

### 3

## Implementação das Ferramentas Numéricas

O Método de Análise Limite tem como objetivo encontrar a carga de colapso de uma estrutura, como foi revisado no capítulo anterior, o qual pode ser estudado como um problema de programação matemática. Esta solução deverá ser encontrada utilizando um processo iterativo no qual as restrições de igualdade e desigualdade do problema sejam respeitadas usando para isso algoritmos de otimização matemática, e usando o MEF como ferramenta para a formulação numérica do problema de Análise Limite.

No MEF (Desai 1979; Bathe, 1982; Zienkiewicz e Taylor, 1989), o meio contínuo é dividido em sub-regiões, chamadas elementos. Esses elementos são ligados entre si através de pontos situados em suas arestas e vértices, chamados nós. As variáveis de interesse e as propriedades do meio são definidas, no interior do elemento, pela interpolação de seus valores nodais utilizando-se funções pré-estabelecidas chamadas funções de interpolação ou de base. O MEF apresenta maior flexibilidade em descrever geometrias multidimensionais com contornos e cargas irregulares em comparação ao Método das Diferenças Finitas (MDF). O MEF tem maior facilidade na inclusão das propriedades em um meio não homogêneo.

Na área de análise de estruturas, a formulação do MEF pode ser feita a partir do Princípio da Mínima Energia Potencial Total, do Método de Resíduos Ponderados ou do Princípio Trabalho Virtual. Ele usa conceitos de “discretização” do contínuo e de “matriz de interpolação” que fornece os deslocamentos em um ponto no interior do elemento em função de seus deslocamentos nodais. O termo discretização se refere a um modelo com um número finito de incógnitas para a análise de meios contínuos em contraposição a uma análise como a feita pela Teoria de Elasticidade que usam funções contínuas como solução.

### 3.1. Formulação Numérica da Análise Limite pelo Método de Elementos Finitos

Neste trabalho é usada a formulação classificada como mista fraca. A formulação se baseia no princípio dos deslocamentos virtuais, o qual estabelece que, para qualquer pequeno movimento cinematicamente admissível, o trabalho das forças externas será igual ao trabalho das forças internas, além disso, considera-se que os elementos são de tensão constante.

#### 3.1.1. Condição de Equilíbrio

Na formulação mista fraca a condição de equilíbrio é garantida pelo princípio do trabalho virtual. Este princípio estabelece que, para qualquer campo de velocidades cinematicamente admissível, o trabalho das forças internas é igual ao trabalho das forças externas. Para o caso em estudo que é a estabilidade de taludes, as forças externas estão representadas pelo peso próprio do material, as quais atuam como forças que fazem um desequilíbrio na estrutura do solo. As forças internas são geradas pelo estado de tensão presente no corpo do solo. Dessa forma o princípio de deslocamentos virtuais é formulado da forma apresentada na Equação (3.1):

$$\begin{aligned} \text{Trabalho interno:} \quad & \int_V \left\{ \dot{\delta \varepsilon} \right\}^T \{ \sigma \} dV \\ \text{Trabalho externo:} \quad & \int_V \left\{ \dot{\delta u} \right\}^T \lambda \{ F_o \} dV \end{aligned} \tag{3.1}$$

onde:

$\dot{\delta \varepsilon}$  : Campo de velocidade de deformação.

$\dot{\delta u}$  : Campo de velocidades.

$F_o$  : Vetor de Força de Volume

$\sigma$  : Campo de Tensões.

$\lambda$  : Fator do colapso do sistema.

Aplicando-se a Equação (3.1) a um elemento e considerando, da teoria dos elementos finitos, que existe uma matriz de interpolação ( $N$ ) que relaciona os campos de velocidades com os campos de velocidades nodais, e uma matriz que relaciona o campo de velocidade de deformação com o campo de velocidades nodais conhecida como matriz de compatibilidade cinemática ( $B$ ), desse modo tem-se, para cada elemento finito:

$$\begin{aligned}\left\{\delta \dot{u}\right\}^T &= \{\delta \hat{u}\}^T [N_u]^T \\ \left\{\delta \dot{\varepsilon}\right\}^T &= \{\delta \hat{u}\}^T [B]^T \\ \{\sigma\} &= [N_\sigma] \left\{\hat{\sigma}\right\}^T\end{aligned}\quad (3.2)$$

Onde:

- $\{\hat{u}\}$  : Campo de Velocidades Nodais.
- $\{\sigma\}$  : Campo de Tensões
- $\{\hat{\sigma}\}$  : Campo de Tensões Nodais.
- $\{\dot{\varepsilon}\}$  : Campo de velocidade de deformação.
- $[N_u]$  : Matriz de interpolação de velocidades.
- $[N_\sigma]$  : Matriz de interpolação de tensões.
- $[B]$  : Matriz de compatibilidade cinemática.

Substituindo as Equações (3.2) nas Equações (3.1), tem-se a seguinte igualdade:

$$\int_V \{\delta \hat{u}\}^T [B]^T [N_\sigma] \{\hat{\sigma}\} dV = \lambda \int_V \{\delta \hat{u}\}^T [N_u]^T \{F_o\} dV \quad (3.3)$$

Fatorizando:

$$\{\delta \hat{u}\}^T \left( \int_V [B]^T [N_\sigma] \{\hat{\sigma}\} dV - \lambda \int_V [N_u]^T \{F_o\} dV \right) = 0 \quad (3.4)$$

Como os deslocamentos virtuais ( $\{\delta \hat{u}\}$ ) são arbitrários diferentes de zero, pode-se obter:

$$\left( \int_V [B]^T [N_\sigma] \{\hat{\sigma}\} dV - \lambda \int_V [N_u]^T \{Fo\} dV \right) = 0 \quad (3.5)$$

Considerando a matriz de equilíbrio  $G$  como:

$$\int_V [B]^T [N_\sigma] dV = [G] \quad (3.6)$$

Considerando o caso de estabilidade de taludes em encostas, só a força de volume devido ao peso próprio material foi considerada na implementação. O vetor de forças externas aplicadas fica da seguinte forma:

$$f_0 = \int_V [N_u]^T \{Fo\} dV \quad (3.7)$$

Com a discretização via Elementos Finitos, a Equação (3.5) pode ser reescrita na forma:

$$[G] \{\hat{\sigma}\} = \lambda f_0 \quad (3.8)$$

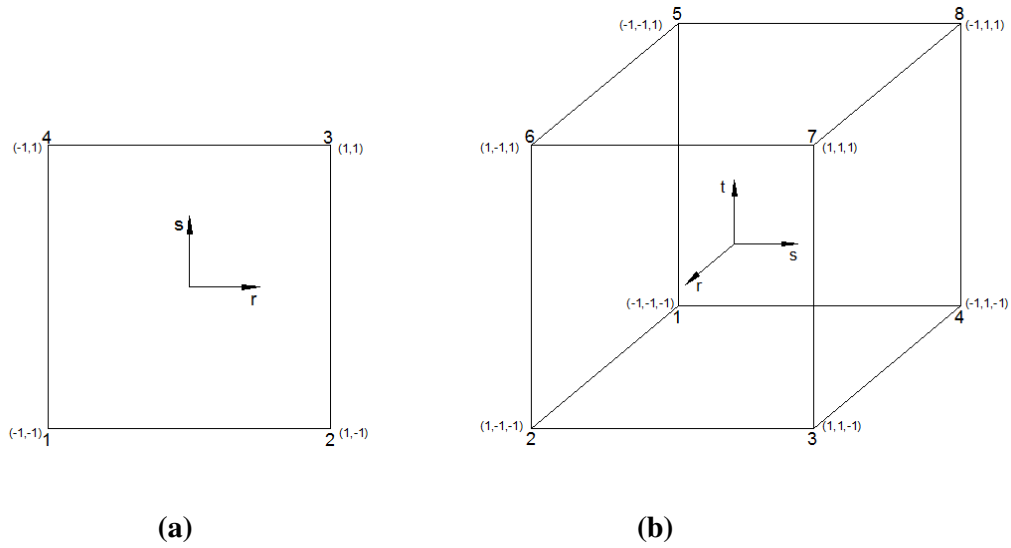
A Equação (3.8) é a condição de equilíbrio para um elemento, que tem como incógnitas as componentes de tensão no elemento e o vetor das forças equivalentes nodais. Quando os elementos são considerados de tensão constante a matriz  $N_\sigma$  é igual à matriz identidade  $I$ . Nesse caso, as Equações (3.6) e (3.8) ficam da seguinte forma:

$$\int_V [B]^T dV = [G] \quad (3.9)$$

$$[G] \{\sigma\} = \lambda f_0 \quad (3.10)$$

### 3.1.2. Discretização Espacial

Na discretização espacial foram utilizados dois tipos de elementos: o elementos isoparamétrico quadrilateral de quatro nós (Q4) em 2D e o elemento isoparamétrico hexaédrico de oito nós (BRICK8) em 3D (Bathe, 1982; Zienkiewicz e Taylor, 1989), os quais são mostrados no plano paramétrico na Figura 3.1 (a) e (b):



**Figura 3.1 - (a) Elemento Quadrilateral, (b) Elemento Hexaédrico**

As funções de interpolação para os elementos usados na implementação, no caso de elemento de elemento quadrilateral de quatro nós:

$$\begin{aligned} N_1 &= \frac{1}{4}(1-r)(1-s) & N_2 &= \frac{1}{4}(1+r)(1-s) \\ N_3 &= \frac{1}{4}(1+r)(1+s) & N_4 &= \frac{1}{4}(1-r)(1+s) \end{aligned} \quad (3.11)$$

As funções de interpolação no caso de elemento hexaédrico de oito nós, são as seguintes

$$\begin{aligned} N_1 &= \frac{1}{8}(1-r)(1-s)(1-t) & N_2 &= \frac{1}{8}(1+r)(1-s)(1-t) \\ N_3 &= \frac{1}{8}(1+r)(1+s)(1-t) & N_4 &= \frac{1}{8}(1-r)(1+s)(1-t) \\ N_5 &= \frac{1}{8}(1-r)(1-s)(1+t) & N_6 &= \frac{1}{8}(1+r)(1-s)(1+t) \\ N_7 &= \frac{1}{8}(1+r)(1+s)(1+t) & N_8 &= \frac{1}{8}(1-r)(1+s)(1+t) \end{aligned} \quad (3.12)$$

sendo  $r$ ,  $s$  e  $t$  as coordenadas paramétricas dos elementos.

As Equações (3.6) para a matriz de equilíbrio  $G$  e as Equações (3.7) para o vetor  $f_0$  referente à estrutura global podem ser obtidas como indicado a seguir, onde a matriz e o vetor para cada elemento passam a ser denominados  $G_e$  e  $fe_0$ :

$$[G] = \sum_e [G_e] = \sum_e \int_{\Omega^e} [B_e]^T d\Omega^e \quad (3.13)$$

$$\{f_0\} = \sum_e \{fe_0\} = \sum_e \int_{\Omega^e} [N]^T \{Fo^e\} d\Omega^e \quad (3.14)$$

onde:

$\Omega^e$  : representa o domínio elementar.

$\sum_e$  : representa a montagem da matriz ou vetor global.

Fazendo uso da integração numérica de acordo com o tipo de elemento utilizado, a matriz  $G_e$  e o vetor  $fe_0$  podem ser reescritos da seguinte forma:

$$[G_e] = \sum_e \int [B_e]^T |J_e| drdsdt \quad (3.15)$$

$$\{fe_0\} = \sum_e \int [N]^T \{Fo^e\} |J_e| drdsdt \quad (3.16)$$

onde:

$r, s$  e  $t$ : são as coordenadas paramétricas dos elementos.

$J_e$ : Matriz Jacobiana do elemento  $e$ .

$Fo^e$ : Vetor de forças de corpo no elemento  $e$ .

A matriz que relaciona deformação com deslocamento para o elemento  $e$   $[B_e]$  pode ser escrita em função das derivadas das funções de interpolação em relação das coordenadas globais, da seguinte forma:

$$[B_e] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \cdots & \frac{\partial N_n}{\partial x} \\ \frac{\partial N_1}{\partial y} & \cdots & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial z} & \cdots & \frac{\partial N_n}{\partial z} \end{bmatrix} \quad (3.17)$$

O determinante da matriz Jacobiana do elemento  $e$  ( $|J_e|$ ) pode ser interpretado como um fator de escala entre o sistema cartesiano global e o sistema paramétrico local de coordenadas. A matriz Jacobiana, relaciona as derivadas em relação às coordenadas no sistema local com derivadas em relação às coordenadas no sistema global:

$$\begin{Bmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{Bmatrix} = [J_e] \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} \quad (3.18)$$

Nos elementos isoparamétricos, as funções de forma, responsáveis pelo mapeamento das coordenadas globais nas suas equivalentes paramétricas e vice-versa, são as próprias funções de interpolação, assim:

$$x = \sum_{i=1}^n N_i x_i \quad y = \sum_{i=1}^n N_i y_i \quad z = \sum_{i=1}^n N_i z_i \quad (3.19)$$

Então:

$$\frac{\partial x}{\partial r} = \sum_{i=1}^n \frac{\partial N_i}{\partial r} x_i \quad \frac{\partial x}{\partial s} = \sum_{i=1}^n \frac{\partial N_i}{\partial s} x_i \quad \frac{\partial x}{\partial t} = \sum_{i=1}^n \frac{\partial N_i}{\partial t} x_i \quad (3.20)$$

Segue-se raciocínio análogo para as coordenadas globais nos eixos  $y$  e  $z$ . Desta forma, a matriz  $[B_e]$  pode ser definida como:

$$[B_e] = [J_e]^{-1} \begin{bmatrix} \frac{\partial N_1}{\partial r} & \cdots & \frac{\partial N_n}{\partial r} \\ \frac{\partial N_1}{\partial s} & \cdots & \frac{\partial N_n}{\partial s} \\ \frac{\partial N_1}{\partial t} & \cdots & \frac{\partial N_n}{\partial t} \end{bmatrix} \quad (3.21)$$

De acordo com isso a matriz  $G_e$  e o vetor  $fe_0$  são obtidos pela integração numérica das Equações (3.15) e (3.16).

### 3.1.3. Integração Espacial

A integração da matriz  $G_e$  e do vetor  $fe_0$  não pode ser feita analiticamente, então adotou-se a quadratura de Gauss para sua avaliação numérica.

Na quadratura de Gauss, a função a ser integrada é avaliada em determinados pontos, chamados pontos de integração ou pontos de Gauss. A integração é feita através da soma ponderada dos valores assumidos pela respectiva função nestes pontos de integração.

$$\int f(x)dx = \sum_{i=1}^n W_i f(x_i) \quad (3.22)$$

Onde  $f(x)$  é a função cuja integral se deseja avaliar,  $W_i$  são os pesos de ponderação e  $x_i$  são os pontos onde a função é avaliada.

Na quadratura de Gauss os valores dos pontos de integração correspondem as raízes dos polinômios de Legendre. Os pontos de integração, assim como os fatores de ponderação, são determinados de maneira de maximizar a precisão da técnica de integração. A integração numérica pode apresentar elevado custo computacional, devido ao número de vezes que cada matriz deve ser avaliada. Como a integração numérica pode introduzir uma nova fonte de erro na solução do problema, o número de pontos de integração deve ser o mínimo necessário, de modo a preservar a ordem de grandeza do erro de discretização (Zienckiewicz e Taylor, 1989).

Neste trabalho, a integração do elemento quadrilateral de 4 nós foi feita com 4 pontos de integração e do elemento hexaédrico de 8 nós com 8 pontos de integração. As tabelas 3.1 e 3.2 apresentam os pontos de integração (Pontos de Gauss) e os respectivos fatores de ponderação.



**Tabela 3.1 - Coordenadas paramétricas dos pontos de integração e fatores de ponderação para o elemento quadrilateral de 4 nós.**

$i$	Pontos de Integração		Fator de Ponderação	
	$r_i$	$s_i$	$W_{ri}$	$W_{si}$
1	-0.57735	-0.57735	1	1
2	0.57735	-0.57735	1	1
3	0.57735	0.57735	1	1
4	-0.57735	0.57735	1	1

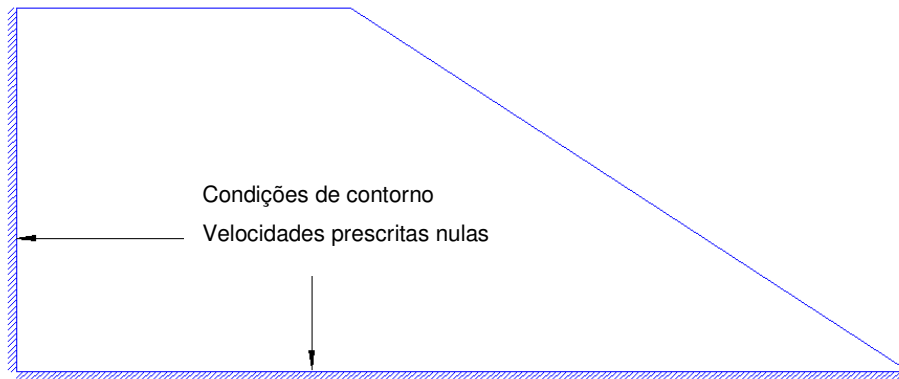
**Tabela 3.2 - Coordenadas paramétricas dos pontos de integração e fatores de ponderação para o elemento hexaédrico de 8 nós.**

$i$	Pontos de Integração			Fator de Ponderação		
	$r_i$	$s_i$	$t_i$	$W_{ri}$	$W_{si}$	$W_{ti}$
1	-0.57735	-0.57735	-0.57735	1	1	1
2	0.57735	-0.57735	-0.57735	1	1	1
3	0.57735	0.57735	-0.57735	1	1	1
4	-0.57735	0.57735	-0.57735	1	1	1
5	-0.57735	-0.57735	0.57735	1	1	1
6	0.57735	-0.57735	0.57735	1	1	1
7	0.57735	0.57735	0.57735	1	1	1
8	-0.57735	0.57735	0.57735	1	1	1

A integração das Equações (3.15) e (3.16) e a montagem das matrizes elementares, resultando na matriz global, produz um sistema de equações, onde as incógnitas são as tensões e o fator de colapso, este sistema de equações está dado pela Equação (3.10) apresentada na formulação pelo MEF.

### 3.1.4. Condição de Contorno

Para que a solução forme um sistema de equações, precisam ser fornecidas as condições de contorno para a solução do problema. Como se pode ver na Figura 3.2, deve-se especificar os nós e suas direções de velocidade prescritas nulas.



**Figura 3.2 - Superfície de escoamento e o vetor de deformação plástica.**

Determinadas as direções de velocidades prescritas nulas, deve-se eliminar as correspondentes linhas na matriz  $G$  e no vetor  $f_0$  na Equação (3.10).

### 3.1.5. Condição de Escoamento

Sabendo-se que para que o estado de tensão na solução da Equação (3.10) seja considerado admissível, que é uns dos requisitos do teorema de limite inferior, deve-se garantir que este satisfaça o critério de escoamento, ou seja,

$$F(\sigma) < 0 \quad (3.23)$$

O estado de tensões deve ser garantido em todos os elementos que conformam a malha de elementos finitos do problema, o critério de escoamento do material deve ser satisfeito.

$$F(\sigma)_i < 0 \text{ para } i=1...n \quad (3.24)$$

onde  $n$  é o número de elementos que compõem a malha de elementos finitos.

No caso de problemas 2D os critérios de escoamento usados foram o critério de Mohr Coulomb e Drucker-Prager, e no caso de problemas 3D o critério utilizado foi o critério de escoamento de Drucker-Prager. Estes critérios de escoamento foram colocados de duas formas, para cada caso, a primeira forma é a forma convencional, ou seja, aquelas apresentadas nas Equações (2.9) e (2.10) do capítulo dois; a segunda forma é usando uma transformação de maneira a colocar estas restrições na forma cônica quadrática de segunda ordem.

### 3.1.5.1.

#### **Critério de Mohr-Coulomb na forma Cônica Quadrática**

De acordo com a revisão bibliográfica, para estado plano de deformações, o critério de Mohr-Coulomb pode ser expresso na forma da Equação (25)

$$\sqrt{(\sigma_x - \sigma_y)^2 + 4\tau_{xy}^2} + (\sigma_x + \sigma_y)\sin\phi - 2c\cos\phi \leq 0 \quad (3.25)$$

Mais adiante se vão usar expressões matriciais onde será útil o vetor de tensões  $\sigma$  como definido a seguir:

$$\sigma = \begin{bmatrix} \sigma_x & \sigma_y & \tau_{xy} \end{bmatrix}_i^T \text{ para } i=1\dots n \quad (3.26)$$

onde  $c$  é a coesão e  $\phi$  é o ângulo de atrito,  $n$  é o numero de elementos da malha de elementos finitos. Adotando a seguinte notação,

$$\sigma_m = \frac{1}{2}(\sigma_x + \sigma_y), \quad s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij} \quad (3.27)$$

$$\begin{aligned} \sigma_{xx} &= \sigma_x \\ \sigma_{yy} &= \sigma_y \\ \sigma_{xy} &= \tau_{xy} \end{aligned} \quad (3.28)$$

onde  $\delta$  é o delta de Kronecker,  $\sigma_m$  é a tensão media,  $s_{ij}$  são os componentes do tensor desviador.

Substituindo a Equação (3.28) na Equação (3.27), tem-se

$$\begin{aligned} s_{xx} &= \sigma_x - \sigma_m \\ s_{yy} &= \sigma_y - \sigma_m \\ s_{xy} &= \tau_{xy} \end{aligned} \quad (3.29)$$

Então substituindo as Equações (3.29) na Equação (3.25), o critério de escoamento toma uma forma mais simples:

$$\sqrt{s_{xx}^2 + s_{xy}^2} + \sigma_m \sin \phi - c \cos \phi \leq 0 \quad (3.30)$$

Com a introdução de uma variável auxiliar denominada  $z$ , esta restrição pode ser transformada como uma restrição linear acoplada com uma restrição cônica quadrática:

$$\begin{aligned} z &= k - a \sigma_m \\ \sqrt{s_{xx}^2 + s_{xy}^2} &\leq z \end{aligned} \quad (3.31)$$

onde:  $a = \sin \phi$  e  $k = c \cdot \cos \phi$

Definindo-se o vetor  $s$  como:

$$s = \begin{bmatrix} \sigma_m & s_{xx} & s_{xy} \end{bmatrix}_i^T \quad i=1 \dots n \quad (3.32)$$

onde  $n$  é o número de elementos da malha de elementos finitos. Este vetor  $s$  está relacionado com o vetor  $\sigma$  através de uma matriz  $Q$ , tal que:

$$\sigma = Q \cdot s \quad \text{onde} \quad Q = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.33)$$

Considerando a inversa da matriz  $Q$  para relacionar os componentes de  $s$  e  $\sigma$ , substituindo os valores de  $a, k, z$  no vetor  $s$  e introduzindo um novo vetor  $s_2$ , tem-se a seguinte relação:

$$s_2 = \begin{Bmatrix} z \\ s_{xx} \\ s_{xy} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} -\sin\phi & -\sin\phi & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} + \begin{Bmatrix} c \cos\phi \\ 0 \\ 0 \end{Bmatrix} \quad (3.34)$$

Finalmente introduz-se um novo vetor  $\rho$ , tal que:

$$\rho = [\rho_1 \quad \rho_2 \quad \rho_3]^T, \quad \text{para } i=1\dots n \quad (3.35)$$

$$\rho = 2.s_2$$

Então substituindo a Equação (3.35) na Equação (3.34) tem-se que:

$$\rho = D\sigma + d \quad (3.36)$$

onde:

$$D = \begin{bmatrix} -\sin\phi & -\sin\phi & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad d = \begin{Bmatrix} 2c \cos\phi \\ 0 \\ 0 \end{Bmatrix}, \quad (3.37)$$

$$\rho \in \mathcal{K}q, \quad \mathcal{K}q = \{ \rho \in \mathbb{R}^3 \mid \rho_1 \geq \sqrt{\rho_2^2 + \rho_3^2} \}$$

$\mathcal{K}q$  é o cone quadrático de segunda ordem.

As Equações (3.36) e (3.37) mostram a restrição do critério de escoamento de Mohr Coulomb, na forma cônica quadrática de segunda ordem que é usada para o algoritmo de programação cônica quadrática de segunda ordem (SOCP). A matriz de transformação  $D$  é não singular para  $\phi > 0$  (Krabbenhoft, Lyamin e Sloan, 2006).

### 3.1.5.2.

#### Critério de Drucker-Prager na forma Cônica Quadrática

O critério de Drucker-Prager pode ser utilizado como critério de escoamento para problemas em 2D e 3D. Este critério de acordo com a revisão bibliográfica tem a seguinte forma:

$$\sqrt{J_2} + \alpha \sigma_m - k \leq 0 \quad (3.38)$$

Onde  $J_2$  é o segundo invariante do tensor desviador,  $\sigma_m$  tensão média,  $\alpha$  e  $k$  são os parâmetros do material de acordo com o critério de resistência de Drucker-Prager. Em seguida mostra-se como a Equação (3.38) pode ser reescrita na forma cônica quadrática para os casos 2D e 3D:

##### a. Caso 2D

$$\sigma_m = \frac{1}{2}(\sigma_x + \sigma_y), \quad s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij} \quad (3.39)$$

$$\begin{aligned} \sigma_{xx} &= \sigma_x \\ \sigma_{yy} &= \sigma_y \\ \sigma_{xy} &= \tau_{xy} \end{aligned} \quad (3.40)$$

onde  $\delta$  é o delta de Kronecker,  $\sigma_m$  é a tensão média,  $s_{ij}$  são os componentes do tensor desviador.

Substituindo a Equação (3.40) na Equação (3.39), tem-se:

$$\begin{aligned} s_{xx} &= \sigma_x - \sigma_m \\ s_{yy} &= \sigma_y - \sigma_m \\ s_{xy} &= \tau_{xy} \end{aligned} \quad (3.41)$$

$$\text{Sabe-se que:} \quad J_2 = \frac{1}{2} s_{ij} s_{ji} \quad (3.42)$$

Então, substituindo os sub-índices na Equação (3.42), tem-se a seguinte expressão:

$$J_2 = \frac{1}{2}s_{xx}^2 + \frac{1}{2}s_{yy}^2 + s_{xy}^2 \quad (3.43)$$

Agora sabe-se das equações (3.39) e (3.41) que:

$$s_{xx} = -s_{yy} \quad (3.44)$$

Então substituindo as Equações (3.43) e (3.44) na Equação (3.38) para o caso 2D, o critério de escoamento toma uma forma mais simples:

$$\sqrt{s_{xx}^2 + s_{xy}^2} + \alpha\sigma_m - k \leq 0 \quad (3.45)$$

Com a introdução de uma variável auxiliar denominada  $z$ , esta restrição pode ser transformada numa restrição do tipo cônica quadrática:

$$\begin{aligned} z &= k - \alpha\sigma_m \\ \sqrt{s_{xx}^2 + s_{xy}^2} &\leq z \end{aligned} \quad (3.46)$$

Agora construindo o seguinte vetor denominado  $s$ :

$$s = \begin{bmatrix} \sigma_m & s_{xx} & s_{xy} \end{bmatrix}_i^T \quad \text{para } i=1\dots n \quad (3.47)$$

Onde  $n$  é o numero de elementos da malha de elementos finitos.

O vetor  $s$  está relacionado com o vetor  $\sigma$  através de uma matriz  $Q$ , tal que:

$$\sigma = Qs \quad \text{onde} \quad Q = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

Considerando a inversa da matriz  $Q$  para relacionar os componentes de  $s$  e  $\sigma$ , e introduzindo um novo vetor  $s_2$  tem-se a seguinte relação:

$$s_2 = \begin{Bmatrix} z \\ s_{xx} \\ s_{xy} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} -\alpha & -\alpha & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} + \begin{Bmatrix} k \\ 0 \\ 0 \end{Bmatrix} \quad (3.49)$$

Finalmente introduz-se uma nova variável  $\rho$ , tão que:

$$\rho = [\rho_1 \quad \rho_2 \quad \rho_3]^T, \quad \text{para } i=1\dots n \quad (3.50)$$

$$\rho = 2s_2$$

Substituindo a Equação (3.50) na Equação (3.49) tem-se que:

$$\rho = D\sigma + d \quad (3.51)$$

onde:

$$D = \begin{bmatrix} -\alpha & -\alpha & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad d = \begin{bmatrix} 2k \\ 0 \\ 0 \end{bmatrix}, \quad (3.52)$$

$$\rho \in \mathcal{K}q, \quad \mathcal{K}q = \{ \rho \in \mathbb{R}^3 \mid \rho_1 \geq \sqrt{\rho_2^2 + \rho_3^2} \}$$

$\mathcal{K}q$  é o cone quadrático de segunda ordem.

As Equações (3.51) e (3.52) mostram a restrição do critério de escoamento de Drucker-Prager na forma cônica quadrática de segunda ordem para o caso 2D, que é usada para o algoritmo de programação cônica quadrática da segunda ordem (SOCP).



**b. Caso 3D**

$$\sigma_m = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z), \quad s_{ij} = \sigma_{ij} - \sigma_m \delta_{ij} \quad (3.53)$$

$$\sigma = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z & \tau_{xy} & \tau_{xz} & \tau_{yz} \end{bmatrix}_i^T \quad \text{para } i=1 \dots n$$

$$\begin{aligned} \sigma_{xx} &= \sigma_x & \sigma_{xy} &= \tau_{xy} \\ \sigma_{yy} &= \sigma_y & \sigma_{xz} &= \tau_{xz} \\ \sigma_{zz} &= \sigma_z & \sigma_{yz} &= \tau_{yz} \end{aligned} \quad (3.54)$$

onde  $\delta$  é o delta de Kronecker,  $s_{ij}$  são os componentes do tensor desviador. Substituindo a Equação (3.54) na Equação (3.53), tem-se:

$$\begin{aligned} s_{xx} &= \sigma_x - \sigma_m & s_{xy} &= \tau_{xy} \\ s_{yy} &= \sigma_y - \sigma_m & s_{xz} &= \tau_{xz} \\ s_{zz} &= \sigma_z - \sigma_m & s_{yz} &= \tau_{yz} \end{aligned} \quad (3.55)$$

Agora sabe-se que:  $J_2 = \frac{1}{2} s_{ij} \cdot s_{ji} \quad (3.56)$

Substituindo os sub-índices na Equação (3.56), tem-se a seguinte expressão:

$$J_2 = \frac{1}{2} s_{xx}^2 + \frac{1}{2} s_{yy}^2 + \frac{1}{2} s_{zz}^2 + s_{xy}^2 + s_{xz}^2 + s_{yz}^2 \quad (3.57)$$

A Equação (3.38) pode ser reduzida de maneira a poder colocar a expressão na forma cônica quadrática da segunda ordem (Makrodimopoulos e Martin, 2005), utilizando as seguintes expressões:

$$s_{zz} = -(s_{xx} + s_{yy}) \quad (3.58)$$

$$J_2 = \|y\| \quad (3.59)$$

$$y = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} s_{xx} \\ s_{yy} \\ s_{xy} \\ s_{xz} \\ s_{yz} \end{Bmatrix} \quad (3.60)$$

Com a introdução de uma variável auxiliar denominada  $z$  e as Equações (3.58), (3.59) e (3.60), a restrição do critério de escoamento de Drucker-Prager pode ser transformada como uma restrição linear acoplada com uma restrição cônica quadrática:

$$\begin{aligned} z &= k - \alpha \sigma_m \\ \|y\| &\leq z \end{aligned} \quad (3.61)$$

Agora introduzindo o seguinte vetor denominado  $s_3$ :

$$s_3 = \begin{bmatrix} \sigma_m & s_{xx} & s_{yy} & s_{xy} & s_{xz} & s_{yz} \end{bmatrix}_i^T \quad \text{para } i=1 \dots n \quad (3.62)$$

Onde  $n$  é o numero de elementos da malha de elementos finitos.

Este vetor  $s_3$  esta relacionado com o vetor  $\sigma$  através de uma matriz  $Q$ , tal que:

$$\sigma = Q_3 s_3 \quad \text{onde } Q_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.63)$$

Considerando a inversa da matriz  $Q_3$  para relacionar os componentes de  $s_3$  e  $\sigma$ , e substituindo os valores de  $a$ ,  $k$ ,  $z$  no vetor  $s$  e introduzindo uma nova variável  $\rho$  tem-se a seguinte relação:

$$\rho = D\sigma + d \quad (3.64)$$

Onde:

$$D = \begin{bmatrix} -\alpha/3 & -\alpha/3 & -\alpha/3 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 \\ -\sqrt{3}/6 & 2\sqrt{3}/6 & -\sqrt{3}/6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad d = \begin{bmatrix} k \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (3.65)$$

$$\rho \in Kq, \quad Kq = \{ \rho \in \mathbb{R}^6 \mid \rho_1 \geq \sqrt{\rho_2^2 + \rho_3^2 + \rho_4^2 + \rho_5^2 + \rho_6^2} \}$$

$Kq$  é o cone quadrático de segunda ordem.

As Equações (3.64) e (3.65) mostram a restrição do critério de escoamento de Drucker-Prager, na forma cônica quadrática de segunda ordem para o caso 3D, que é usada para o algoritmo de programação cônica quadrática da segunda ordem (SOCP).

### 3.2. Ferramentas de Otimização

A solução do problema pela Análise Limite consiste em determinar o campo de estado de tensões correspondentes ao valor  $\lambda$  que maximiza as cargas aplicadas na estrutura, de maneira que o estado de tensões satisfaça à condição de equilíbrio e ao critério de escoamento, ou seja, seja estaticamente admissível. O problema pode ser colocado na forma do seguinte problema de programação matemática.

$$\begin{aligned} \text{Maximizar:} \quad & \lambda \\ \text{Sujeito a:} \quad & [G]\{\sigma\} = \lambda f_0 \end{aligned} \quad (3.66)$$

$$F(\sigma) < 0 \quad (3.67)$$

O problema é resolvido via o uso de algoritmos de programação matemática. Com a finalidade de desenvolver diferentes rotinas para a solução do

problema de programação matemática, foi escrito um código de programação utilizando o programa MATLAB R2009, no qual foram utilizados dois otimizadores acoplados ao programa:

### 3.2.1. Programa de Otimização Mosek

O programa de otimização Mosek, desenvolvido por Andersen et. al (2003), é um pacote comercial utilizado para problemas de otimização de grande escala. O programa Mosek pode resolver as diferentes classes de problemas de otimização, tais como problemas de otimização linear, quadrática, cônica, e mistos integrados. Com efeito, Mosek inclui os seguintes otimizadores:

- Problemas de programação linear (PL).
- Problema de programação quadrática (PQ).
- Problemas de programação cônica quadrática.
- Problemas Mistos integrados.

O otimizador Mosek tem suporte nativo para muitas linguagens diferentes de programação, tais como: C/C++, Java, MATLAB Toolbox, .NET (C#/Visual basic), Python. Dois tipos diferentes de otimizadores estão disponíveis, o padrão é um método de pontos interiores e outro o método simplex:

#### a. Otimizador de Ponto Interior

É o otimizador padrão do programa, onde se cria a partir de formulação do problema padrão de otimização (para os tipos de problemas que o programa resolve) um sistema de equações chamado “*Homogeneous Self-Dual Model*” ou modelo homogêneo, o qual é estabelecido impondo as condições de otimalidade do problema, e que é resolvido finalmente utilizando o algoritmo de otimização de pontos interiores no problema primal-dual. Este otimizador é utilizado para solução de problemas de otimização linear, cônica, quadrática.

### b. Otimizador Simplex

Usa um método diferente que permite explorar uma estimativa inicial para a solução ideal para reduzir o tempo de solução. Dependendo do problema, pode ser mais rápido ou mais lento para obter uma estimativa inicial. Quando se utiliza o otimizador simplex, pode ser possível a reutilização de uma solução existente e, conseqüentemente, reduzir o tempo de solução de forma significativa.

A forma que é de muito proveito é a forma da programação cônica de segunda ordem (Cônica Quadrática). Com a finalidade de aplicar este programa de otimização no presente trabalho se revisa a seguir a estrutura e a descrição que o programa utiliza quando realiza a otimização na forma cônica de segunda ordem.

#### 3.2.1.1. Estrutura do Problema de Otimização Cônica (Mosek)

O programa Mosek, formula o problema na seguinte forma:

$$\begin{aligned}
 \text{Minimiza} \quad & c^T x + c^f \\
 \text{Sujeito a:} \quad & l^c \leq Ax \leq u^c \\
 & l^x \leq x \leq u^x \\
 & x \in K
 \end{aligned} \tag{3.68}$$

Onde:

$l^c, u^c$ : limite inferior e superior das restrições.

$l^x, u^x$ : limite inferior e superior das variáveis.

$A$ : Matriz de coeficientes constantes.

$c^T, c^f$ : Vetores com elementos constantes.

$x$ : variáveis do problema.

$K$ : cone convexo.

Seguinte, definição:

$$\mathcal{K} = \{x \in R^n : x^t \in \kappa_t, t = 1, 2, \dots, m\}$$

Onde:

$$R \text{ é um conjunto: } \quad \kappa_t = \{x \in R^n\}$$

$\kappa_t$  é um cone na forma quadrática ou um cone rotativo.

Uma variável é um membro padrão do conjunto  $R$ , a menos que ela pertença a um cone específico.

### 3.2.1.2.

#### Descrição geral do Problema de Otimização Cônica (Mosek)

Para resolver problemas de otimização, o programa utiliza a teoria da dualidade que liga o problema primal e seu correspondente dual. Devido às fortes relações entre estes dois problemas, os métodos modernos de pontos interiores resolvem simultaneamente os dois problemas fazendo uso de informações de um para ajudar o progresso na outra.

$$\text{Primal } \begin{cases} \min_x c^T x \\ \text{s.t. : } Ax = b \\ x \in \mathcal{K} \end{cases} \quad \text{Dual } \begin{cases} \max_{y,s} b^T x \\ \text{s.t. : } A^T y + s = c \\ s \in \mathcal{K}^*, y \in R^m \end{cases} \quad (3.69)$$

Aqui  $\mathcal{K}^*$ , denota o cone dual do cone  $\mathcal{K}$ . Portanto pode-se empregar o método eficiente do pontos interiores simétrico primal-dual (Nesterov e Todd, 1997; 1998). O objetivo então é resolver o modelo homogêneo dual do problema da Equação (3.69).

Então pode-se escrever:  $z = (x, \tau, y, s, k) \in S = R_+^n \times R_+ \times R^m \times R_+^n \times R_+$  e introduzindo:

$$\begin{aligned} r_p(z) &= Ax - b\tau \\ r_d(z) &= -A^T y - s + c\tau \\ r_g(z) &= -c^T x + b^T y - k \\ \mu(z) &= (x^T s + \tau k) / (v + 1) \end{aligned} \quad (3.70)$$

Agora colocando  $z^0 = (x^0, \tau^0, y^0, s^0, k^0) \in S$  como ponto inicial, e assumindo que  $\theta$  é uma variável escalar. Tem-se o seguinte problema:

Minimizar<sub>(z, \theta)</sub>  $\theta\mu(z^0)$

Sujeito a:

$$\begin{cases} Ax - b\tau = \theta r_p(z^0) \\ -A^T y - s + c\tau = \theta r_d(z^0) \\ -c^T x + b^T y - k = \theta r_g(z^0) \\ r_p(z^0)^T y - r_d(z^0)^T x + r_g(z^0)\tau = \theta\mu(z^0) \\ (x, \tau) \geq 0, (s, k) \geq 0 \end{cases} \quad (3.71)$$

A solução para o sistema de Equações (3.71) fornece uma ótima solução para o problema de otimização cônica original. A solução é encontrada fazendo uso do algoritmo de pontos interiores (Skajaa, et. al., 2012; Andersen et. al., 2003).

### 3.2.2. Otimizador do Matlab (*fmincon*)

O programa MATLAB tem entre suas ferramentas, um comando para resolver problemas de otimização, chamado *fmincon*. Utiliza-se esse comando para encontrar a solução deste tipo de problema. Esta ferramenta determina as variáveis do problema que minimizam uma função objetivo e que satisfazem as restrições lineares e não lineares. O problema é colocado da forma da Equação (3.72)

$$\begin{cases} \min_x f(x) \\ s.t. : c(x) \leq 0 \\ ceq(x) = 0 \\ Ax \leq b \\ Aeq x = 0 \\ lb \leq x \leq ub \end{cases} \quad (3.72)$$

Onde:

$x$  : Vetor das Variáveis do problema.

$f(x)$  : Função objetivo.

$Aeq, beq$  : Matriz e vetor das restrições lineares de igualdade do problema.

$A, b$  : Matriz e vetor das restrições lineares de desigualdade do problema.

$c(x)$ ,  $ceq(x)$  : Vetores das restrições de igualdade e desigualdade não linear do problema.

$lb$ ,  $ub$  : Vetores dos limites inferiores e superiores.

Deve-se criar um arquivo com uma função escalar que representa a função objetivo, além disso, deve-se ter uma função onde armazenar as restrições de igualdade e desigualdade não linear. Com esta estrutura realizada, a ferramenta *fmincon*, faz a minimização da função objetivo, utilizando um ponto inicial, que é colocado pelo usuário. Recomenda-se que este valor inicial deve ser definido por um valor tão próximo quanto possível da solução de maneira a acelerar a convergência do algoritmo. Caso exista uma técnica de busca inteligente para um valor inicial de um dado problema, essa técnica deve ser usada.

O programa tem entre seus algoritmos para realizar o processo de otimização o algoritmo de pontos interiores. O usuário pode fazer uso da forma esparsa das matrizes  $A$  e  $Aeq$  com a finalidade de reduzir o tempo e a memória computacional durante o processo de otimização.

### 3.2.3.

#### **Aplicação dos otimizadores ao problema de Análise Limite**

Para a solução do problema de análise de estabilidade via análise limite utilizando a programação matemática, se faz uso dos otimizadores do programa comercial Mosek e o próprio otimizador do Matlab (*fmincon*). Com a finalidade de ver a eficiência do algoritmo com as diferentes formas de colocar a função de escoamento, o problema é analisado de diferentes formas.

#### 3.2.3.1.

##### **Problema de Análise Limite na forma convencional**

Neste caso se trabalha com o sistema de equações que fica como produto da aplicação do teorema de trabalho virtual, e que é apresentando nas equações.



Neste problema as restrições de desigualdade estão dadas pelas funções de escoamento que dependem do tipo de critério de resistência. Este sistema é como seguir:

$$\begin{aligned} \text{Maximizar:} \quad & \lambda \\ \text{Sujeito a:} \quad & [G]\{\sigma\} = \lambda f_0 \end{aligned} \quad (3.73)$$

$$F(\sigma) < 0 \quad (3.74)$$

Este sistema é equivalente ao seguinte:

$$\begin{aligned} \text{Minimizar:} \quad & -\lambda \\ \text{Sujeito a:} \quad & [G]\{\sigma\} = \lambda f_0 \end{aligned} \quad (3.75)$$

$$F(\sigma) < 0 \quad (3.76)$$

Como pode-se ver este problema corresponde a um problema de otimização na forma padrão, no qual se tem restrições lineares, e as restrições tornam-se não lineares, constituindo portanto um problema de programação não linear.

Este problema é analisado utilizando-se a ferramenta de otimização do programa Matlab (*fmincon*), o qual pode resolver este tipo de problemas.

### 3.2.3.2.

#### Problema de Análise Limite na forma cônica quadrática.

Neste caso deve-se realizar uma mudança na forma convencional do problema, de forma que as restrições de desigualdade dadas pelas funções de escoamento do critério de resistência fiquem na forma cônica quadrática. Com essa finalidade se utiliza as transformações realizadas na seção 3.1.5, onde em todos os casos se tem que a transformação das variáveis de tensões do problema se relaciona com as variáveis de uma função cônica da seguinte forma:

$$\rho = D\sigma + d \quad (3.77)$$

Dessa maneira a restrição de desigualdade que é mostrada na Equação (3.76) fica na forma cônica quadrática, onde a função de escoamento é agora função de uma nova variável chamada  $\rho$ . Então substituindo a Equação (3.77) nas Equações (3.75 e 3.76), tem-se o seguinte:

Minimizar:  $-\lambda$

$$\text{Sujeito a: } [G][D]^{-1}\{\rho\} = \lambda f_0 + [G][D]^{-1}d \quad (3.78)$$

$$\rho \in \kappa \quad (3.79)$$

Onde  $\kappa$  é o cone quadrático.

Este problema está colocado na forma cônica quadrática, o qual pode ser resolvido utilizando o otimizador do programa comercial Mosek transformando o problema utilizando a equação (3.78) e utilizando as novas variáveis. Este problema é também analisado com o uso do otimizador do Matlab (*fmincon*), tendo que fazer uma transformação na função de armazenamento das restrições não lineares de maneira que fique como a Equação (3.79) e das equações de equilíbrio utilizando a Equação (3.78).

### 3.2.3.3.

#### Estimativa do valor inicial.

Como já mencionado, o algoritmo de otimização melhora seu desempenho quando uma boa estimativa é feita para o vetor inicial das variáveis. Uma boa estimativa para as tensões como valores iniciais é obtida com a solução da análise linear elástica da estrutura. Essa opção tem se mostrado como um ponto inicial adequado para conseguir uma boa convergência do problema numérico. Para se obter essa solução, resolve-se o problema como indicado a seguir.

- **Caso do Estado Plano de Deformações (2D)**

Utilizando o método de elementos finitos (MEF) para estruturas de comportamento linear elástico, pode-se obter a matriz de rigidez de um elemento como:

$$[K_e] = \int_{A^e} [B_e]^T [C_e] [B_e] dA^e \quad (3.80)$$

Onde:

$B_e$  : Matriz de relação da deformação e deslocamento do elemento  $e$ .

$K_e$  : Matriz de rigidez do elemento  $e$ .

$A^e$  : Área do elemento  $e$ .

$C_e$  : Matriz constitutiva para o estado plano de deformações no elemento  $e$ .

Esta matriz constitutiva é escrita da seguinte forma no caso de estado plano de deformações:

$$[C_e] = \frac{E_e}{(1 + \nu_e)(1 - 2\nu_e)} \begin{bmatrix} 1 - \nu_e & \nu_e & 0 \\ \nu_e & 1 - \nu_e & 0 \\ 0 & 0 & \frac{1 - 2\nu_e}{2} \end{bmatrix} \quad (3.81)$$

Onde:

$E_e$ : módulo de Young ou modulo elástico do material do elemento  $e$ .

$\nu_e$ : módulo de Poisson do material do elemento  $e$ .

A integração da matriz definida na Equação (3.81), é feita, como já foi mostrado, utilizando a integração nos pontos de Gauss

$$[K_e] = \sum_1^{ng} [B_e(r_i, s_i)]^T [C_e] [B_e(r_i, s_i)] \det(J_e(r_i, s_i)) W_{r_i} W_{s_i} \quad (3.82)$$

Onde:

$r_i, s_i$ : Pontos de Gauss do elemento paramétrico.

$W_{r_i}, W_{s_i}$ : Fatores de ponderação.

$J_e$ : Matriz Jacobiana do elemento  $e$ .

a matriz de rigidez na forma global  $K$  é obtida a partir da montagem das matrizes de rigidez de cada elemento.

De acordo com o Método dos Elementos Finitos os deslocamentos nodais são calculados da seguinte forma:

$$\{d\} = [K]^{-1} \{fo\} \quad (3.83)$$

onde:

$d$ : Vetor dos deslocamentos a nível global

$fo$ : Vetor das forças nodais.

$K$ : Matriz de rigidez a nível global.

Finalmente são calculadas as deformações e as tensões no elemento  $e$  utilizando as seguintes equações:

$$\{\varepsilon_e(r, s)\} = [B_e(r, s)]\{d_e\} \quad (3.84)$$

$$\{\sigma_e(r, s)\} = [C_e]\{\varepsilon_e(r, s)\} \quad (3.85)$$

Onde  $d_e$  são os deslocamentos nodais do elemento  $e$  extraídos do vetor dos deslocamentos globais  $d$ .

Os pontos onde se devem avaliar as deformações e as tensões são nas coordenadas paramétricas  $r = 0, s = 0$ . Já que os elementos são considerados de tensão constante, então as tensões são colocadas no centroide do elemento. A inicialização do valor do fator de colapso para os casos estudados foi de  $\lambda = 1$ , não tendo este valor muita influência no resultado final.

- **Caso do Análise Tridimensional (3D)**

Para o caso da análise tridimensional o ponto inicial é calculado da mesma forma do que no caso de estado plano de deformações, a diferença aqui é a matriz constitutiva deve mudar para o caso tridimensional da seguinte forma:

$$C_e = \frac{E_e}{(1 + \nu_e)(1 - 2\nu_e)} \begin{bmatrix} 1 - \nu_e & \nu_e & 0 \\ \nu_e & 1 - \nu_e & 0 \\ 0 & 0 & \frac{1 - 2\nu_e}{2} \end{bmatrix} \quad (3.86)$$

onde:

$E_e$ : módulo de Young ou módulo elástico do material do elemento  $e$ .

$\nu_e$ : módulo de Poisson do material do elemento  $e$ .

A integração da matriz da matriz de rigidez do elemento é feita utilizando a integração nos pontos de Gauss.

$$[K_e] = \sum_1^{ng} [B_e(r_i, s_i, t_i)]^T [C_e] [B_e(r_i, s_i, t_i)] \det(J_e(r_i, s_i, t_i)) W_{r_i} W_{s_i} W_{t_i} \quad (3.87)$$

onde:

$r_i, s_i, t_i$ : Pontos de Gauss do elemento paramétrico.

$W_{r_i}, W_{s_i}, W_{t_i}$ : Fatores de ponderação.

$J_e$ : Matriz Jacobiana do elemento  $e$ .

Baseado na teoria da elasticidade são calculados os deslocamentos, as deformações e as tensões de forma similar do que no caso 2D.

### 3.3. Cálculo do Fator de Segurança

Em 1975, Zienkiewicz propõe que o fator de segurança ( $FS$ ) usado nas aplicações geotécnicas pode ser calculado pelo incremento das cargas externas ou pela redução da resistência através do método de elementos finitos (MEF).

No presente trabalho utiliza-se a técnica da redução da resistência para encontrar o fator de segurança ( $FS$ ), no qual se aplica um fator de redução ( $FR$ ), aos parâmetros de resistência do solo:

$$c_{red} = \frac{c}{FR} \quad (3.88)$$

$$\phi_{red} = \arctan\left(\frac{\tan \phi}{FR}\right) \quad (3.89)$$

onde:

$c$ : Coesão do material.

$\phi$ : Ângulo de atrito do material.

$c_{red}$ : Coesão reduzida.

$\phi_{red}$ : Ângulo de atrito reduzido.

$FR$ : Fator de redução.

No método de Análise Limite, o fator de segurança ( $FS$ ) é obtido utilizando a técnica da redução da resistência, no qual o fator de colapso ( $\lambda$ ) é obtido via programação matemática para diferentes valores do fator de redução ( $FR$ ), e o fator de segurança é aquele fator de redução que corresponde ao fator de colapso

( $\lambda=1.0$ ). A Figura 3.3, mostra a variação do fator de colapso ( $\lambda$ ) com o fator de redução da resistência ( $FR$ ), onde o fator de segurança ( $FS$ ) é o fator de redução para o qual o fator de colapso é um ( $\lambda=1.0$ ).

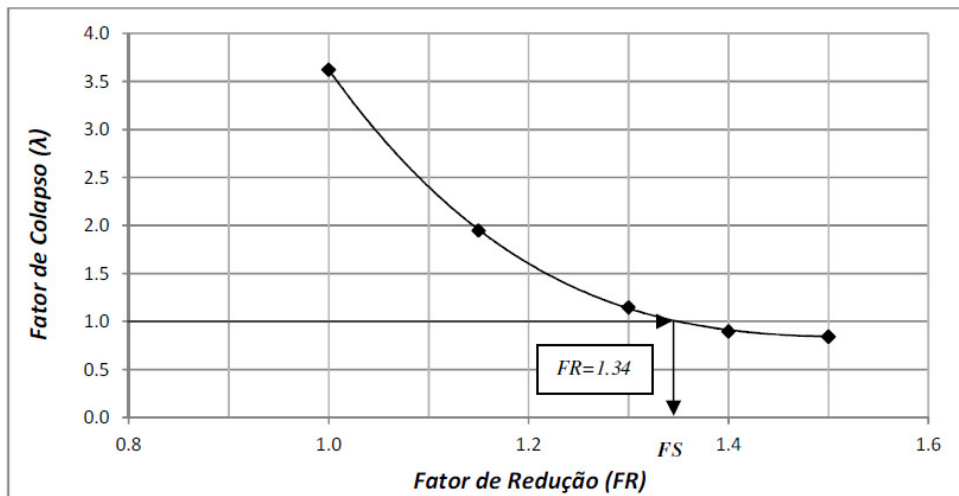


Figura 3.3 - Variação do fator de colapso ( $\lambda$ ) com o fator de redução ( $FR$ ).

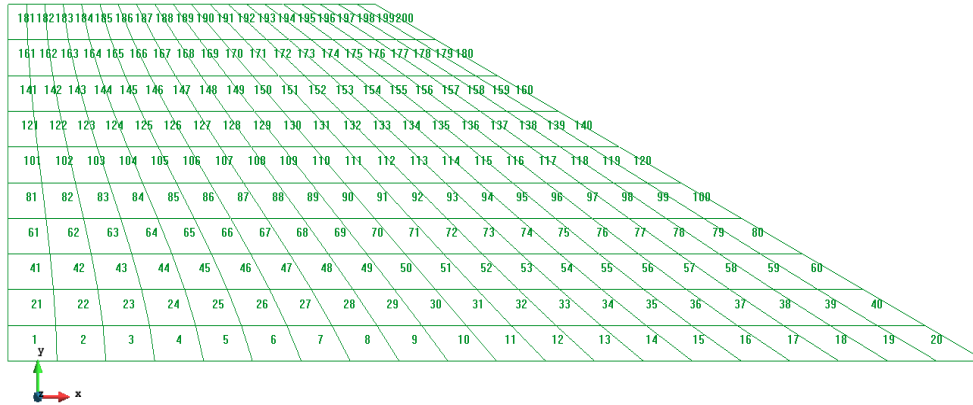
### 3.4. Implementação Numérica

Baseado na metodologia explicada foi desenvolvido uma ferramenta numérica no programa Matlab versão R2009b e em linguagem C. Este código foi feito num computador com processador Intel Core i3 de 2.13 GHz de velocidade e memória RAM 3.00 GB. Para execução do programa criou-se arquivos de entrada no formato *Neutral File* (.nf), estes arquivos contém:

- Coordenadas nodais.
- Matriz de incidência nodal.
- Nós com campos de velocidade prescritos (Condições de contorno).
- Propriedades dos materiais e outros parâmetros necessários para a simulação.

Estes arquivos são gerados depois de fazer a discretização da estrutura geotécnica. Para a discretização da estrutura, foram utilizados geradores de malha. Os programas utilizados para gerar a malha foram o GID versão 11.0.2 e o Mtool 5.1 desenvolvido na TeCGraf-PUC-Rio. Os elementos utilizados para a geração

da malha foram: elemento quadrilateral de 4 nós (Q4) em aplicações 2D e o elemento hexaédrico de 8 nós (BRICK8) em aplicações 3D.



**Figura 3.4 - Discretização feita com gerador de malha.**

Com estes arquivos criados é feita a leitura de dados para o problema. Em seguida, se faz uso do método dos elementos finitos utilizando o programa Matlab para obter as informações necessárias para a formulação do problema.

- Vetores de força, a matriz de rigidez, a matriz de relação de relação deformação-deslocamento, vetor de forças nodais, etc.
- Montagem das matrizes e vetores a nível global.
- Imposição das condições de contorno.
- Geração do sistema de equações utilizando a condição de equilíbrio e desigualdades utilizando as condições de escoamento.

Uma vez montado o sistema a ser resolvido utilizam-se os otimizadores MOSEK e do próprio Matlab (*fmincon*) para obter a solução do problema de análise limite primal e do correspondente problema dual. No trabalho estes otimizadores foram testados utilizando a forma de programação não linear convencional e a programação cônica, da seguinte maneira:

- Otimizador do Matlab (*fmincon*), com as condições do problema na forma da programação convencional não linear, ou seja, utilizando as Equações (3.75) e (3.76), e utilizando o processo de minimização. Neste caso foi

necessário criar dois arquivos, um deles para o armazenamento da função objetivo, e o segundo para o armazenamento das restrições de desigualdade (critérios de escoamento).

- Otimizador do Matlab (*fmincon*), com as condições do problema na forma da programação cônica quadrática, ou seja, utilizando as Equações (3.78) e (3.79), e utilizando o processo de minimização. Neste caso foi necessário criar dois arquivos, um deles para o armazenamento da função objetivo, e o segundo para o armazenamento das restrições de desigualdade (critérios de escoamento).
- Otimizador do programa comercial Mosek, com as condições do problema na forma da programação cônica quadrática, ou seja, utilizando as Equações (3.78) e (3.79), e utilizando o processo de minimização. Neste caso a rotina de otimização foi implementada no código principal do programa.

Deve-se ressaltar que uma rotina externa foi implementada para se obter o valor do ponto inicial das variáveis do problema conforme a metodologia descrita na seção 3.2.3.3.

O cálculo do fator de segurança (*FS*) é realizado construindo-se a curva que relaciona o fator de colapso obtido ao final do problema com o fator de redução imposto, de acordo com a metodologia descrita na seção 3.3.