Value
lasy = deltas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 0.4 * h2) - Range("'101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'NF
h3 = Range("O9").Value - Range("'101!I" & linha4).Value
gama3 = Range("I3").Value - 9.81
la = 0.5 * gama3 * h3 ^ 2 * ka2
deltas = 0.5 * gama3 * h3 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - h2 - h3 * 2 / 3 - Range("'101!I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 * 2 / 3) - Range("'101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - h2 - h3 * 0.4 - Range("'101!I" & linha5).Value
lasy = deltas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + h2 + h3 * 0.4) - Range("'101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("'101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 / 2) - Range("'101!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

```

$$lq2 = (sq + \text{gama1} * h1) * h2 * ka2$$

$$lq2x = lq2 * \text{Cos}(\text{Delta2} + \text{lamedada} + \text{alpha})$$

$$lq2y = lq2 * \text{Sin}(\text{Delta2} + \text{lamedada} + \text{alpha})$$

$$y = zmax - h1 - h2 / 2 - \text{Range}("101!I" \& \text{linha5}).\text{Value}$$

$$x = xmax + \text{Tan}(\text{lamedada}) * (h1 + h2 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lq2x * y - lq2y * x$$

$$lq3 = (sq + \text{gama1} * h1 + \text{gama2} * h2) * h3 * ka2$$

$$lq3x = lq3 * \text{Cos}(\text{Delta2} + \text{lamedada} + \text{alpha})$$

$$lq3y = lq3 * \text{Sin}(\text{Delta2} + \text{lamedada} + \text{alpha})$$

$$y = zmax - h1 - h2 - h3 / 2 - \text{Range}("101!!" \& \text{linha5}).\text{Value}$$

$$x = xmax + \text{Tan}(\text{lamedada}) * (h1 + h2 + h3 / 2) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lq3x * y - lq3y * x$$

'água

$$lw = 0.5 * 9.81 * (zmf - \text{Range}("101!!" \& \text{linha4}).\text{Value})^2$$

$$lwx = lw * \text{Cos}(\text{lamedada} + \text{alpha})$$

$$lwy = lw * \text{Sin}(\text{lamedada} + \text{alpha})$$

$$y = zmax - h1 - h2 - h3 * 2 / 3 - \text{Range}("101!!" \& \text{linha5}).\text{Value}$$

$$x = xmax + \text{Tan}(\text{lamedada}) * (h1 + h2 + (h3) * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lwx * y - lwy * x$$

$$T = T + lq1x + lq2x + lq3x + lwx$$

$$N = N + lq1y + lq2y + lq3y + lwy$$

$$\text{escinst} = \text{escinst} + (lq1x + lq2x + lq3x + lwx) - (lq1y + lq2y + lq3y + lwy) * \text{Tan}(\text{Range}("C11").\text{Value})$$

End If

If zmf > zprim Then 'cam1 - NF - cam2

'1ª camada

$$h1 = \text{Range}("O6").\text{Value} - \text{Range}("O9").\text{Value}$$

$$ka1 = \text{Range}("F6").\text{Value}$$

$$kas = \text{Range}("F7").\text{Value}$$

$$\text{gama1} = \text{Range}("F3").\text{Value}$$

$$\text{Delta1} = \text{Range}("F5").\text{Value}$$

$$la = 0.5 * \text{gama1} * h1^2 * ka1$$

$$\text{deltalas} = 0.5 * \text{gama1} * h1^2 * ((1 + kv) * kas - ka1)$$

$$lax = la * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta1})$$

$$y = zmax - 2 / 3 * h1 - \text{Range}("101!!" \& \text{linha5}).\text{Value}$$

$$lay = la * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta1})$$

$$x = xmax + \text{Tan}(\text{lamedada}) * 2 / 3 * h1 - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + lax * y - lay * x$$

$$\text{lasx} = \text{deltalas} * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta1})$$

$$y = zmax - 0.4 * h1 - \text{Range}("101!!" \& \text{linha5}).\text{Value}$$

$$\begin{aligned} \text{lasy} &= \text{deltalas} * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta1}) \\ x &= \text{xmax} + \text{Tan}(\text{lamedada}) * 0.4 * h1 - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lasy} * x \\ T &= T + \text{lax} + \text{lasy} \\ N &= N + \text{lay} + \text{lasy} \\ \text{escinst} &= (\text{lax} + \text{lasy}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value}) \end{aligned}$$

'NF

$$\begin{aligned} h2 &= \text{Range}("O9").\text{Value} - \text{Range}("O7").\text{Value} \\ \text{gama2} &= \text{Range}("F3").\text{Value} - 9.81 \\ la &= 0.5 * \text{gama2} * h2^2 * ka1 \\ \text{deltalas} &= 0.5 * \text{gama2} * h2^2 * ((1 + kv) * kas - ka1) \\ lax &= la * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta1}) \\ y &= \text{zmax} - h1 - h2 * 2 / 3 - \text{Range}("101!I" \& \text{linha5}).\text{Value} \\ lay &= la * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta1}) \\ x &= \text{xmax} + \text{Tan}(\text{lamedada}) * (h1 + h2 * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lay} * x \\ lax &= \text{deltalas} * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta1}) \\ y &= \text{zmax} - h1 - h2 * 0.4 - \text{Range}("101!I" \& \text{linha5}).\text{Value} \\ lasy &= \text{deltalas} * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta1}) \\ x &= \text{xmax} + \text{Tan}(\text{lamedada}) * (h1 + h2 * 0.4) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lasy} * x \\ T &= T + \text{lax} + \text{lasy} \\ N &= N + \text{lay} + \text{lasy} \\ \text{escinst} &= \text{escinst} + (\text{lax} + \text{lasy}) - (\text{lay} + \text{lasy}) * \text{Tan}(\text{Range}("C11").\text{Value}) \end{aligned}$$

'2ª camada

$$\begin{aligned} h3 &= \text{Range}("O7").\text{Value} - \text{Range}("101!I" \& \text{linha4}).\text{Value} \\ ka3 &= \text{Range}("I6").\text{Value} \\ kas &= \text{Range}("I7").\text{Value} \\ \text{gama3} &= \text{Range}("I3").\text{Value} - 9.81 \\ \text{Delta3} &= \text{Range}("I5").\text{Value} \\ la &= 0.5 * \text{gama3} * h3^2 * ka3 \\ \text{deltalas} &= 0.5 * \text{gama3} * h3^2 * ((1 + kv) * kas - ka3) \\ lax &= la * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta3}) \\ y &= \text{zmax} - h1 - h2 - h3 * 2 / 3 - \text{Range}("101!I" \& \text{linha5}).\text{Value} \\ lay &= la * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta3}) \\ x &= \text{xmax} + \text{Tan}(\text{lamedada}) * (h1 + h2 + h3 * 2 / 3) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lay} * x \\ lax &= \text{deltalas} * \text{Cos}(\text{lamedada} + \text{alpha} + \text{Delta3}) \\ y &= \text{zmax} - h1 - h2 - h3 * 0.4 - \text{Range}("101!I" \& \text{linha5}).\text{Value} \\ lasy &= \text{deltalas} * \text{Sin}(\text{lamedada} + \text{alpha} + \text{Delta3}) \\ x &= \text{xmax} + \text{Tan}(\text{lamedada}) * (h1 + h2 + h3 * 0.4) - \text{Range}("101!H" \& \text{linha5}).\text{Value} \\ \text{mdst} &= \text{mdst} + \text{lax} * y - \text{lasy} * x \end{aligned}$$

```

T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedas + alpha)
lq1y = lq1 * Sin(Delta1 + lamedas + alpha)
y = zmax - h1 / 2 - Range("'101'!" & linha5).Value
x = xmax + Tan(lamedas) * (h1 / 2) - Range("'101'H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

lq2 = (sq + gama1 * h1) * h2 * ka1
lq2x = lq2 * Cos(Delta2 + lamedas + alpha)
lq2y = lq2 * Sin(Delta2 + lamedas + alpha)
y = zmax - h1 - h2 / 2 - Range("'101'!" & linha5).Value
x = xmax + Tan(lamedas) * (h1 + h2 / 2) - Range("'101'H" & linha5).Value
mdst = mdst + lq2x * y - lq2y * x

lq3 = (sq + gama1 * h1 + gama2 * h2) * h3 * ka3
lq3x = lq3 * Cos(Delta3 + lamedas + alpha)
lq3y = lq3 * Sin(Delta3 + lamedas + alpha)
y = zmax - h1 - h2 - h3 / 2 - Range("'101'!" & linha5).Value
x = xmax + Tan(lamedas) * (h1 + h2 + h3 / 2) - Range("'101'H" & linha5).Value
mdst = mdst + lq3x * y - lq3y * x

'água
lw = 0.5 * 9.81 * (znf - Range("'101'!" & linha4).Value) ^ 2
lwx = lw * Cos(lamedas + alpha)
lwy = lw * Sin(lamedas + alpha)
y = zmax - h1 - (h2 + h3) * 2 / 3 - Range("'101'!" & linha5).Value
x = xmax + Tan(lamedas) * (h1 + (h2 + h3) * 2 / 3) - Range("'101'H" & linha5).Value
mdst = mdst + lwx * y - lwy * x
T = T + lq1x + lq2x + lq3x + lwx
N = N + lq1y + lq2y + lq3y + lwy
escinst = escinst + (lq1x + lq2x + lq3x + lwx) - (lq1y + lq2y + lq3y + lwy) * Tan(Range("C11").Value)

End If

End If

Elseif zprim <> True Then 'cam1 - NF
'1ª camada
h1 = Range("O6").Value - Range("O9").Value

```

```

ka1 = Range("F6").Value
kas = Range("F7").Value
gama1 = Range("F3").Value
Delta1 = Range("F5").Value
la = 0.5 * gama1 * h1 ^ 2 * ka1
deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - 2 / 3 * h1 - Range("'101!'I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("'101!'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta1)
y = zmax - 0.4 * h1 - Range("'101!'I" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 0.4 * h1 - Range("'101!'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'NF
h2 = Range("O9").Value - Range("'101!'I" & linha4).Value
gama2 = Range("F3").Value - 9.81
la = 0.5 * gama2 * h2 ^ 2 * ka1
deltalas = 0.5 * gama2 * h2 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - h1 - h2 * 2 / 3 - Range("'101!'I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * (h1 + h2 * 2 / 3) - Range("'101!'H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta1)
y = zmax - h1 - h2 * 0.4 - Range("'101!'I" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * (h1 + h2 * 0.4) - Range("'101!'H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("'101!'I" & linha5).Value

```

```

x = xmax + Tan(lamedada) * (h1 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

lq2 = (sq + gama1 * h1) * h2 * ka1
lq2x = lq2 * Cos(Delta1 + lamedada + alpha)
lq2y = lq2 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 - h2 / 2 - Range("101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq2x * y - lq2y * x

'água
lw = 0.5 * 9.81 * (zmf - Range("101!I" & linha4).Value) ^ 2
lwx = lw * Cos(lamedada + alpha)
lwy = lw * Sin(lamedada + alpha)
y = zmax - h1 - (h2) * 2 / 3 - Range("101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + (h2) * 2 / 3) - Range("101!H" & linha5).Value
mdst = mdst + lwx * y - lwy * x
T = T + lq1x + lq2x + lwx
N = N + lq1y + lq2y + lwy
escinst = escinst + (lq1x + lq2x + lwx) - (lq1y + lq2y + lwy) * Tan(Range("C11").Value)

```

End If

ElselF zzNF <> True Then 'sem Nível freático

If zzprim = True And zprim > zinf Then

```

If zzseg = True And zseg > zinf Then 'cam1 - cam2 - cam3
'1ª camada
h1 = zmax - Range("O7").Value
ka1 = Range("F6").Value
kas = Range("F7").Value
gama1 = Range("F3").Value
Delta1 = Range("F5").Value
la = 0.5 * gama1 * h1 ^ 2 * ka1
deltas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - 2 / 3 * h1 - Range("101!I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltas * Cos(lamedada + alpha + Delta1)
y = zmax - 0.4 * h1 - Range("101!I" & linha5).Value
lasy = deltas * Sin(lamedada + alpha + Delta1)

```



$$x = x_{\max} + \tan(\lambda_{\text{meda}}) * 0.4 * h_1 - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + \text{lasx} * y - \text{lasx} * x$$

$$T = T + \text{lax} + \text{lasx}$$

$$N = N + \text{lay} + \text{lasx}$$

$$\text{escinst} = (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasx}) * \tan(\text{Range}("C11").\text{Value})$$

'2ª camada

$$h_2 = \text{Range}("O7").\text{Value} - \text{Range}("O8").\text{Value}$$

$$ka_2 = \text{Range}("I6").\text{Value}$$

$$kas = \text{Range}("I7").\text{Value}$$

$$\gamma_2 = \text{Range}("I3").\text{Value}$$

$$\Delta_2 = \text{Range}("I5").\text{Value}$$

$$l_a = 0.5 * \gamma_2 * h_2^2 * ka_2$$

$$\text{deltal}_a = 0.5 * \gamma_2 * h_2^2 * ((1 + kv) * kas - ka_2)$$

$$\text{lax} = l_a * \cos(\lambda_{\text{meda}} + \alpha + \Delta_2)$$

$$y = z_{\max} - h_1 - 2/3 * h_2 - \text{Range}("101!I" \& \text{linha5}).\text{Value}$$

$$\text{lay} = l_a * \sin(\lambda_{\text{meda}} + \alpha + \Delta_2)$$

$$x = x_{\max} + \tan(\lambda_{\text{meda}}) * (h_1 + 2/3 * h_2) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + \text{lax} * y - \text{lay} * x$$

$$\text{lasx} = \text{deltal}_a * \cos(\lambda_{\text{meda}} + \alpha + \Delta_2)$$

$$y = z_{\max} - h_1 - 0.4 * h_2 - \text{Range}("101!I" \& \text{linha5}).\text{Value}$$

$$\text{lasx} = \text{deltal}_a * \sin(\lambda_{\text{meda}} + \alpha + \Delta_2)$$

$$x = x_{\max} + \tan(\lambda_{\text{meda}}) * (h_1 + 0.4 * h_2) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + \text{lasx} * y - \text{lasx} * x$$

$$T = T + \text{lax} + \text{lasx}$$

$$N = N + \text{lay} + \text{lasx}$$

$$\text{escinst} = \text{escinst} + (\text{lax} + \text{lasx}) - (\text{lay} + \text{lasx}) * \tan(\text{Range}("C11").\text{Value})$$

'3ª camada

$$h_3 = \text{Range}("O8").\text{Value} - \text{Range}("101!I" \& \text{linha4}).\text{Value}$$

$$ka_3 = \text{Range}("L6").\text{Value}$$

$$kas = \text{Range}("L7").\text{Value}$$

$$\gamma_3 = \text{Range}("L3").\text{Value}$$

$$\Delta_3 = \text{Range}("L5").\text{Value}$$

$$l_a = 0.5 * \gamma_3 * h_3^2 * ka_3$$

$$\text{deltal}_a = 0.5 * \gamma_3 * h_3^2 * ((1 + kv) * kas - ka_3)$$

$$\text{lax} = l_a * \cos(\lambda_{\text{meda}} + \alpha + \Delta_3)$$

$$y = z_{\max} - h_1 - h_2 - 2/3 * h_3 - \text{Range}("101!I" \& \text{linha5}).\text{Value}$$

$$\text{lay} = l_a * \sin(\lambda_{\text{meda}} + \alpha + \Delta_3)$$

$$x = x_{\max} + \tan(\lambda_{\text{meda}}) * (h_1 + h_2 + 2/3 * h_3) - \text{Range}("101!H" \& \text{linha5}).\text{Value}$$

$$\text{mdst} = \text{mdst} + \text{lax} * y - \text{lay} * x$$

$$\text{lasx} = \text{deltal}_a * \cos(\lambda_{\text{meda}} + \alpha + \Delta_3)$$

$$y = z_{\max} - h_1 - h_2 - 0.4 * h_3 - \text{Range}("101!I" \& \text{linha5}).\text{Value}$$

$$\text{lasx} = \text{deltal}_a * \sin(\lambda_{\text{meda}} + \alpha + \Delta_3)$$

```

x = xmax + Tan(lamedada) * (h1 + h2 + 0.4 * h3) - Range("101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

lq2 = (sq + gama1 * h1) * h2 * ka2
lq2x = lq2 * Cos(Delta2 + lamedada + alpha)
lq2y = lq2 * Sin(Delta2 + lamedada + alpha)
y = zmax - h1 - h2 / 2 - Range("101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq2x * y - lq2y * x

lq3 = (sq + gama1 * h1 + gama2 * h2) * h3 * ka3
lq3x = lq3 * Cos(Delta3 + lamedada + alpha)
lq3y = lq3 * Sin(Delta3 + lamedada + alpha)
y = zmax - h1 - h2 - h3 / 2 - Range("101!I" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 + h3 / 2) - Range("101!H" & linha5).Value
mdst = mdst + lq3x * y - lq3y * x
T = T + lq1x + lq2x + lq3x
N = N + lq1y + lq2y + lq3y
escinst = escinst + (lq1x + lq2x + lq3x) - (lq1y + lq2y + lq3y) * Tan(Range("C11").Value)

Elseif zzseg <> True Then 'cam1 - cam2

'1ª camada
h1 = zmax - Range("O7").Value
ka1 = Range("F6").Value
kas = Range("F7").Value
gama1 = Range("F3").Value
Delta1 = Range("F5").Value
la = 0.5 * gama1 * h1 ^ 2 * ka1
deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - 2 / 3 * h1 - Range("101!I" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)

```

---

```

x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("'101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta1)
y = zmax - 0.4 * h1 - Range("'101!!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 0.4 * h1 - Range("'101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'2ª camada
h2 = Range("O7").Value - Range("'101!!" & linha4).Value
ka2 = Range("I6").Value
kas = Range("I7").Value
gama2 = Range("I3").Value
Delta2 = Range("I5").Value
la = 0.5 * gama2 * h2 ^ 2 * ka2
deltalas = 0.5 * gama2 * h2 ^ 2 * ((1 + kv) * kas - ka2)
lax = la * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 2 / 3 * h2 - Range("'101!!" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 2 / 3 * h2) - Range("'101!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta2)
y = zmax - h1 - 0.4 * h2 - Range("'101!!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta2)
x = xmax + Tan(lamedada) * (h1 + 0.4 * h2) - Range("'101!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = escinst + (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("'101!!" & linha5).Value
x = xmax + Tan(lamedada) * (h1 / 2) - Range("'101!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x

lq2 = (sq + gama1 * h1) * h2 * ka2
lq2x = lq2 * Cos(Delta2 + lamedada + alpha)
lq2y = lq2 * Sin(Delta2 + lamedada + alpha)

```

```

y = zmax - h1 - h2 / 2 - Range("101!!" & linha5).Value
x = xmax + Tan(lamedada) * (h1 + h2 / 2) - Range("101!!H" & linha5).Value
mdst = mdst + lq2x * y - lq2y * x
T = T + lq1x + lq2x
N = N + lq1y + lq2y
escinst = escinst + (lq1x + lq2x) - (lq1y + lq2y) * Tan(Range("C11").Value)

End If

Elseif zzprim <> True Then 'cam1

'1ª camada
h1 = zmax - Range("101!!" & linha4).Value
ka1 = Range("F6").Value
kas = Range("F7").Value
gama1 = Range("F3").Value
Delta1 = Range("F5").Value
la = 0.5 * gama1 * h1 ^ 2 * ka1
deltalas = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kas - ka1)
lax = la * Cos(lamedada + alpha + Delta1)
y = zmax - 2 / 3 * h1 - Range("101!!" & linha5).Value
lay = la * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 2 / 3 * h1 - Range("101!!H" & linha5).Value
mdst = mdst + lax * y - lay * x
lasx = deltalas * Cos(lamedada + alpha + Delta1)
y = zmax - 0.4 * h1 - Range("101!!" & linha5).Value
lasy = deltalas * Sin(lamedada + alpha + Delta1)
x = xmax + Tan(lamedada) * 0.4 * h1 - Range("101!!H" & linha5).Value
mdst = mdst + lasx * y - lasy * x
T = T + lax + lasx
N = N + lay + lasy
escinst = (lax + lasx) - (lay + lasy) * Tan(Range("C11").Value)

'sobrecarga
lq1 = sq * h1 * ka1
lq1x = lq1 * Cos(Delta1 + lamedada + alpha)
lq1y = lq1 * Sin(Delta1 + lamedada + alpha)
y = zmax - h1 / 2 - Range("101!!" & linha5).Value
x = xmax + Tan(lamedada) * (h1 / 2) - Range("101!!H" & linha5).Value
mdst = mdst + lq1x * y - lq1y * x
T = T + lq1x
N = N + lq1y
escinst = escinst + (lq1x) - (lq1y) * Tan(Range("C11").Value)

```

End If

End If

'=====

'cálculo do impulso passivo

'=====

lamedap = Range("T9").Value

zinf = Range("'101!I" & linha5).Value

If zzNFF = True And zNFF > zinf And zpass > zinf Then

'1ª camada

h1 = zpass - zNFF

kp1 = Range("T6").Value

kps = Range("T7").Value

gama1 = Range("T3").Value

Delta1 = Range("T5").Value

lp = 0.5 \* gama1 \* h1 ^ 2 \* kp1

deltalps = 0.5 \* gama1 \* h1 ^ 2 \* ((1 + kv) \* kps - kp1)

lpx = lp \* Cos(lamedap + alpha + Delta1)

y = zpass - 2 / 3 \* h1 - Range("'101!I" & linha5).Value

lpy = lp \* Sin(lamedap + alpha + Delta1)

x = Range("N11").Value - Tan(lamedap + alpha) \* 2 / 3 \* h1 - Range("'101!H" & linha5).Value

mdst = mdst - lpx \* y - lpy \* x

lpsx = deltalps \* Cos(lamedap + alpha + Delta1)

y = zpass - 0.4 \* h1 - Range("'101!I" & linha5).Value

lpsy = deltalps \* Sin(lamedap + alpha + Delta1)

x = Range("N11").Value - Tan(lamedap + alpha) \* 0.4 \* h1 - Range("'101!H" & linha5).Value

mdst = mdst - lpsx \* y - lpsy \* x

T = T - lpx - lpsx

N = N + lpy + lpsy

escinst = escinst - (lpx + lpsx) - (lpy + lpsy) \* Tan(Range("C11").Value)

'NF

h2 = zNFF - zinf

gama2 = Range("T3").Value - 9.81

lp = 0.5 \* gama2 \* h2 ^ 2 \* kp1

deltalps = 0.5 \* gama2 \* h2 ^ 2 \* ((1 + kv) \* kps - kp1)

lpx = lp \* Cos(lamedap + alpha + Delta1)

y = zpass - h1 - h2 \* 2 / 3 - Range("'101!I" & linha5).Value

lpy = lp \* Sin(lamedap + alpha + Delta1)

x = Range("N11").Value - Tan(lamedap + alpha) \* (h1 + h2 \* 2 / 3) - Range("'101!H" & linha5).Value

mdst = mdst - lpx \* y - lpy \* x

```

lpsx = delta1ps * Cos(lamedap + alpha + Delta1)
y = zpass - h1 - h2 * 0.4 - Range("101!!" & linha5).Value
lpsy = delta1ps * Sin(lamedap + alpha + Delta1)
x = Range("N11").Value - Tan(lamedap + alpha) * (h1 + h2 * 0.4) - Range("101!H" & linha5).Value
mdst = mdst - lpsx * y - lpsy * x
T = T - lpx - lpsy
N = N + lpy + lpsy
escinst = escinst - (lpx + lpsy) - (lpy + lpsy) * Tan(Range("C11").Value)

```

'água

```

lw = 0.5 * 9.81 * (zNFF - Range("101!!" & linha5).Value) ^ 2
lwx = lw * Cos(lamedap + alpha)
lwy = lw * Sin(lamedap + alpha)
y = zpass - h1 - (h2) * 2 / 3 - Range("101!!" & linha5).Value
x = Range("N11").Value - Tan(lamedap + alpha) * (h1 + (h2) * 2 / 3) - Range("101!H" & linha5).Value
mdst = mdst - lwx * y - lwy * x
T = T - lwx
N = N + lwy
escinst = escinst - (lwx) - (lwy) * Tan(Range("C11").Value)

```

Elselz zzNF <> True And zpass > zinff Then 'sem Nível freático

'1ª camada

```

h1 = zpass - zinff
kp1 = Range("T6").Value
kps = Range("T7").Value
gama1 = Range("T3").Value
Delta1 = Range("T5").Value
lp = 0.5 * gama1 * h1 ^ 2 * kp1
delta1ps = 0.5 * gama1 * h1 ^ 2 * ((1 + kv) * kps - kp1)
lpx = lp * Cos(lamedap + alpha + Delta1)
y = zpass - 2 / 3 * h1 - Range("101!!" & linha5).Value
lpy = lp * Sin(lamedap + alpha + Delta1)
x = Range("N11").Value - Tan(lamedap + alpha) * 2 / 3 * h1 - Range("101!H" & linha5).Value
mdst = mdst - lpx * y - lpy * x
lpsx = delta1ps * Cos(lamedap + alpha + Delta1)
y = zpass - 0.4 * h1 - Range("101!!" & linha5).Value
lpsy = delta1ps * Sin(lamedap + alpha + Delta1)
x = Range("N11").Value - Tan(lamedap + alpha) * 0.4 * h1 - Range("101!H" & linha5).Value
mdst = mdst - lpsx * y - lpsy * x
T = T - lpx - lpsy
N = N + lpy + lpsy
escinst = escinst - (lpx + lpsy) - (lpy + lpsy) * Tan(Range("C11").Value)

```

End If

```
'=====
'cálculo
'=====
```

Range("U16").Value = mdst

Range("R16").Value = N

Range("S16").Value = T

Mo = mstb - mdst

Range("V16").Value = Mo

eo = Mo / N

Range("W16").Value = eo

xb = Range("'101!H" & linha5).Value + eo \* Cos(alpha)

yb = Range("'101!I" & linha5).Value - eo \* Sin(alpha)

Nx = xb + N / 100 \* Sin(alpha)

Ny = yb + N / 100 \* Cos(alpha)

Tx = xb + T / 100 \* Cos(alpha)

Ty = yb - T / 100 \* Sin(alpha)

Rx = Tx + N / 100 \* Sin(alpha)

Ry = Ny - T / 100 \* Sin(alpha)

Range("Y4").Value = xb

Range("Z4").Value = yb

Range("Y5").Value = Rx

Range("Z5").Value = Ry

Sheets("111").Select

Range("M4:Q4").Select

Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

Range("M4").Value = linha1 - 1

Range("N4").Value = (N ^ 2 + T ^ 2) ^ 0.5

Range("'109!AA1").Value = escinst

If escinst > escest Then

    Range("O4").Value = "OK"

Else

    Range("O4").Value = "KO"

End If

If mstb > mdst Then

    Range("P4").Value = "OK"

Else

    Range("P4").Value = "KO"

End If

Sheets("112").Select

linha1 = linha1 + 1

linha2 = linha2 + 6

linha3 = linha3 + 6

linha4 = linha4 + 6

linha5 = linha5 + 6

ncamada = ncamada - 1

Wend

' aranjear o gráfico

Sheets("111").Select

ActiveSheet.ChartObjects("Chart 1").Activate

ActiveChart.Axes(xlValue).MinimumScale = Range("'101!B18").Value

ActiveChart.Axes(xlValue).MaximumScale = Range("'101!C18").Value

ActiveChart.Axes(xlCategory).MinimumScale = Range("'101!D18").Value

ActiveChart.Axes(xlCategory).MaximumScale = Range("'101!E18").Value

End Sub