

3

GEOFLUX3D - Implementação computacional

3.1.

Considerações gerais

Um dos principais objetivos desta dissertação foi o desenvolvimento de um programa computacional para solução de problemas de fluxo através de meios porosos saturados e não-saturados em domínios tridimensionais. Para esta finalidade, foi escrito em linguagem Fortran e com base no método dos elementos finitos o programa computacional GEOFLUX3D.

O programa foi desenvolvido a partir do programa GEOFLUX (Machado Jr., 2000) para análise de fluxo através de meios porosos não-saturados em análises bidimensionais. Foram efetuadas as adaptações necessárias para a resolução de problemas tridimensionais, mantendo-se a mesma estrutura de macro-comandos do programa original.

Nesta nova versão, adotaram-se elementos triangulares de 3 nós (TRI3) para análises bidimensionais e elementos tetraédricos de 4 nós (TETR4) para estudos tridimensionais. A razão da escolha destes elementos foi a facilidade na construção de malhas, que se adaptam bastante bem a domínios com contornos irregulares, bem como por não apresentarem necessidade da execução de integrações numéricas, diminuindo assim o tempo de processamento, mas exigindo malhas com maior refinamento de discretização.

Cabe ressaltar também que no programa GEOFLUX3D foram implementadas novas subrotinas para a variação das condições de contorno no tempo, tanto em termos das variáveis primárias (cargas hidráulicas) quanto secundárias (velocidades).

Para a solução da não linearidade foram empregados os métodos de Picard e de Picard modificado utilizando para esta última a formulação mista, é dizer a equação de Richards baseada na carga de pressão e no teor de umidade volumétrico, diminuindo desta forma erros por balanço de massa como apontado antes por Celia et al., 1991.

Para a geração das malhas de elementos finitos trabalhou-se com o pré-processador do programa GID v.8.0, desenvolvido pelo Centro Internacional de Métodos Numéricos em Engenharia (CIMNE) com sede em Barcelona – Espanha. Optou-se pelo GID devido à facilidade de se operar com distintos tipos de materiais e elementos, assim como devido à flexibilidade na prescrição das condições de contorno em geometrias 3D complexas. Os arquivos contendo as informações das malhas possuem extensão *.flavia.msh e têm um formato de saída muito parecido com os arquivos do tipo “neutral file”.

O código do GEOFLUX3D foi desenvolvido utilizando-se a versão 6.6 do compilador Compaq Visual FORTRAN Professional. O programa não exige recursos computacionais especiais porém, dependendo do problema a ser simulado, é fundamental que se trabalhe com uma máquina atualizada e com boa capacidade de memória e velocidade de processamento. Para a realização das simulações descritas no Capítulo 5 foram utilizados desde computadores com capacidade de processamento de 1GHz e 1GB de memória RAM a computadores com capacidade de processamento de 3,40 GHz e 3,25GB de memória RAM. Os tempos de processamento, em termos gerais, foram bastante rápidos, alcançando em malhas de grandes tamanhos menos de um minuto por iteração.

Para a visualização dos resultados utilizou-se também o pós-processador do GID v.8.0. Os arquivos contendo as informações dos resultados do GEOFLUX3D têm a extensão *.flavia.res, sendo um formato de entrada muito fácil de implementar tanto em relação a grandezas escalares (cargas hidráulicas) quanto vetoriais (velocidades e gradientes).

3.2. O programa GEOFLUX3D

3.2.1. Macro-comandos

O programa é capaz de interpretar macro-comandos utilizados para acionar um conjunto de sub-rotinas responsáveis por uma tarefa computacional específica. O programa GEOFLUX3D possui oito macro-comandos principais, descritos a seguir de forma sintética:

MANAGER: Macro-comando responsável pelo gerenciamento de todos os outros macro comandos.

DADOS: responsável pela entrada de dados, ativando as subrotinas que efetuam a leitura dos dados gerais, da geometria do problema.

MATMSH: responsável pela leitura das distribuições e propriedades dos materiais do modelo.

DATBOU: macro-comando que realiza a leitura das condições iniciais, a partir de um arquivo-texto independente que contem valores iniciais de carga de pressão ou de carga hidráulica para todos os nós da malha de elementos finitos.

DIRICHLET: controla a leitura das condições de contorno primárias (também chamadas de Dirichlet), as quais podem ser admitidas constantes ou variáveis no tempo.

NEWMAN: controla a leitura dos valores de fluxo normal prescrito no contorno (condições de contorno de Newman) e também calcula a correspondente vazão nodal equivalente se a vazão for distribuída.

SOLVE_T: ativa a subrotina COEFS para a solução do sistema de equações do problema e apresentação dos resultados assim calculados.

FEXEC: responsável pela finalização do programa.

3.2.2. Fluxograma básico

A figura 3.1 mostra um fluxograma básico do programa GEOFLUX3D desenvolvido nesta dissertação:

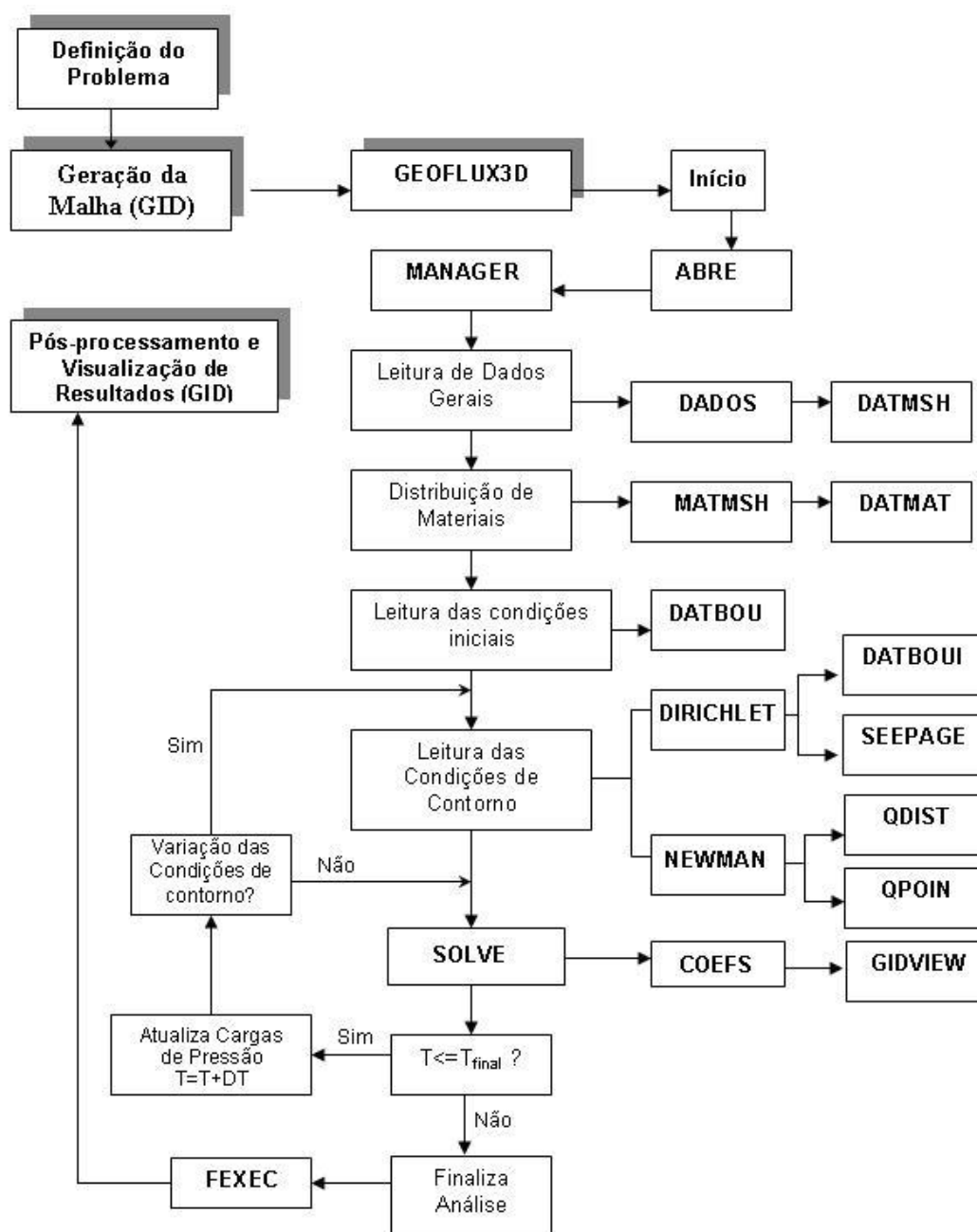


Figura 3.1 – Fluxograma básico do programa GEOFLUX3D.

Apresentam-se a seguir as características das principais subrotinas empregadas pelo programa:

ABRE: responsável pela abertura dos arquivos do tipo texto de entrada (*.D).

DADOS: responsável pela leitura dos dados gerais via arquivo texto (*.D);

DATMSH: responsável pela leitura dos dados geométricos (coordenadas dos nós e conectividades dos elementos) via arquivo texto (*.flavia.msh);

DATMAT: responsável pela leitura de dados e distribuição dos materiais no domínio do problema via arquivo texto (*.D);

DATBOU: responsável pela leitura das condições iniciais via arquivo texto (*.INI) contendo a distribuição das cargas de pressão ou das cargas hidráulicas totais no tempo $t = 0$;

DATBOUI: responsável pela leitura das condições de contorno, via arquivo texto (*.D), em termos de cargas de pressão prescritas, cujos valores podem ou não variar com o tempo. Esta subrotina foi modificada em relação à versão inicial do GEOFLUX e será discutida em detalhe no próximo item desta seção;

SEEPAGE: responsável pela leitura, via arquivo texto (*.D), dos nós inicialmente sem condição de contorno prescrita, mas que com o tempo e sob determinadas circunstâncias, podem sofrer a imposição de cargas de pressão prescritas (valores nulos) com o propósito de definir uma superfície de fluxo livre em problemas não-confinados. Esta nova subrotina implementada no GEOFLUX3D será também apresentada em detalhe no item 3.2.3;

Q_DIST: responsável pela leitura das condições de contorno em termos de fluxo normal prescrito, via arquivo texto (*.D), e pelo cálculo do vetor de vazão nodal equivalente, cujos valores podem ou não variar com o tempo;

Q_POIN: responsável pela leitura da vazão nodal prescrita via arquivo texto (*.D);

CALCOEFS: responsável pelo gerenciamento do processo de solução no tempo e no espaço, considerando iterações para obtenção da solução não-linear. Esta subrotina é considerada a principal do programa e será discutida com suas respectivas modificações em detalhe no item 3.2.4;

GIDVIEW: responsável pela impressão dos resultados: cargas nodais, gradientes e velocidades, no formato exigido para o pós-processamento através do utilitário GID (*.flavia.res);

FEXEC: responsável pela finalização da análise.

3.2.3. Comando DATBOUI

O comando DATBOUI define as condições de contorno em termos da variável primária (carga hidráulica). Na versão original do GEOFLUX estas condições eram consideradas fixas durante a simulação. O GEOFLUX3D oferece a alternativa de mudar essas condições de contorno no tempo. Os nós pertencentes ao contorno, com suas respectivas cargas hidráulicas iniciais e a função que define sua variação no tempo, tem que ser listados na utilização deste comando.

Esta mudança das condições de contorno primárias pode ser aplicada, por exemplo, quando se deseja simular o enchimento ou o rebaixamento rápido do reservatório de uma barragem. Neste caso, os nós com condição de contorno variável são aqueles pertencentes à face do talude da montante, como mostrado na figura 3.3, onde as condições de contorno variam no tempo, de acordo com a posição do nível de água no reservatório.



Figura 3.2 – Exemplo de aplicação da variação de condição de contorno primária.

3.2.4. Comando SEEPAGE

O comando SEEPAGE refere-se a valores de contorno que variam de acordo com as condições de fluxo durante a simulação. Devem ser listados os nós que definem as superfícies através das quais a água possa fluir para o exterior do domínio do problema.

Esta condição SEEPAGE pode ser acionada, por exemplo, quando ocorre fluxo através do talude de jusante de uma barragem, onde a posição final da linha freática é desconhecida (figura 3.3). Nesta figura, admitem-se conhecidas as cargas hidráulicas no talude da montante (4,5 m) e no talude da jusante (1,0 m), procurando-se então determinar a posição da linha freática em um determinado instante de tempo com auxílio da condição de contorno SEEPAGE aplicada na face de jusante.

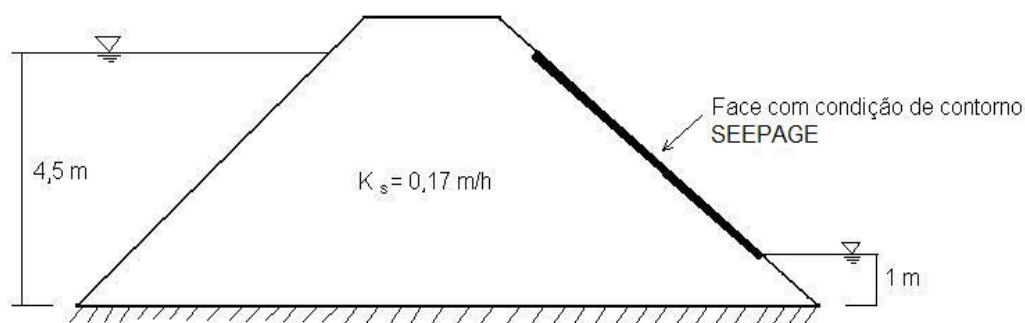


Figura 3.3 – Exemplo de aplicação da condição de contorno de fluxo livre SEEPAGE.

Nos segmentos de superfície sob a condição SEEPAGE, o programa GEOFLUX3D assume uma carga de pressão nula quando tais segmentos estão na condição saturada (SEEPAGE ATIVO) ou um fluxo normal nulo quando estes estão na condição não-saturada (SEEPAGE INATIVO).

Na figura 3.4 são mostrados os nós que formam os segmentos de superfície considerados sob condições de contorno SEEPAGE ATIVO / INATIVO, assim como as linhas equipotenciais na condição de fluxo permanente no caso da barragem mostrada na figura 3.3.

Como o número de segmentos de superfície sob condição SEEPAGE ATIVO sofre variação durante a simulação transiente do problema de fluxo, o processo de cálculo é necessariamente de natureza não-linear.

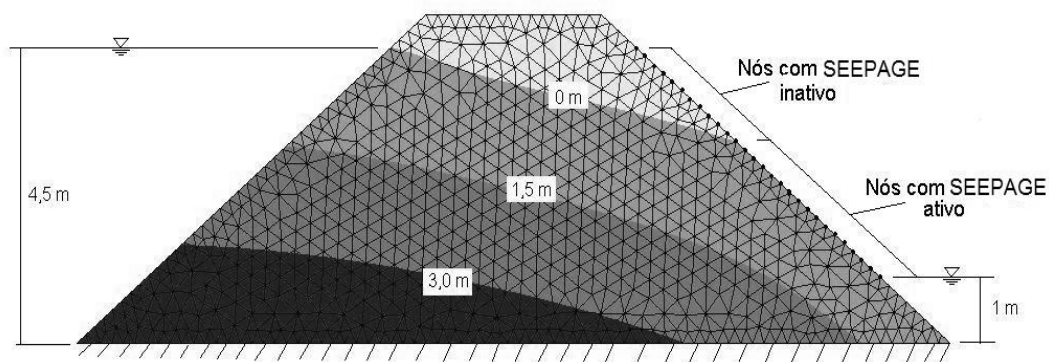


Figura 3.4 – Resultados do GEOFLUX3D aplicando condição de contorno SEEPAGE.

3.2.5. Comando CALCOEFS

O comando CALCOEFS esquematizado na figura 3.5 é responsável pela resolução do problema de fluxo, no espaço e no tempo, e pelo tratamento da não-linearidade do processo utilizando as seguintes subrotinas:

PSIO: Constrói o vetor global de cargas de pressão ψ_0 iniciais;

COEFS: Monta os termos da equação (2.32), iniciando um processo iterativo de acordo com o método de Picard atualizando as variáveis primárias (cargas de poropressão) até atingir um critério de convergência. Maiores detalhes desta subrotina nos itens 3.5, 3.6 e 3.7;

SOLVE: Resolve o sistema representado pela equação matricial (2.32), iniciando com um esquema de armazenamento espacial da matriz esparsa e em seguida resolvendo pelo método de gradiente bi-conjugado. Maiores detalhes desta subrotina no item 3.8;

RESULT: Controla a saída dos resultados para o pós-processamento;

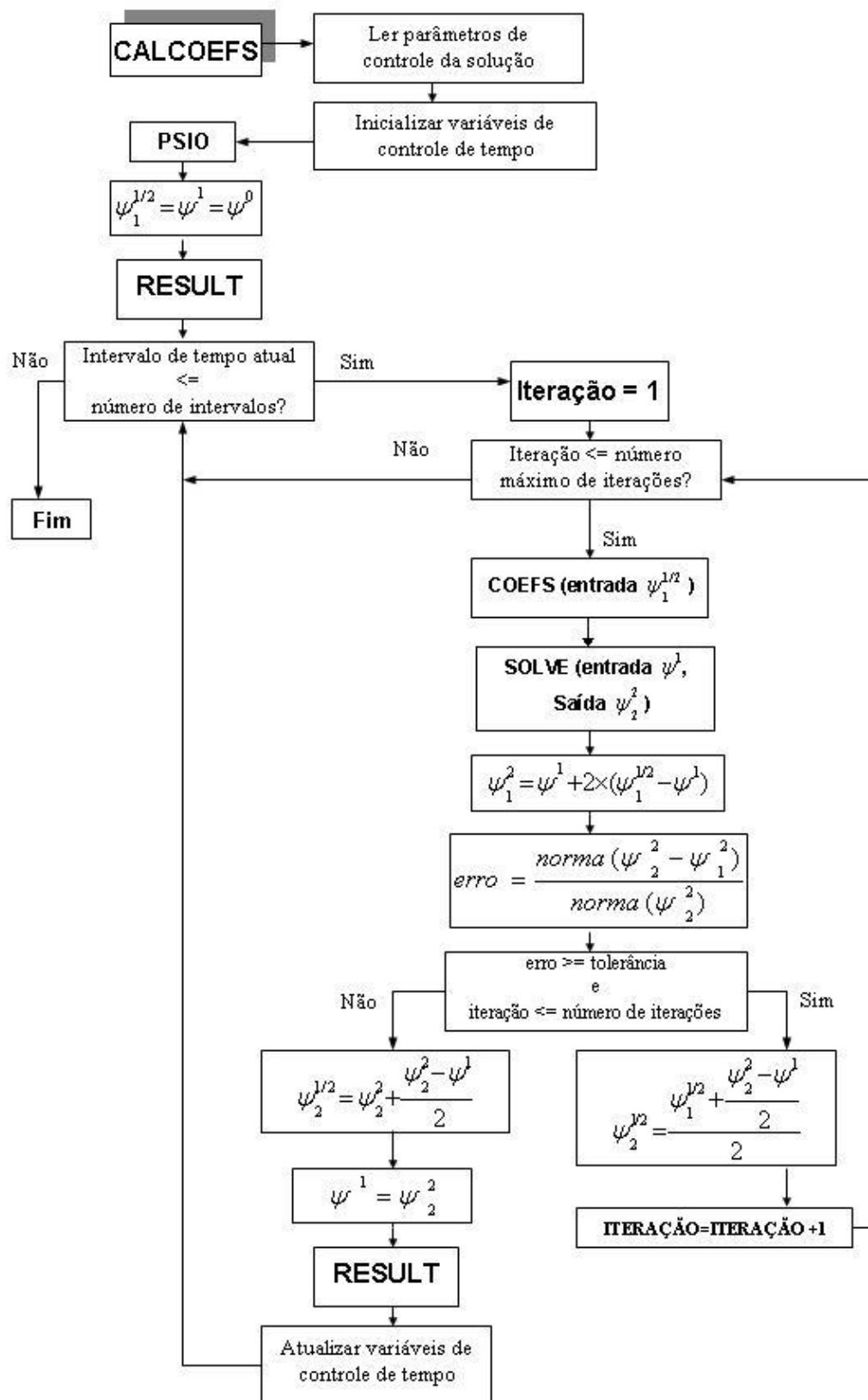


Figura 3.5 – Fluxograma básico do comando CALCOEFS.

3.3. Discretização no espaço

3.3.1. Elemento TRI3

O elemento TRI3, de aproximação linear para as cargas de pressão, é utilizado nas análises bidimensionais. Como mencionado anteriormente, optou-se por este tipo de elemento por ser de fácil implementação numérica e extrema flexibilidade de adaptação na construção de qualquer malha de elementos finitos 2D.

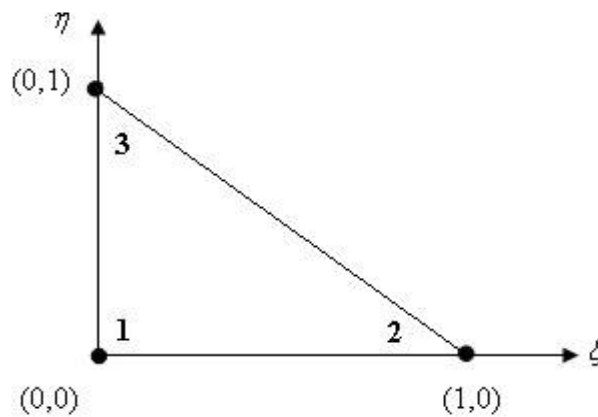


Figura 3.6 – Elemento TRI3.

O elemento TRI3 é ilustrado na figura 3.3 e suas respectivas funções de interpolação (N_i), em relação ao sistema de coordenadas locais (ξ, η) , são apresentadas nas expressões abaixo:

$$\begin{aligned} N_1(\xi, \eta) &= 1 - \xi - \eta \\ N_2(\xi, \eta) &= \xi \\ N_3(\xi, \eta) &= \eta \end{aligned} \quad (3.1)$$

3.3.2. Elemento TETR4

O elemento TETR4, também de aproximação linear da variável primária (cargas de pressão), é utilizado nas análises de fluxo tridimensionais.

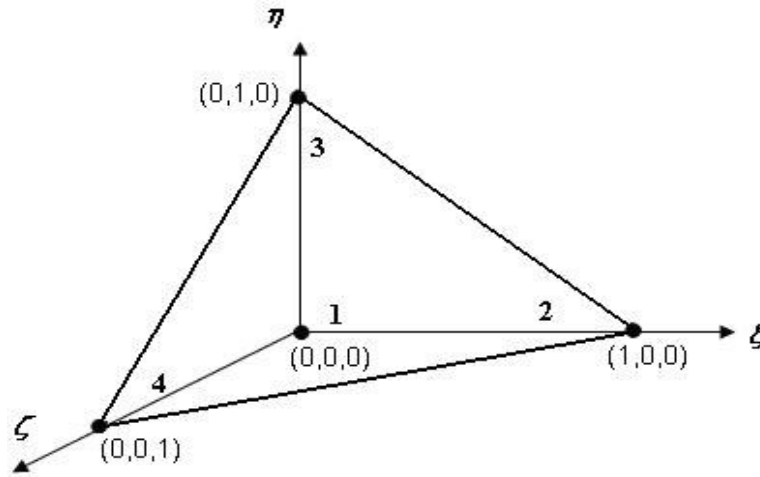


Figura 3.7 – Elemento TETR4.

O elemento TETR4 é ilustrado na figura 3.4 e suas respectivas funções de interpolação (N_i), em relação ao sistema de coordenadas locais (ξ, η, ζ) , são expressas na equação (3.2); :

$$\begin{aligned}
 N_1(\xi, \eta, \zeta) &= 1 - \xi - \eta - \zeta \\
 N_2(\xi, \eta, \zeta) &= \xi \\
 N_3(\xi, \eta, \zeta) &= \eta \\
 N_4(\xi, \eta, \zeta) &= \zeta
 \end{aligned} \tag{3.2}$$

3.4. Discretização no tempo

Reescrevendo-se a equação (2.32) com o termo dependente do tempo à esquerda do sinal de igualdade,

$$\overline{F} \cdot \frac{\partial \overline{\psi}}{\partial t} = -\overline{H} \cdot \overline{\psi} + \overline{Q} - \overline{Q}' \tag{3.3}$$

Assumindo-se que nessa equação $\overline{\psi} = \overline{\psi}(t)$ e integrando-se no elemento,

$$\int_{\psi} \overline{F} \cdot d\overline{\psi} = \int_t [-\overline{H} \cdot \overline{\psi} + \overline{Q} - \overline{Q}'] dt \tag{3.4}$$

Aplicando-se o método das diferenças finitas, onde o tempo total é subdividido em intervalos de tempo $\Delta t = t_{n+1} - t_n$, e supondo ainda uma variação linear das matrizes \bar{F} , \bar{H} , \bar{Q} e \bar{Q}' dentro desse intervalo Δt , pode-se então adotar as seguintes aproximações para cada um dos termos na equação (3.4). (Hageman e Young, 1981)

$$\int_{\psi_n}^{\psi_{n+1}} \bar{F} \cdot d\psi = \bar{F} \cdot (\bar{\psi}^{n+1} - \bar{\psi}^n) \quad (3.5)$$

$$\int_{t_n}^{t_{n+1}} \bar{H} \cdot \bar{\psi} \cdot dt = \Delta t \cdot [\alpha \cdot \bar{H}^{n+1} \cdot \bar{\psi}^{n+1} + (1-\alpha) \cdot \bar{H}^n \cdot \bar{\psi}^n] \quad (3.6)$$

$$\int_{t_n}^{t_{n+1}} (\bar{Q} - \bar{Q}') \cdot dt = \Delta t \cdot [\alpha \cdot (\bar{Q}^{n+1} - \bar{Q}^{m+1}) + (1-\alpha) \cdot (\bar{Q}^n - \bar{Q}^m)] \quad (3.7)$$

Introduzindo-se na equação (3.4) as expressões (3.5), (3.6) e (3.7), resulta a seguinte equação após rearranjo dos termos:

$$\left(\frac{\bar{F}}{\Delta t} + \alpha \cdot \bar{H}^{n+1}\right) \bar{\psi}^{n+1} = \left[\frac{\bar{F}}{\Delta t} - (1-\alpha) \bar{H}^n\right] \bar{\psi}^n + \alpha (\bar{Q}^{n+1} - \bar{Q}^{m+1}) + (1-\alpha) (\bar{Q}^n - \bar{Q}^m) \quad (3.8)$$

onde n representa o passo de tempo anterior (resultados conhecidos) e $n+1$ o passo de tempo corrente (variável primária desconhecida). O coeficiente α é aquele que define o tipo de algoritmo no tempo, podendo variar entre os valores 0 a 1. Para $\alpha = 0$ tem-se um esquema explícito, para $\alpha = 0.5$ resulta no esquema de Crank-Nicolson e para $\alpha = 1$ tem-se o esquema puramente implícito (diferenças finitas descendentes).

3.5. Método de Picard modificado

De acordo com Paniconi (1991) a equação de Richards é altamente não-linear devido às características das funções de condutividade hidráulica e da

capacidade de retenção específica, que por sua vez são dependentes das cargas de pressão e dos teores de umidade volumétricos.

Segundo Neuman (1973), o esquema puramente implícito ($\alpha = 1$) é o que melhor se aplica a problemas que possuem, no seu domínio, fluxo em condições não-saturada e saturada. Adotando esse critério na equação 3.8; chega-se à seguinte equação:

$$\left(\frac{\bar{F}}{\Delta t} + \bar{H}^{n+1}\right)\bar{\psi}^{n+1} = \frac{\bar{F}}{\Delta t} \cdot \bar{\psi}^n + \bar{Q}^{n+1} - \bar{Q}^{n+1} \quad (3.9)$$

para cuja solução são utilizados métodos iterativos, dentre os quais os mais conhecidos são os métodos de Newton-Raphson e o de Picard.

O método de Picard, conhecido também como método das aproximações sucessivas, tem entre suas principais vantagens a facilidade de implementação (bem mais simples do que o método de Newton-Raphson), a manutenção da simetria do sistema de equações e um menor custo computacional para cada iteração. Porém, pode apresentar problemas de convergência em problemas altamente não lineares, conforme reportado por Paniconi et al., 1991.

O método de Picard é definido de acordo com o seguinte algoritmo baseado na equação (3.9), onde m representa à iteração anterior e $m+1$ à iteração corrente.

$$\left(\frac{\bar{F}^{n+1,m}}{\Delta t} + \bar{H}^{n+1,m}\right)\bar{\psi}^{n+1,m+1} = \frac{\bar{F}^{n+1,m}}{\Delta t} \cdot \bar{\psi}^n + \bar{Q}^{n+1,m} - \bar{Q}^{n+1,m} \quad (3.10)$$

Célia et al. (1990) introduziram o método de Picard Modificado, escrito de forma mista e considerando, na equação de fluxo, termos de carga de pressão e de teor de umidade volumétrico, já que a solução numérica baseada na equação de Richards, expressa somente em termos de carga de pressão, geralmente produz resultados insatisfatórios caracterizados por erros na conservação de massa. A formulação mista, por outro lado, minimiza os erros no balanço de massa, ainda que este fato, por si só, não seja suficiente para garantir a obtenção de uma boa solução numérica.

Adotou-se no desenvolvimento do GEOFLUX3D o método de Picard Modificado, cuja formulação é similar ao algoritmo da equação (3.10), acrescentada de mais um termo dependente do teor de umidade volumétrico no lado direito, precisamente aquele que procura minimizar os erros de conservação de massa, como mostrado a seguir:

$$\left(\frac{\bar{F}^{n+1,m}}{\Delta t} + \bar{H}^{n+1,m}\right)\bar{\psi}^{n+1,m+1} = \frac{\bar{F}^{n+1,m}}{\Delta t} \cdot \bar{\psi}^{n+1,m} + \bar{Q}^{n+1,m} - \bar{Q}^{n+1,m} + \bar{F}_\theta \cdot \frac{\bar{\theta}^{n+1,m} - \bar{\theta}^n}{\Delta t} \quad (3.11)$$

em que

$$\bar{F}_\theta = \int_{\Omega_e} N_i \cdot d\Omega_e. \quad (3.12)$$

Como pode ser observado na equação (3.11), diferentemente do método de Picard da equação (3.10), o vetor de cargas de pressão não é mais aquele vetor que era conhecido no passo de tempo anterior, e sim o vetor de cargas de pressão calculado na iteração anterior, tendo que ser atualizado em todas as iterações até atingir a solução dentro do limite de erro relativo previamente estabelecido.

Neuman (1973) sugere que o cálculo dos coeficientes F , H , Q e Q' seja efetuado no ponto médio do intervalo de tempo, resultando em

$$\left(\frac{\bar{F}^{n+\frac{1}{2},m}}{\Delta t} + \bar{H}^{n+\frac{1}{2},m}\right)\bar{\psi}^{n+\frac{1}{2},m+1} = \frac{\bar{F}^{n+\frac{1}{2},m}}{\Delta t} \cdot \bar{\psi}^{n+\frac{1}{2},m} + \bar{Q}^{n+\frac{1}{2},m} - \bar{Q}^{n+\frac{1}{2},m} + \bar{F}_\theta \cdot \frac{\bar{\theta}^{n+\frac{1}{2},m} - \bar{\theta}^n}{\Delta t} \quad (3.13)$$

Definindo-se

$$\bar{D}^{n+\frac{1}{2},m} = \left(\frac{\bar{F}^{n+\frac{1}{2},m}}{\Delta t} + \bar{H}^{n+\frac{1}{2},m}\right) \quad (3.14)$$

$$\bar{R}^{n+\frac{1}{2},m} = \frac{\bar{F}^{n+\frac{1}{2},m}}{\Delta t} \cdot \bar{\psi}^{n+\frac{1}{2},m} + \bar{Q}^{n+\frac{1}{2},m} - \bar{Q}'^{n+\frac{1}{2},m} + \bar{F}_\theta \cdot \frac{\bar{\theta}^{n+\frac{1}{2},m} - \bar{\theta}^n}{\Delta t} \quad (3.15)$$

e substituindo-as na equação matricial 3.13, obtém-se então a equação (3.16), que constitui o sistema de equações a ser resolvido em cada iteração até que seja atingido o critério de convergência.

$$\overline{D}^{\frac{n+1}{2},m} \cdot \overline{\psi}^{\overline{n+1},m+1} = \overline{R}^{\frac{n+1}{2},m} \quad (3.16)$$

3.6. Critério de convergência

Neste trabalho foi adotado como critério de convergência o algoritmo proposto por Gerscovich (1994), que utiliza o método de Picard baseado nas propostas de Neuman (1973) e de Huyakorn e Pinder (1983). Este critério foi também empregado por Machado Jr. (2000) na versão original do GEOFLUX.

Neste algoritmo, uma primeira aproximação para a carga de pressão é calculada a partir da condição inicial:

$$\overline{\psi}^{\overline{n},m} = \overline{\psi}^{\frac{n+1}{2},m} = \overline{\psi}^0 \quad (3.17)$$

Em seguida, uma segunda aproximação $\overline{\psi}^{\overline{n+1},m}$ é calculada através de uma extrapolação linear :

$$\overline{\psi}^{\overline{n+1},m} = \overline{\psi}^{\overline{n}} + 2(\overline{\psi}^{\frac{n+1}{2},m} - \overline{\psi}^{\overline{n}}) \quad (3.18)$$

Com o vetor de carga de pressão da equação (3.18) entra-se na equação (3.16) e calcula-se um novo vetor de cargas de pressão representado por $\overline{\psi}^{\overline{n+1},m+1}$, que em seguida é verificado pelo critério de convergência definido por:

$$\left\| \frac{\overline{\psi}^{\overline{n+1},m+1} - \overline{\psi}^{\overline{n+1},m}}{\overline{\psi}^{\overline{n+1},m+1}} \right\| \leq Tolerância \quad (3.19)$$

na qual:

$$\| \| = \sqrt{\sum \psi_i^2} \quad (3.20)$$

denota a norma Euclidiana do vetor ψ_i .

Se o critério de convergência não for satisfeito, então o valor de $\bar{\psi}^{n+\frac{1}{2},m+1}$ é atualizado através de

$$\bar{\psi}^{n+\frac{1}{2},m+1} = \frac{\frac{\bar{\psi}^{n+1,m+1} + \bar{\psi}^n}{2} + \bar{\psi}^{n+\frac{1}{2},m}}{2} \quad (3.21)$$

Com este novo valor de $\bar{\psi}^{n+\frac{1}{2},m+1}$ reavaliam-se os coeficientes da equação (3.16), obtém-se uma nova aproximação para $\bar{\psi}^{n+1,m+1}$, e mais uma vez o critério de convergência na equação (3.19) é verificado. Esse processo é repetido até que o critério de convergência seja satisfeito, quando então o valor de $\bar{\psi}^{n+\frac{1}{2},m+1}$ é finalmente calculado como

$$\bar{\psi}^{n+\frac{1}{2},m+1} = \bar{\psi}^{n+1,m+1} + \frac{\bar{\psi}^{n+1,m+1} - \bar{\psi}^n}{2} \quad (3.22)$$

Concluído o ciclo iterativo, atualiza-se o valor de $\bar{\psi}^n$

$$\bar{\psi}^n = \bar{\psi}^{n+1,m+1} \quad (3.23)$$

e continua-se à marcha no tempo.

3.7.

Matrizes e vetores

Neste item são apresentadas as matrizes B , \bar{H} e \bar{F} , assim como os vetores \bar{Q} , \bar{Q}' e \bar{F}_θ utilizados na equação (3.13). Apresentam-se ainda o processo de determinação da matriz de condutividade hidráulica $\bar{K}(\psi)$ e do coeficiente de retenção específica $C(\psi)$.

Todas as matrizes e vetores foram avaliados para análises tridimensionais utilizando o elemento tetraédrico TETR4. e modo análogo, mas simplificado, podem ser obtidas as correspondentes matrizes e vetores para o elemento triangular TRIA3 utilizado em análises de fluxo 2D.

3.7.1.

Matriz \bar{B}

O operador diferencial para problemas tridimensionais pode ser escrito como:

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \quad (3.24)$$

e a matriz das funções de interpolação para elementos tetraédricos como

$$N = \begin{bmatrix} N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \\ N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \\ N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \end{bmatrix} \quad (3.25)$$

Substituindo as equações (3.24) e (3.25) na equação (2.30) tem-se:

$$\bar{B} = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} \begin{bmatrix} N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \\ N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \\ N_1(\xi, \eta, \zeta) & N_2(\xi, \eta, \zeta) & N_3(\xi, \eta, \zeta) & N_4(\xi, \eta, \zeta) \end{bmatrix} \quad (3.26)$$

Fazendo-se uma transformação de coordenadas globais para coordenadas locais, com a aplicação da regra da cadeia do cálculo diferencial e rearranjando-se os termos resulta em

$$\bar{B}_{3 \times 4} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \xi} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \eta} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \zeta} \end{bmatrix} \quad (3.27)$$

Aplicando o conceito da matriz Jacobiana, responsável pela transformação das derivadas espaciais em relação do sistema de coordenadas global para o natural de coordenadas (ξ, η, ζ) , vem:

$$J_{3 \times 3} = \begin{bmatrix} \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \xi} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \eta} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \zeta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix} \quad (3.28)$$

onde (x_i, y_i, z_i) , representa as coordenadas globais dos nós do tetraedro mostrado na figura 3.4 .

A inversa da matriz Jacobiana é dada pela equação:

$$J^{-1}_{3 \times 3} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \quad (3.29)$$

que substituída na equação (3.27) produz:

$$\bar{B}_{3 \times 4} = J^{-1}_{3 \times 3} \begin{bmatrix} \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \xi} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \xi} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \eta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \eta} \\ \frac{\partial N_1(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_2(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_3(\xi, \eta, \zeta)}{\partial \zeta} & \frac{\partial N_4(\xi, \eta, \zeta)}{\partial \zeta} \end{bmatrix} \quad (3.30)$$

Aplicando-se, finalmente, as correspondentes funções de interpolação para o elemento TETR4, definidas na equação (3.2), tem-se finalmente

$$\bar{B}_{3 \times 4} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \end{bmatrix}^{-1} \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \quad (3.31)$$

onde se nota que a matriz B é função única das coordenadas globais dos nós do elemento.

3.7.2. Matriz \bar{H}

A matriz H , denominada matriz de fluxo, resulta da integral apresentada na equação (2.33). Efetuando-se a transformação do sistema de coordenadas tem-se

$$\bar{H}_{4 \times 4} = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} B_{4 \times 3}^T \cdot \bar{K}(\psi)_{3 \times 3} \cdot B_{3 \times 4} \cdot |J| \cdot d\xi \cdot d\eta \cdot d\zeta \quad (3.32)$$

onde $|J|$ é o determinante da matriz jacobiana, e $\overline{K}(\psi)$ é a matriz de condutividade hidráulica, cuja determinação será discutida posteriormente.

A integração da equação (3.32) pode ser obtida analiticamente para tetraedros com 4 nós, pois a matriz B é constante no elemento. Logo,

$$\overline{H}_{4 \times 4} = B_{4 \times 3}^T \cdot \overline{K}(\psi)_{3 \times 3} \cdot B_{3 \times 4} \cdot \frac{|J|}{6} \quad (3.33)$$

3.7.3. Matriz \overline{F}

A matriz F resulta da integral apresentada na equação (2.37) que, em coordenadas locais, é expressa por:

$$\overline{F}_{4 \times 4} = \delta_{i,j} \cdot C_i(\psi) \cdot \sum_e \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} N_i \cdot |J| \cdot d\xi \cdot d\eta \cdot d\zeta \quad (3.34)$$

onde $\delta_{i,j}$ é o delta de Kroenecker e $C_i(\psi)$ é a capacidade de retenção específica avaliada no nó i , cuja determinação também será discutida posteriormente.

$$\overline{F}_{4 \times 4} = \delta_{i,j} \cdot C_i(\psi) \cdot \frac{|J|}{24} \quad (3.35)$$

3.7.4. Vetor \overline{Q}

Vetor que representa às vazões nodais impostas como condição de contorno prescritas (condição de Newman), expresso pela integral apresentada na equação (2.34). O vetor \overline{Q} pode ser especificado diretamente como entrada de valor nodal ou facilmente calculado quando representado de forma distribuída sobre a face do elemento.

No GEOFLUX3D, quando o fluxo for distribuído e imposto na direção normal à face do elemento, o valor da vazão nodal será dado pela taxa de fluxo

$(L^3/T/L^2)$ multiplicada pela área da respectiva face (L^2) e dividido finalmente pelo numero de nós (3) ligados à face em questão do elemento TETR4.

3.7.5. \bar{Q}' Vetor \bar{Q}'

O vetor \bar{Q}' incorpora uma parcela de vazão relacionada com efeitos gravitacionais (carga de elevação) e é definido pela integral apresentada na equação (2.35).

Com a transformação do sistema de coordenadas, a equação em termos de coordenadas locais é escrita como :

$$\bar{Q}'_{4 \times 1} = \int_0^1 \int_0^{1-\xi} \int_0^{1-\xi-\eta} B_{4 \times 3}^T \cdot \bar{K}(\psi)_{3 \times 3} \cdot \bar{e}_{3 \times 1} \cdot |J| \cdot d\xi \cdot d\eta \cdot d\zeta \quad (3.36)$$

cuja solução é dada por:

$$\bar{Q}'_{4 \times 1} = B_{4 \times 3}^T \cdot \bar{K}(\psi)_{3 \times 3} \cdot \bar{e}_{3 \times 1} \cdot \frac{|J|}{6} \quad (3.37)$$

Onde:

$$\bar{e}_{3 \times 1} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (3.38)$$

3.7.6. \bar{F}_θ Vetor \bar{F}_θ

Vetor definido pela integral apresentada na equação (3.12), cuja solução analítica é dada por:

$$\bar{F}_\theta_{4 \times 1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \cdot \frac{|J|}{24} \quad (3.39)$$

3.7.7.

Matriz de condutividade hidráulica $\bar{K}(\psi)$

Para o caso mais geral, na análise de fluxo 3D em um meio anisotrópico, $\bar{K}(\psi)$ representa uma matriz (3x3) com nove componentes não-nulas. Assumindo o tensor como simétrico, é possível definir em qualquer ponto do domínio de fluxo um sistema de coordenadas locais para o qual $\bar{K}(\psi)$ é diagonal.

O GEOFLUX3D permite variar a orientação das direções principais de fluxo nos elementos, aplicando-se uma rotação nos eixos locais de coordenadas de modo que se tornem coincidentes com as direções principais do tensor de condutividade hidráulica.

As componentes principais na condição saturada K_{1s} , K_{2s} , K_{3s} , junto com os co-senos dos ângulos formados entre as direções principais do tensor de condutividade e os eixos do sistema global de coordenadas, devem ser especificados.

As componentes principais K_1 , K_2 e K_3 são avaliadas em cada elemento em função das componentes principais saturadas K_{1s} , K_{2s} , K_{3s} e da carga de pressão (ψ), caso o meio se encontre não-saturado. Para essa avaliação, o GEOFLUX3D incorporou duas formas de análise; na primeira, utiliza-se a função de condutividade hidráulica apresentada por Srivastava e Yeh (1991), dada pela equação (2.7), enquanto que na segunda, faz-se uso da função de condutividade hidráulica proposta pelo modelo de van Genuchten (1980), expressa pela equação (2.8).

Assim, as componentes principais são transformadas para o sistema de coordenadas globais (x,y,z) no início das simulações através das seguintes relações:

$$K_{xx} = K_1 a_{11} a_{11} + K_2 a_{12} a_{12} + K_3 a_{13} a_{13} \quad (3.40a)$$

$$K_{yy} = K_1 a_{12} a_{12} + K_2 a_{22} a_{22} + K_3 a_{23} a_{23} \quad (3.40b)$$

$$K_{zz} = K_1 a_{13} a_{13} + K_2 a_{23} a_{23} + K_3 a_{33} a_{33} \quad (3.40c)$$

$$K_{xy} = K_1 a_{11} a_{12} + K_2 a_{12} a_{22} + K_3 a_{13} a_{23} \quad (3.40d)$$

$$K_{xz} = K_1 a_{11} a_{13} + K_2 a_{12} a_{23} + K_3 a_{13} a_{33} \quad (3.40e)$$

$$K_{yz} = K_1 a_{12} a_{13} + K_2 a_{22} a_{23} + K_3 a_{23} a_{33} \quad (3.40f)$$

onde a_{ij} representa o co-seno do ângulo formado entre a direção principal i da matriz $\overline{K}(\psi)$ e o eixo j do sistema de coordenadas globais.

3.7.8.

Capacidade de retenção específica $C(\psi)$

Para o tratamento da capacidade de retenção específica na zona não-saturada, apresentam-se também duas alternativas: a primeira, obtendo-se a capacidade de retenção específica a partir do modelo de Srivastava e Yeh (1991),

$$C(\psi) = \frac{d\theta}{d\psi} = (\theta_s - \theta_r) \cdot \alpha_{\text{exp}} \cdot e^{\alpha_{\text{exp}} \cdot \psi} \quad (3.41)$$

Na segunda alternativa, definindo-se a capacidade de retenção específica com base no modelo de van Genuchten (1980),

$$C(\psi) = \frac{d\theta}{d\psi} = \frac{\alpha_{vg} \cdot p \cdot q \cdot (\theta_s - \theta_r) \cdot (\alpha_{vg} \cdot |\psi|)^{q-1}}{(1 + (\alpha_{vg} \cdot |\psi|)^q)^{p+1}} \quad (3.42)$$

Para a zona saturada e para a zona de ascensão capilar a capacidade de retenção específica é normalmente nula. Porém, é conveniente considerar um valor de $C(\psi)$ diferente de zero para estas regiões, mas bastante baixo, com o propósito de evitar dificuldades de origem numérica.

Paniconi *et al.* (1991) destacam que numericamente uma capacidade de retenção específica diferente de zero preserva o caráter parabólico da equação diferencial, superando portanto dificuldades na convergência que podem surgir caso a equação se torne elíptica tanto na região residual quanto na de saturação e as condições de contorno naturais não forneçam uma solução única. Valores de capacidade de retenção específica não nulos são necessários para evitar uma singularidade na interface entre as regiões saturada e não-saturada conforme observado por Machado Jr. (2000). Portanto, adota-se para a zona saturada, para a zona de ascensão capilar e ainda para a região de saturação residual um valor de $C(\psi)$ bastante pequeno, mas diferente de zero. Nos exemplos estudados nesta dissertação considerou-se um valor da ordem de 10^{-10} .

3.8.

Armazenamento de dados e solução do sistema de equações

A equação matricial (3.16) tem que ser resolvida em cada iteração. Para análises 3D são geradas uma enorme quantidade de equações, dependendo da quantidade de nós que compõem a malha de elementos finitos e do tipo de elemento empregado. No caso do elemento TETR4, conforme já mencionado, é necessário uma discretização espacial e temporal bastante pormenorizada.

A matriz dos coeficientes representada por \overline{D} na equação 3.16 é esparsa, visto que nem todos os nós são interconectados entre si, sendo a maioria dos elementos desta matriz nulos. Procura-se então um sistema de armazenamento eficiente a fim de economizar espaço na memória e agilizar o tempo de processamento.

No programa GEOFLUX3D implementou-se um sistema de armazenamento bastante versátil, onde a matriz de coeficientes \overline{D} constituída por um número de coeficientes igual ao quadrado do número de nós ($ne \times ne$), foi armazenada em outras duas matrizes denominadas Posic ($nne \times 50$) e MatVal ($nne \times 50$), onde nne denota o número de coeficientes não nulos. A matriz Posic contém os indicadores das posições dos coeficientes não nulos (nne) da matriz \overline{D} , enquanto que a matriz MatVal contém propriamente os valores dos coeficientes.

Após o preenchimento de ambas as matrizes, se procede então à solução do sistema de equações pelo método do gradiente bi-conjugado. constatando-se que os tempos para a solução do sistema de equações são muito rápidos, mesmo com sistemas de equações de grande porte.

Com a finalidade de esquematizar a técnica de implementação do algoritmo e as vantagens da opção de armazenamento da matriz esparsa nas matrizes Posic e MatVal, apresenta-se a seguir um exemplo com a malha simplificada da figura 3.8, que contém 6 nós, denotados por O, formando 4 elementos triangulares, denotados por Δ .

A tabela 3.1 apresenta as incidências nodais de todos os elementos.

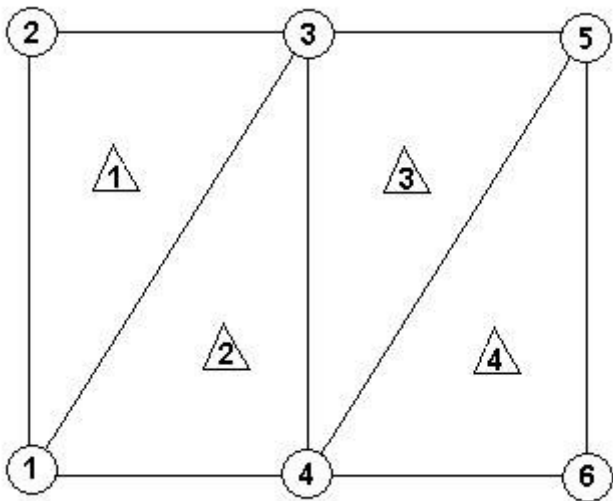


Figura 3.8 – Malha de elementos finitos simplificada.

Tabela 3.1 – Incidência nodal dos elementos da malha simplificada.

Elementos	Incidências nodais		
1	1	3	2
2	1	4	3
3	4	5	3
4	4	6	5

Observando-se as incidências nodais, a tabela 3.2 pode ser construída para a montagem da matriz Posic, onde a primeira coluna representa os nós principais, em ordem ascendente, e as colunas seguintes representam os nós com os quais aqueles são interligados.

Tabela 3.2 – Montagem da matriz Posic.

Nó principal	Nós ligados ao nó principal			
1	3	2	4	-
2	1	3	-	-
3	1	2	4	5
4	1	5	3	6
5	4	3	6	-
6	4	5	-	-

Nota-se que para este exemplo seriam reservados na memória do computador um total de $2 \times ne \times 50$ posições enquanto que armazenando-se toda a matriz esparsa \bar{D} seriam necessárias $ne \times ne$ posições.

Assim, é possível determinar-se o valor de ne para o qual a técnica de armazenamento apresentada é a mais indicada e eficiente,

$$2 \times ne \times 50 = ne \times ne \quad (3.43)$$

Resolvendo, obtém-se $ne = 100$, o que significa que para malhas contendo uma quantidade de nós superior a 100, o método de armazenamento em duas matrizes otimiza as posições ocupadas de memória e resultando num processamento mais rápido do sistema de equações.