

# 5

---

## Implementação do Sistema Integrado

### 5.1. Introdução

Uma vez definida a metodologia para o desenvolvimento da modelagem física, apresentada no Capítulo 3, passa-se para a descrição das implementações que visam incorporar ao sistema *Recon* um pré-processador, um módulo de análise e um pós-processador integrados, provendo um ambiente único para o balanceamento dentro da proposta apresentada.

Neste capítulo, inicialmente, é apresentada a nova organização das classes de atributos do sistema, no caso extensões das classes originais já descritas no Capítulo 3 e que foram implementadas para a representação dos novos atributos incorporados em função da integração do módulo de análise. Na seqüência é descrita a implementação, dentro do programa de análise, das novas restrições impostas ao deslocamento nodal a fim de viabilizar a simulação da movimentação de um bloco sobre uma falha. Finalmente são apresentadas as etapas necessárias para a integração do sistema de balanceamento com o módulo de análise, quando são descritas as implementações do pré- e do pós-processador.

### 5.2. Organização das Classes de Atributos

Com a inserção da modelagem por Elementos Finitos no sistema *Recon*, surge a necessidade de se definir novas classes para melhor gerenciar os novos atributos e funcionalidades adicionadas. Uma questão então passa a ser aonde acrescentar, na estrutura de dados descrita no Capítulo 3, as informações referentes à malha de elementos finitos e as subdivisões das arestas que servem de base para a geração da malha. Outra questão diz respeito aos resultados obtidos no programa de análise e como armazená-los.

Finalmente, existem também outros atributos associados às camadas que também deverão ser pendurados na classe de atributos das faces. Para guardar todas essas informações fez-se, uma reestruturação na hierarquia de classes de atributos original, sendo algumas classes redefinidas, enquanto outras foram criadas.

A Figura 5.1 apresenta a reestruturação da hierarquia das classes responsáveis pelo controle dos atributos das entidades geológicas do sistema *Recon*. É possível observar quatro níveis de abstração desde a primeira camada, que correspondem as entidades topológicas do HED (geometria do modelo) até a última camada de abstração dos atributos geológicos.

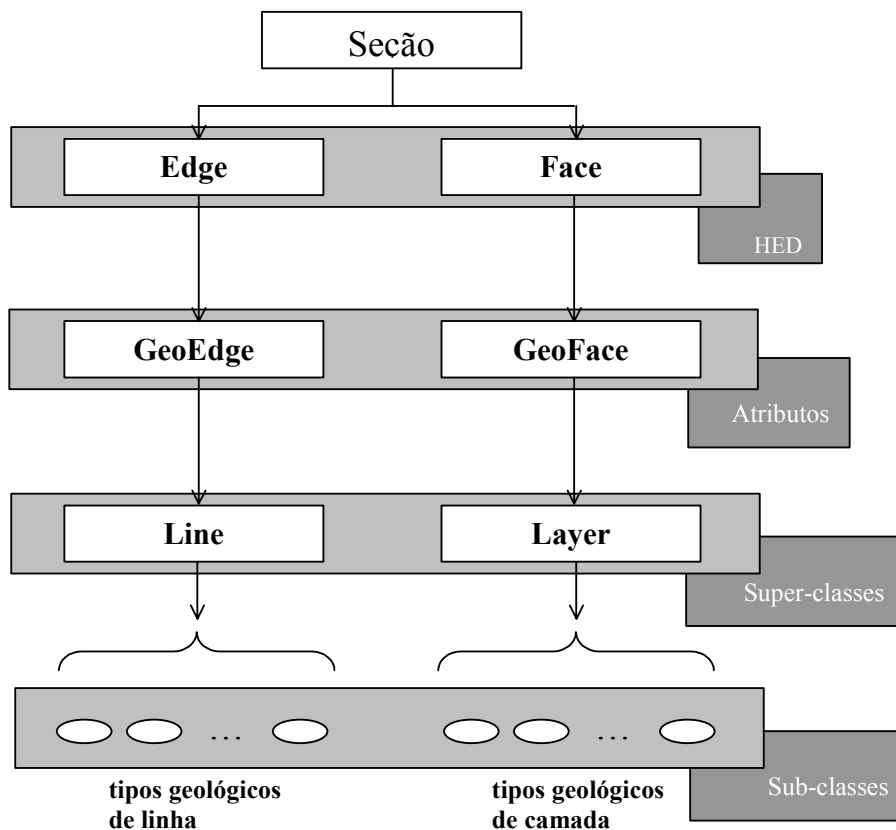


Figura 5.1 - Nova organização das classes de atributos no Sistema Recon.

Às entidades topológicas da subdivisão planar *Face* e *Edge* estão agora relacionadas as classes *GeoFace* e *GeoEdge*, respectivamente representando os blocos de atributos das faces e arestas. A primeira diferença que se pode observar em relação à versão original do

*Recon* é que agora existe mais uma camada de abstração entre a linha topológica e os tipos geológicos de linha. Esta camada é representada pela classe *GeoEdge*. A segunda alteração visível diz respeito à substituição da classe *Polygon* pela classe *GeoFace*. Essa nova hierarquia permite acomodar as novas informações de forma mais organizada. Na seqüência são descritas de forma mais detalhada as classes acima citadas.

### 5.2.1. Classe *GeoFace*

Esta classe contém métodos relacionados aos atributos geológicos associados às faces da subdivisão espacial. Uma face geológica no *Recon* apresenta informações como por exemplo a que módulo e a que tipo de camada a face pertence. Tem também pendurados a si dados dos marcadores de transformação. Somadas a essas informações, já presentes na antiga classe *Polygon*, *GeoFace* passa a apresentar também as informações referentes à malha de elementos finitos e aos dados relacionados aos resultados obtidos do módulo de análise, vide Figura 5.2.

<b>Classe <i>GeoFace</i>:</b>	
• <i>Layer</i>	<i>ponteiro para a superclasse Layer</i>
• <i>Mdul</i>	<i>ponteiro para o module a que pertence</i>
• <i>Mark</i>	<i>ponteiro para a lista de marcadores de transformação</i>
• <i>Mesh</i>	<i>geometria da malha de elementos finitos</i>
• <i>AnRes</i>	<i>resultados provenientes do módulo de análise</i>

Figura 5.2 – Definição da classe *GeoFace*.

O algoritmo de análise é sempre aplicado a um módulo de trabalho. Este, por sua vez, é composto por uma lista de faces HED. A malha de elementos finitos é mais um atributo atrelado à face. Uma vez processados os valores de deformação, que são o objetivo final da presente análise numérica, precisa-se mais uma vez amarrá-los às entidades topológicas do modelo. Para tal foi criado um outro atributo pendurado à face topológica, a classe *AnRes*, que guarda as informações referentes aos resultados obtidos pelo módulo de análise.

No campo que representa um ponteiro para o módulo ao qual a face pertence foi pendurado um informação adicional: a malha unificada do módulo, o que é discutido detalhadamente na Seção 5.4.1.3.

A classe *GeoFace* também gerencia a detecção automática de camadas geológicas, que pode se dar de três formas distintas e que será melhor explicado na definição da classe *Layer*.

### 5.2.2. Classe *GeoEdge*

De forma análoga, esta classe apresenta métodos relacionados aos atributos geológicos de linha. Uma linha geológica, *GeoEdge*, no *Recon* representa o mesmo papel da classe *GeoFace* com relação ao bloco de atributos das faces. Esta nova abstração foi implementada para gerenciar os novos atributos necessários à modelagem em função da existência da malha de elementos finitos que discretiza o módulo deformado. A geração da malha se dá em função do número de subdivisões das arestas que compõem o contorno da face. Com a criação desta classe, é possível agora armazenar informações sobre a subdivisão das linhas para a geração das malhas associadas às faces. Além disso, a classe *GeoEdge* guarda também um campo para armazenar nomes associados às linhas. Este novo atributo será útil futuramente quando serão gerados modelos de 2 1/2 dimensões (geração de superfícies 3D através da interpolação de várias seções transversais, defasadas porém no seu sentido ortogonal). A Figura 5.3 ilustra a nova classe *GeoEdge*.

<b><i>Classe GeoEdge</i></b>	
• <b>line</b>	<b><i>ponteiro para a superclasse Line</i></b>
• <b>nsdv</b>	<b><i>número de subdivisões da linha</i></b>
• <b>label</b>	<b><i>nome identificador da linha</i></b>

Figura 5.3 – Definição da classe *GeoEdge*.

### 5.2.3. Classe Line

A classe *Line* representa a superclasse de atributos geológicos associados às linhas. É da super-classe *Line* que são derivadas as classes que formam o conjunto de tipos geológicos de linha, aqui identificados pelos nomes: horizonte, falha, topo de sal, limite de seção, discordância e indefinida. A Figura 5.4 ilustra melhor a classe *Line* e seu relacionamento com as sub-classes citadas.

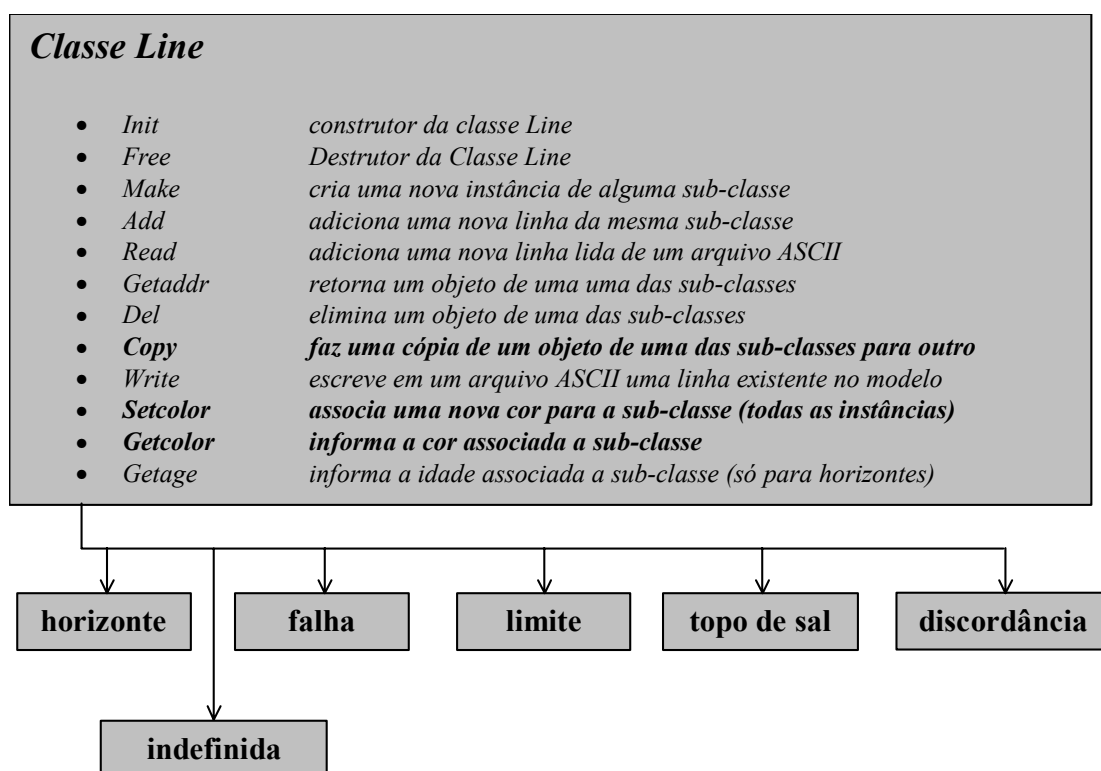


Figura 5.4 – Definição da classe *Line* e suas sub-classes.

Todos os métodos que não estão em negrito são herdados pelas classes derivadas. As classes derivadas são na realidade representações dos tipos de linhas geológicas.

#### 5.2.4. Classe *Layer*

A classe *Layer* representa a super-classe de atributos geológicos associados às camadas. É da super-classe *Layer* que são derivadas as classes que formam o conjunto de tipos geológicos de camadas, aqui identificados pelos nomes: depósito, indefinida, sal e externa.

Os depósitos são as camadas associadas às eras geológicas. A camada externa representa o atributo da entidade topológica face externa do HED. Uma camada que ainda não teve associada a si algum tipo recebe o atributo de indefinida.

A Figura 5.5 apresenta os principais métodos da classe *Layer* e seu relacionamento com as sub-classes citadas. A classe *Layer* guarda todas as informações referentes às camadas de uma forma geral. Dentre essas informações, tem-se o tipo da camada e uma lista com as propriedades de camadas que o *Recon* trata ao aplicar as suas transformações. Essas propriedades são:

- porosidade inicial
- taxa de decaimento [1/km]
- velocidade intervalar [m/s]
- código de litologia
- compactação [s/n]

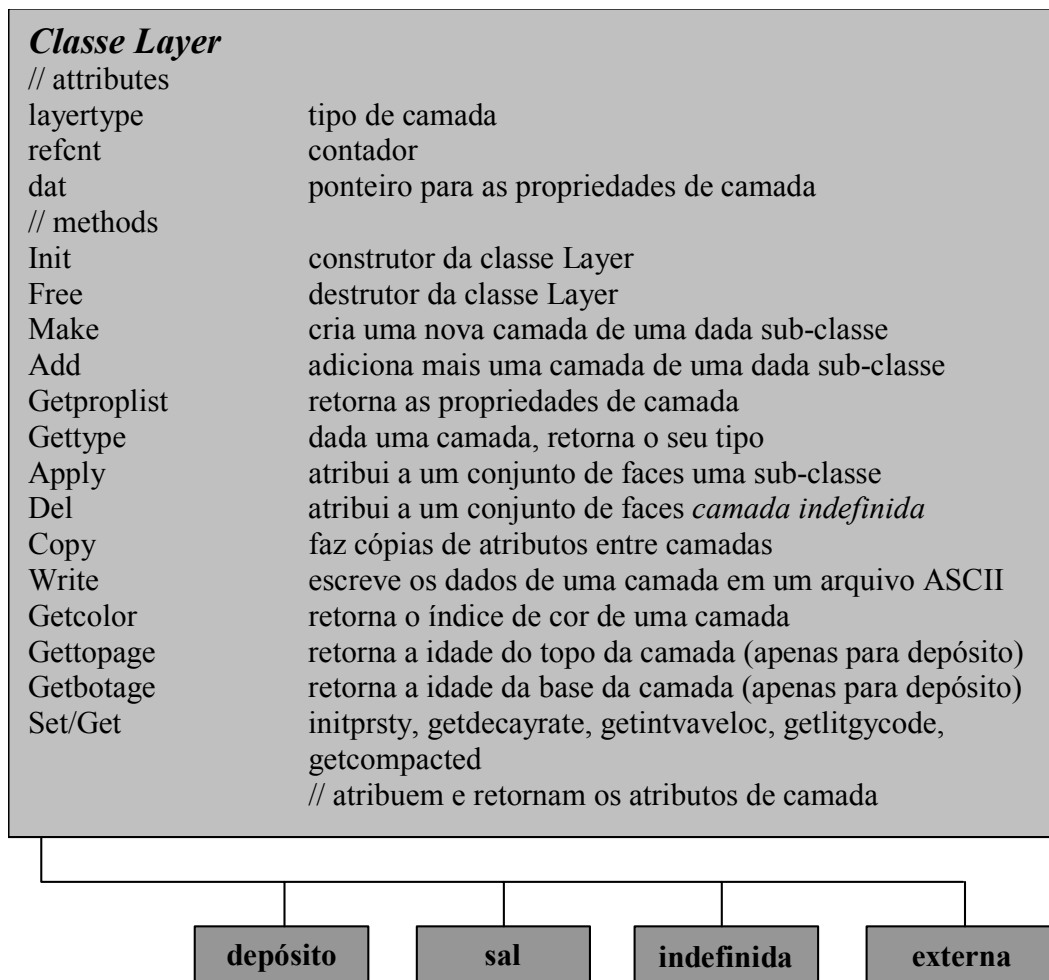


Figura 5.5 – Definição da classe *Layer* e suas sub-classes.

Os métodos que compõem a interface da classe *Layer* são apresentados na descrição da classe acima. Através deles pode-se criar e atribuir propriedades a um dado conjunto de faces topológicas que compõe uma camada geológica.

As sub-classes da classe *Layer*, que representam os tipos de camadas presentes no *Recon*, herdam as propriedades concernentes a camadas de uma forma geral e ainda guardam para si propriedades específicas de seus tipos.

A atribuição de um tipo de camada a faces pode se dar de duas formas distintas no sistema. O usuário pode explicitamente, através da interface gráfica, atribuir a um conjunto de faces topológicas uma determinada camada, ou esta atribuição pode se dar de

forma implícita através dos parâmetros de detecção automática definidos na classe *GeoFace*. Neste caso, a atribuição pode ser feita baseada nos atributos da linha de topo e base conjuntamente, apenas baseada na linha de base ou ainda baseada no atributo que apresenta maior comprimento dentre todos os presentes no topo de uma dada camada.

### ***5.3. O Módulo de Análise***

Para o desenvolvimento desse trabalho foi utilizado, inicialmente para testes, o programa *Relax*, desenvolvido pelo Tecgraf/PUC-Rio. Por dispor-se de seu código fonte, esses testes puderam ser realizados com maior flexibilidade até que definitivamente optou-se por integrá-lo ao sistema *Recon* de forma a prover um ambiente único para a execução do move-sobre-falha não geométrico. O programa *Relax* baseia-se em Figueiredo [28], e implementa a técnica de Relaxação Dinâmica, descrita no Capítulo 4, acoplada ao Método dos Elementos Finitos.

#### ***5.3.1. O Programa Relax***

O Programa *Relax* foi implementado utilizando a linguagem de programação C [37], com a filosofia de programação orientada a objetos, POO. O programa consiste em três classes básicas:

- ***DrvClass*** – É a classe responsável pela definição do tipo de análise que vai ser realizada. Estas análises estão divididas em: *rezone*, que é a solução do modelo que contém passos de erosão, *load-step*, que é a solução do modelo que contém mais de um passo de carregamento e *standard*, que é a solução do modelo com um único passo de carregamento e sem erosão, vide Figura 5.6.



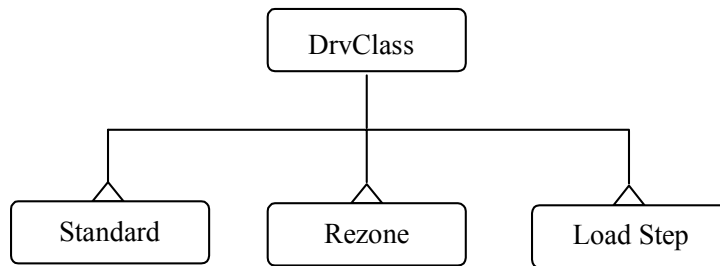


Figura 5.6 – Relax: Hierarquia das classes dos tipos de análise.

- **MatClass** – É a classe responsável pela definição dos comportamentos dos materiais dos elementos que serão utilizados na análise, admitindo o comportamento do material como sendo isotrópico, ou segundo as hipóteses de *Mohr Coulomb* ou *Von Mises*, vide Figura 5.7.

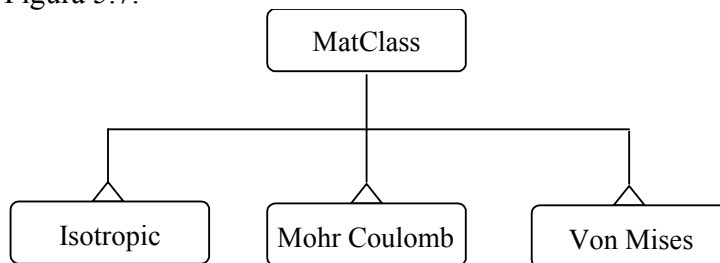


Figura 5.7 – Relax: Hierarquia das classes dos materiais.

- **ElmClass** – É a classe responsável pela definição dos tipos de elementos implementados no programa, vide Figura 5.8.

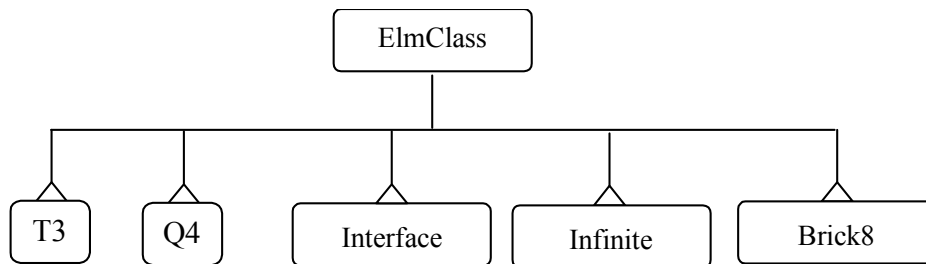


Figura 5.8 – Relax: Hierarquia das classes dos elementos.

O *Relax* contém cinco tipos de elementos finitos implementados. Para a representação de domínios bidimensionais pode-se utilizar elementos triangulares lineares, quadrilaterais de quatro nós e elementos infinitos de quatro e três nós. O programa faz também análise em modelos tridimensionais e para tal existe implementado o elemento hexaedral de oito nós (*Brick8*). Para modelagem de fraturamento é possível usar o elemento de interface linear, apenas para modelos bidimensionais.

O programa de análise comporta ainda considerações de não-linearidades geométricas. Para tal se faz necessário correções nas componentes cartesianas do tensor das tensões, utilizando-se o modelo seguido por Bathe [4].

A entrada dos dados no programa é originalmente feita via arquivo texto no formato *Neutral File* [60], onde são definidos todos os dados necessários à simulação de problemas geológicos em geral. Através dos rótulos predefinidos pelo programa é possível fazer a leitura dos dados referentes às suas respectivas informações. Existem informações relativas a quatro classes distintas de dados:

- **Configuração**
- **Material**
- **Casos de carregamento**
- **Erosão**

Os dados relativos à configuração podem ser referentes ao tipo de análise a ser processada (análise de deformação plana, tensão plana ou tridimensional), definição da tolerância de convergência, para avaliar a ordem de grandeza das forças internas do bloco, consideração de linearidade ou não-linearidade geométrica, etc.

Os dados referentes aos tipos de material podem ser isotrópico elástico linear [23], ortotrópico elástico linear, vários modelos elasto-plásticos perfeitos com superfícies de *Von Mises* [23] ou *Mohr-Coulomb*, modelos visco-elásticos de Maxwell ou Kelvin [45], modelo hiperbólico, modelo elástico linear para interface e modelo elasto-plástico perfeito para interface com superfície de Mohr-Coulomb.

No programa *Relax* é possível interpretar mais de um caso de carregamento. Estão previstos os seguintes casos de carregamentos:

- Gravidade
- Forças nodais
- Velocidades nodais
- Carregamento uniforme ou variavelmente distribuído
- Deslocamento nodal
- Tensão inicial uniforme
- Carga potencial nos elementos da malha
- Variação de temperatura

É previsto também pelo programa um tipo de análise de escavação, que é associado a um rótulo no arquivo de entrada para erosão.

A seguir resume-se a utilização do programa de análise, especialmente na definição dos parâmetros, dos são dados de entrada do algoritmo de relaxação dinâmica e das hipóteses adotadas para a modelagem física. Por fim são explicadas as alterações implementadas dentro do algoritmo original no *Relax* de forma a impor a restrição de deslocamentos ao longo da geometria da falha de acordo com a estratégia apresentada na Seção 4.5

### ***5.3.2. Dados de Entrada***

Nesta seção são definidos os dados de entrada do *Relax* utilizados nesse trabalho para a simulação do *move-sobre-falha*. Os dados foram agrupados de acordo com a classificação abaixo:

#### ***a. Configuração da Análise:***

Os parâmetros associados à configuração da análise são o seu tipo (estado plano de tensões ou estado plano de deformações) e os parâmetros associados a execução do algoritmo especificamente, como tolerância, número de passos, tempo do passo. Além desses, existem também parâmetros referentes a matriz de amortecimento e que não são

alterados por estarem diretamente associados a convergência do algoritmo. O modelo de análise adotado foi o estado plano de deformações.

*b. Material:*

Os valores referentes ao módulo de elasticidade e coeficiente de Poisson são parâmetros de entrada do usuário, cabendo portanto ao geólogo ou geofísico defini-los. O coeficiente de Poisson aqui sugerido é o de algum valor próximo a meio, o que simula uma quase inexistência de variação de volume.

Define-se também para o material o seu tipo de comportamento ao deformar-se (super-classe *MatClass*) Adotou-se neste trabalho um comportamento isotrópico elástico linear do material deformado como uma primeira aproximação, embora seja sabido que essa aproximação não é a mais próxima do que ocorre na realidade, já que estamos claramente trabalhando com grandes deslocamentos e não linearidade física. No entanto, devido ao elevado grau de incertezas quanto ao comportamento do material geológico, optou-se por essa aproximação. Planeja-se no futuro considerar o comportamento elasto-plástico para os materiais.

*c. Carregamentos:*

Os tipos de carregamentos solicitados pelo sistema de balanceamento ao programa de análise são deslocamentos prescritos, quando se faz a análise no sentido inverso do tempo e ação da gravidade quando se faz a simulação do deslizamento do bloco. É possível também simular cargas de atrito nos nós da malha em contato com a falha aplicando-se cargas concentradas nesses nós no sentido contrário ao deslocamento do módulo.

*d. Restrições:*

O *Relax* não apresenta na sua implementação opção para restrição ao deslocamento em direções inclinadas. Mas para modelar o deslizamento sobre uma falha lítrica não basta

amarrar um nó ao deslocamento na direção longitudinal da falha. A Seção 5.3.3 a seguir discute detalhadamente as implementações feitas no programa de análise para garantir tal comportamento.

### ***5.3.3. Restrição de Deslocamento Nodal Sobre a Falha***

Para amarrar o deslocamento desses nós de interface à geometria da falha definiu-se dois novos rótulos no arquivo de entrada para o programa *Relax*, representando assim uma restrição nodal variável, ao longo do processamento.

O primeiro deles, aplicado explicitamente aos nós da malha que pertencem a linha de falha, define que estes nós estão submetidos à restrição de deslocamentos ao longo da falha. Uma vez definida tal restrição, a esses nós já estará definido o seu caminho de deslocamento. São fornecidas as seguintes informações para cada nó:

***<número do nó>      <dn> <dt1> <dt2> <label da falha>***

Para cada nó, representado pelo seu identificador, fornecido pelo gerador da malha, são informados o estado das restrições nas direções normal e tangente (duas direções tangentes para modelos 3D) e a linha de restrição sobre a qual está submetido.

O segundo rótulo é simplesmente para definir a lista de falhas do modelo, quando se define as geometrias das poligonais que modelam as falhas e os seus respectivos identificadores, representadas por numeração.

O cálculo dos deslocamentos e velocidades nodais no *Relax* dependem dos graus de liberdade previamente especificados em cada uma das direções  $(x,y)$  ou  $(x,y,z)$ , caso o modelo seja 3D. No programa já existem os seguintes graus de liberdade:

- **FORCE**- quando um nó apresenta um grau de liberdade expressado em força, em uma dada direção, as velocidades e os deslocamentos são obtidos através, respectivamente, das expressões (4.38) e (4.39).
- **DISPLACEMENT** - quando um nó apresenta um grau de liberdade expresso em deslocamento, em uma dada direção, sua velocidade é nula nesta direção e ao deslocamento nodal é definido o valor atribuído para todas as iterações. É o que ocorre com os deslocamentos prescritos atribuídos a um nó.
- **VELOCITY** - quando um nó apresenta um grau de liberdade expresso em velocidade, em uma dada direção, em todas iterações será atribuído a este nó a mesma velocidade naquela direção. O incremento de deslocamento para cada passo naquela direção é obtido através do produto entre o tempo do passo e a velocidade prescrita definida

A estes tipos de graus de liberdade foi acrescido o grau de liberdade que especifica a restrição de trajetória do nó sobre a falha, denominado de **PATH-DISPLACEMENT** ou trajetória de deslocamento, cujo comportamento é descrito a seguir.

#### **5.3.4 – Implementação da Restrição**

Uma vez estruturados os dados referentes aos nós submetidos à nova restrição, parte-se para a implementação da mesma durante a execução do algoritmo de Relaxação Dinâmica.

Dentro do algoritmo, descrito no capítulo anterior, foi implementado um outro grau de liberdade denominado **PATH-DISPLACEMENT**. Ou seja, os nós localizados sobre a falha apresentam esse novo grau de liberdade.

A primeira abordagem definida para restringir o deslocamento dos nós com esse grau de liberdade corrigia as coordenadas obtidas originalmente pelo algoritmo para a posição equivalente sobre a falha e pode ser melhor compreendida através da Figura 5.9.

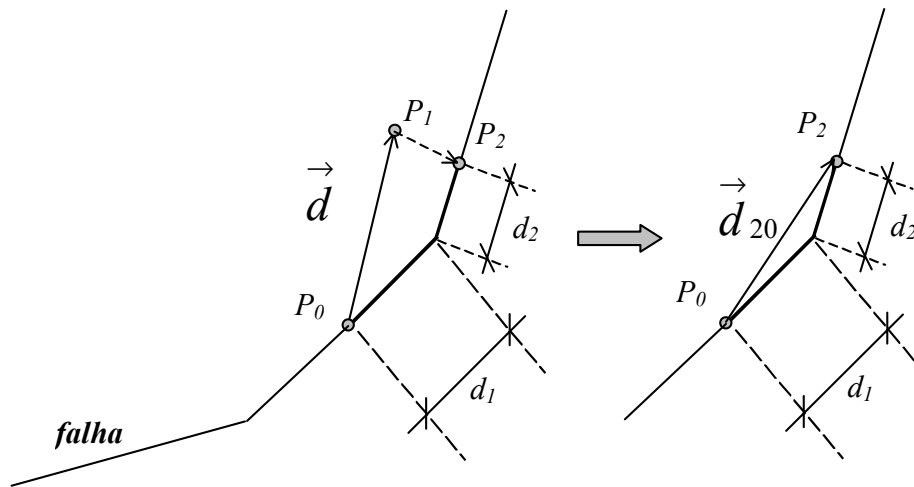


Figura 5.9 – Relax: Ajuste geométrico.

Em um determinado passo  $n$  tem-se o nó  $P_0$  pertencente a malha e situado sobre a falha. São processados os seus deslocamentos, definido pelo vetor  $\vec{d}$ , de tal forma que a sua nova posição é a do ponto  $P_1$ . Tendo os valores das coordenadas dos pontos  $P_0$  e  $P_1$ , é possível caminhar ao longo da falha, partindo-se de  $P_0$  de tal forma que se chegue até o ponto  $P_2$  que por definição terá a sua distância em relação ao ponto  $P_0$  igual ao módulo do vetor de deslocamentos  $d$ :

$$|\vec{d}| = d_1 + d_2 \quad (5.1)$$

Com isso é possível obter as coordenadas do ponto  $P_2$  situado sobre a falha e por conseguinte obter o deslocamento corrigido, definido pelo vetor  $\vec{d}_{20}$  e que será o novo valor de deslocamentos deste nó, definido pelos pontos  $P_0$  e  $P_2$ .

$$\vec{d}_{20} = P_2 - P_0 \quad (5.2)$$

Os resultados obtidos com essa abordagem não foram satisfatórios e o algoritmo nunca conseguiu obter uma configuração de equilíbrio tanto para problemas com carga de deslocamentos prescritos, quanto para ação da gravidade. Isso é devido ao fato de que para cada iteração as novas coordenadas dos nós eram obtidas em função de um sistema

de forças desequilibradas que por sua vez definiam velocidades até efetivamente serem obtidos os deslocamentos. As correções geométricas acima descritas definiam um vetor de deslocamentos nodais não compatível com as forças e velocidades atuantes nesses nós.

O procedimento correto deve calcular forças e velocidades em função das condições de contorno e daí naturalmente os deslocamentos obtidos serão consistentes com essas condições de contorno. Em função disso a abordagem acima descrita foi descartada e uma nova estratégia passou a ser adotada, no caso alterando o sistema de forças e velocidades, descrito na seqüência.

Considera-se inicialmente o caso de falhas planares, ou seja, falhas cuja representação em seções podem ser descritas por uma simples linha reta. Conforme é possível observar na Figura 5.10, fazer um nó deslizar sobre uma falha representa restringir o seu deslocamento na sua direção longitudinal (direção 1), ou seja, impedir que haja deslocamento na direção normal à falha (direção 2).

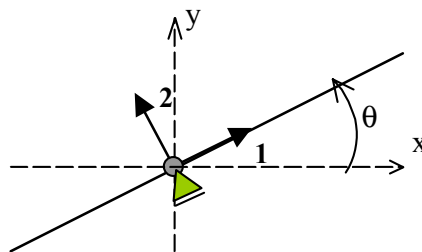


Figura 5.10 – Relax: Apoio inclinado.

O que se deseja é utilizar-se de apoios inclinados de primeiro gênero para restringir o deslocamento de um determinado nó ao longo de uma falha planar. Implementou-se, portanto, dentro do programa de análise, a restrição nodal representando apoios inclinados, já que o *Relax* não dispunha originalmente desse tipo de restrição.

O objetivo é permitir o deslocamento de um nó em uma direção que não é a dos eixos principais  $x$  e  $y$  e impedi-lo em uma outra direção também diferente de  $x$  e  $y$ , mas como o sistema trabalha com as coordenadas globais, as restrições nodais que se deseja



implementar podem ser definidas de acordo com as expressões (5.3), que associam deslocamentos nodais nas direções dos eixos globais com deslocamentos no sistema de coordenadas local.

$$\begin{cases} u_1 = u_x \cos \theta + u_y \operatorname{sen} \theta \\ u_2 = -u_x \operatorname{sen} \theta + u_y \cos \theta = 0 \end{cases} \quad (5.3)$$

Porém é mais fácil trabalhar-se nesse caso com o sistema de coordenadas locais, ou seja, com  $u_1$  livre e  $u_2$  fixo. O que se faz então é transformar, para cada iteração, forças, deslocamentos e velocidades dos nós sobre a falha, do sistema de coordenadas global para o sistema de coordenadas local. Para tal aplica-se, para todas essas grandezas, a matriz de rotação apresentada abaixo na expressão (5.4), onde  $\theta$  é ângulo de rotação no sentido anti-horário entre o sistema global e o sistema local.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \operatorname{sen} \theta \\ -\operatorname{sen} \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (5.4)$$

Tem-se então que, numa dada iteração, para os nós submetidos a restrição da falha, o seguinte procedimento:

1. *Obtém-se os cossenos diretores que definem a inclinação da falha e calcula-se o ângulo  $\theta$  de rotação;*
2. *As forças desequilibradas atuantes em um nó são decompostas no sistema de coordenadas local;*
3. *Os vetores de velocidades e deslocamentos, obtidos na iteração anterior, também são transformados para o sistema de coordenadas locais;*
4. *O novo vetor de velocidades, no sistema local, é obtido de forma idêntica a obtida para um nó sem restrições, exceto na direção normal ao apoio que é igualado a zero;*

5. Da mesma maneira que são obtidos para os nós sem restrição, os novos deslocamentos são obtidos. Como as velocidades na direção normal são nulas, por conseguinte o mesmo ocorrerá para os deslocamentos;
6. Uma vez obtidas forças desequilibradas, velocidades e deslocamentos no sistema local, retorna-se todas essas grandezas novamente para o sistema de coordenadas globais, aplicando-se a mesma matriz de rotação da expressão (5.4), porém com o valor angular de  $(-\theta)$ .

A forma como o *Recon* e o *Relax* aproximam curvas quaisquer é através de poligonais. No caso de falhas lítricas, sob o ponto de vista da restrição nodal, isso significa que um determinado nó de falha, pode estar submetido a mais de uma angulação de apoio inclinado diferente, bastando para isso que este nó mude de tramo ao se deslocar sobre esta falha. A Figura 5.11 ilustra essa situação:

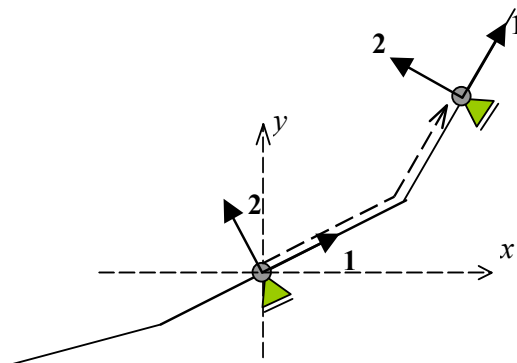


Figura 5.11 – *Relax*: Restrições ao deslocamento sobre a falha lítrica.

Nesse caso são necessários alguns outros procedimentos para garantir que ao mudar de tramo o nó tenha as suas restrições de deslocamento corretamente alteradas de forma que o algoritmo passe a buscar o equilíbrio do módulo de acordo com as suas novas condições de contorno.

A possibilidade do modelo ter condições de contorno variáveis, quanto as suas restrições ao deslocamento, exige que ao final de cada iteração sejam testados, para cada nó submetido ao deslocamento sobre a falha, em qual tramo este nó se encontra após obtido

o novo vetor de deslocamentos nodais. Criou-se então, na estrutura de dados do *Relax*, mais um campo na classe *Node*, que representa os nós da malha, para armazenar o tramo sobre o qual este nó se encontra e a direção na qual este nó se movimentou sobre a falha na iteração anterior. Com isso torna-se mais eficiente o processo de se testar geometricamente a posição deste nó sobre a falha.

A geometria das falhas é definida na entrada de dados do programa de análise sempre ordenadas no seu sentido horário, conforme ilustra a Figura 5.12. Esta convenção é necessária porque a detecção do tramo corrente do nó se baseia nessa ordenação. Assim para a falha definida a esquerda de A, a sua orientação correta é AB, enquanto que para a outra falha a orientação seria CA.

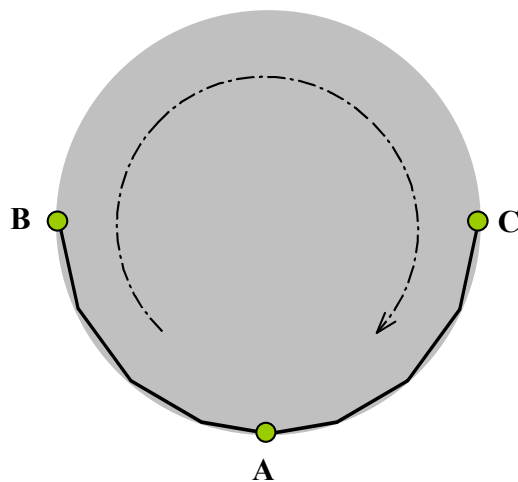
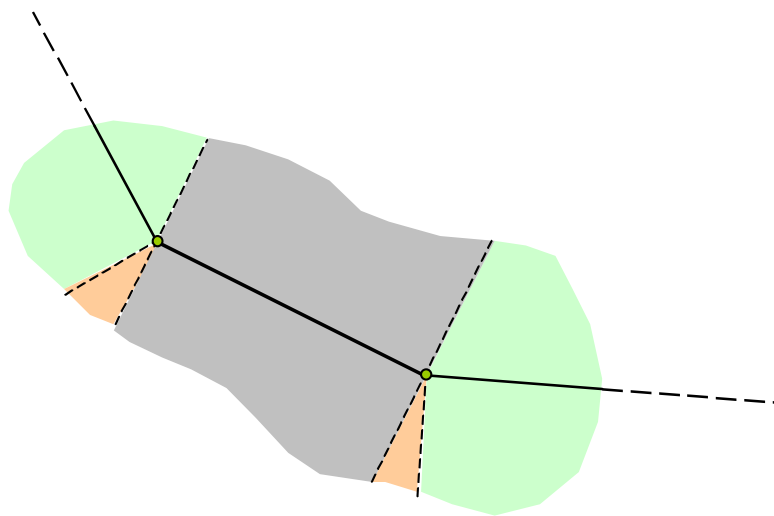


Figura 5.12 – *Relax*: Definição da orientação das falhas.

Se é sabido na iteração anterior aonde se encontrava o nó, a pesquisa começa a ser realizada justamente no mesmo tramo que se encontrava anteriormente e caso não mais nesse tramo esteja, o próximo candidato é justamente o seguinte de acordo com a direção para aonde ele esteve se deslocando na iteração anterior. Já o teste geométrico que define se o nó está de fato em um determinado tramo ou não é explicado a seguir.

Na Figura 5.13 abaixo são identificadas várias regiões possíveis para as quais um nó pode estar posicionado com relação ao tramo central, com a linha mais grossa, e que representa

justamente o tramo onde o nó testado se encontrava na iteração anterior. As regiões são delimitadas pelas linhas tracejadas, que são na realidade as linhas perpendiculares aos tramos corrente superior e inferior. O algoritmo geométrico determina se o ponto em estudo encontra-se na região cinza, o que define que ele continua na área de influência do mesmo tramo, na região laranja, o que determina que este nó se encontra em uma região de interseção entre o tramo corrente e um tramo vizinho ou na região verde, que determina que o nó em estudo efetivamente mudou de tramo.



*Figura 5.13 – Relax: Identificação das regiões próximas a falha.*

Após a identificação da região aonde se encontra o nó é feita a projeção do mesmo sobre o tramo que define a sua restrição e ao mesmo tempo é calculada a distância do nó para este tramo. Caso essa distância seja superior a uma certa tolerância, previamente definida de acordo com a tolerância do modelo HED, o processo é encerrado e a análise não é concluída.

A definição da região sobre a qual o nó se encontra é obtida através do cálculo do ângulo entre vetores que são obtidos de acordo com a Figura 5.14, que ilustra a situação de um ponto que permanece no mesmo tramo nas iterações anterior e corrente.

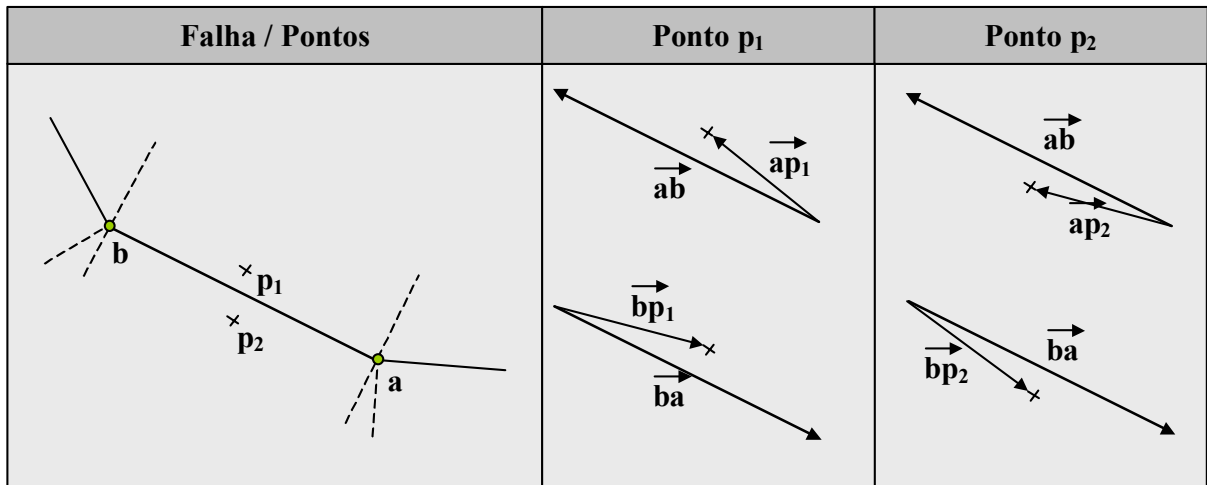


Figura 5.14 – Relax: Teste Geométrico Para Localizar um Ponto Próximo a Falha.

Obtidas as novas coordenadas de um nó  $p$ , e com base no tramo e direções desse nó na iteração anterior, são calculados dois ângulos,  $\theta_a$  e  $\theta_b$ , obtidos respectivamente através da diferença entre dos pseudo-ângulos entre os vetores  $ab$  e  $ap$  e  $ba$  e  $bp$ , definida pelas expressões (5.5).

$$\begin{cases} \theta_a = ps\_ang(ab) - ps\_ang(ap) \\ \theta_b = ps\_ang(ba) - ps\_ang(bp) \end{cases} \quad (5.5)$$

Onde  $ps\_ang$  representa o pseudo-ângulo [8] do vetor e que é definido de acordo com a Figura 5.15:

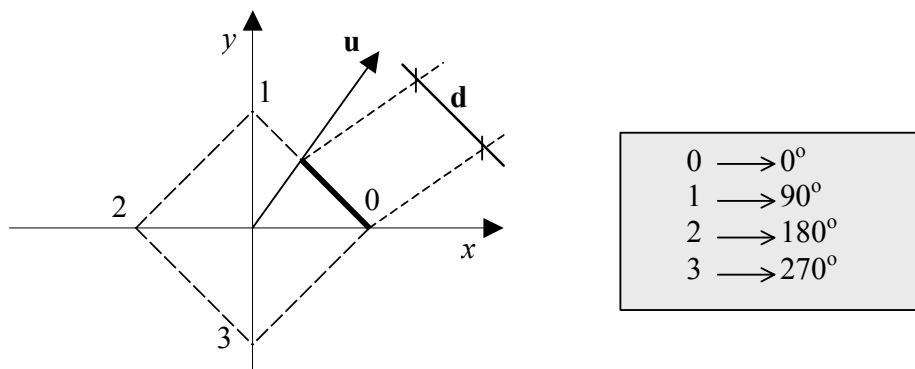


Figura 5.15 – Definição de pseudo-ângulo.

O pseudo-ângulo é uma forma mais eficiente computacionalmente de se obter valores de ângulos relativos de vetores. É obtido caminhando-se sobre o espaço paramétrico do quadrado unitário da Figura 5.15, sendo  $d$  o seu valor para o vetor  $u$ , por exemplo. O pseudo ângulo pode então variar de 0 a 4, exclusive.

Com isso, caso  $\theta_a$  e  $\theta_b$  apresentem ambos valores entre  $-1$  e  $1$ , significa que o nó em estudo permanece no mesmo tramo. Caso algum dos ângulos não apresente o seu valor dentro do intervalo  $-1$  e  $1$ , neste caso, repete-se o mesmo procedimento só que para o tramo referente ao ângulo fora do intervalo. Caso o valor de  $\theta_a$  referente ao novo tramo em estudo também esteja fora do intervalo  $[-1,1]$ , então este nó encontra-se na região de interseção, caso contrário, este nó estará sob a influência da restrição ao deslocamento do novo tramo em estudo.

Quando um nó posiciona-se na região de interseção entre dois tramos, representado na Figura 5.13 pela região laranja, este nó é projetado sobre o ponto de interseção entre dois tramos. Essa distância entre as coordenadas dos nós e a falha são limitadas de acordo com uma dada tolerância, que é dado de entrada do programa de análise, devendo portanto apresentar valores de baixa magnitude, já que representam um erro numérico do algoritmo de relaxação dinâmica. Esse erro torna-se relativamente maior quando o nó muda de tramo na falha de uma iteração do algoritmo para outra, conforme pode-se observar na Figura 5.16. Nessa situação igualmente projeta-se o nó sobre o seu novo tramo e atualiza-se as suas restrições ao deslocamento.

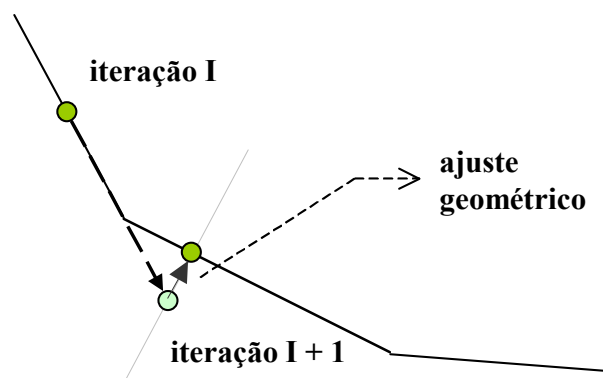


Figura 5.16 – Relax: Ajuste geométrico com projeção do nó sobre a falha.

Quando um nó troca de tramo, significa que o sistema teve as suas condições de contorno alteradas em tempo de execução do algoritmo de relaxação dinâmica. Abaixo é descrito de forma sucinta o algoritmo que calcula os deslocamentos dos nós governados pela geometria de uma falha lítrica.

□ **Para cada nó localizado sobre a falha**

- ***Consulta grau de liberdade***

- *Obtém ângulo  $\theta$*

- ***rotação das forças desequilibradas e das velocidades para sistema local***

- $[f] = [R(\theta)][F]$

- $[v] = [R(\theta)][V] = \begin{bmatrix} v_1 \\ 0 \end{bmatrix}$

- ***obtenção das novas velocidades (4.19) em  $it = n+1$***

- $[V]^{n+1} = \frac{(1/\Delta t - \alpha / 2)}{(1/\Delta t + \alpha / 2)} [v]^n + \frac{R_i(u_i^n)}{m_{ii}(1/\Delta t + \alpha / 2)}$

- ***cálculo dos deslocamentos na iteração  $n+1$  (sistema local)***

- $[\Delta u] = dt [v]$

- ***voltando velocidades para o sistema global***

- $[V] = [R(-\theta)][v]$

- ***deslocamentos na iteração  $n+1$  no sistema global***

- $[\Delta U] = [R(-\theta)][\Delta u]$

- ***acumulando variação de deslocamentos da iteração  $n+1$  no deslocamento total***

- $[U] = [U] + [\Delta U]$

- ***atualização das restrições***

- chama o algoritmo geométrico descrito anteriormente

- ***incremento de passo***

- $n = n + 1$

Numa dada iteração  $n+1$  são obtidas as forças desequilibradas do nó e convertidas para o seu sistema de coordenadas locais, definido pela consulta às suas restrições de deslocamento assim como as velocidades definidas na iteração anterior.  $n$ . Através de interpolações por diferenças finitas centrais (expressões (4.38) e (4.39)) são obtidas as velocidades na iteração  $n+1$  para o sistema de coordenadas locais. Das velocidades são obtidos os deslocamentos da iteração corrente, que após ser transformado para o sistema de coordenadas globais é acumulado no vetor de deslocamentos total, ou seja, que contém acumulados os deslocamentos desde o início da análise e que informa efetivamente o quanto o nó se deslocou. Após esse procedimento é possível, através do algoritmo de ajuste geométrico descrito, precisar em qual tramo da falha o nó se encontra, atualizando as suas restrições e iniciando uma nova iteração até que o equilíbrio seja obtido.

Para que se possa considerar o módulo em equilíbrio é necessário que as componentes das forças desequilibradas em cada nó da malha seja nula. Nos nós submetidos a restrição sobre a falha, essas forças devem estar zeradas na direção tangencial do sistema de coordenadas locais do nó, que é definida por sua vez pela angulação  $\theta$  do tramo onde o nó se encontra.

## ***5.4. O Sistema Integrado***

Um objetivo deste trabalho é obter um ambiente de balanceamento de seções geológicas totalmente integrado. Em outras palavras, o que se deseja é implementar, dentro de um ambiente único, o *Recon*, a preparação dos dados a serem enviados para o módulo de análise, o processamento do algoritmo de Relaxação Dinâmica com base nesses dados, e a visualização das deformações e dos deslocamentos finalmente obtidos.

### ***5.4.1. O Pré-Processador***

Fazer o pré-processamento significa organizar todos os dados necessários para que o programa de análise possa ser executado. No caso, o *Relax* recebe os seus dados de



entrada por intermédio de um arquivo texto que apresenta um formato padrão, o arquivo neutro. Este formato, por trabalhar com rótulos associados aos diversos tipos de dados, permitiu que fossem exportadas pelo *Recon* somente as informações necessárias para o tipo de análise proposto.

Para que o *Recon* possa fornecer ao módulo de análise essas informações foram necessárias diversas implementações. Nas seções anteriores foi descrita a hierarquia de classes dos atributos geológicos que estão atrelados a estrutura de dados do sistema. Foi também documentada a criação de novas classes, métodos e atributos pendurados a essas classes. Essas alterações foram necessárias justamente para armazenar e estruturar as informações que são consultadas pelas rotinas que, de dentro do ambiente *Recon*, geram o arquivo neutro.

Essas informações, encontradas no arquivo neutro, e que o sistema *Recon* exporta para o módulo de análise foram classificadas em cinco categorias diferentes. A seguir são apresentados para cada uma dessas categorias, os procedimentos computacionais necessários para a geração do arquivo neutro.

#### ***5.4.1.1. Atributos globais da análise***

Os atributos considerados globais da análise, ou seja, que não estão diretamente relacionados a geometria não são pendurados na estrutura de dados do programa. Esses dados são pedidos ao usuário sempre antes de ser executado o módulo de análise e são eles:

- Tipo de análise
- Tolerância para convergência
- Número máximo de iterações
- Intervalo de tempo da análise dinâmica
- Parâmetros associados a matriz de amortecimento

Para consultar esses dados ao usuário foram criados diálogos específicos para a captura dessas informações.

#### **5.4.1.2. Atributos do meio contínuo**

Existem ainda atributos que são definidos no contínuo discretizado. No caso, para cada região, face topológica HED, foram pendurados outros atributos conforme descrito na Seção 5.2. Esses atributos estão associados às propriedades dos diferentes materiais existentes em uma seção, no caso, as camadas referentes as eras geológicas e podem ser subdivididos em três categorias distintas:

- *propriedades físicas*: módulo de elasticidade e coeficiente de Poisson
- *comportamento do material*: linear-elástico, elasto-plástico, etc.
- *propriedades geométricas*: espessura do material
- *parâmetros da análise*: ordem de integração

Para cada uma dessas categorias é possível definir diferentes propriedades, associá-las a rótulos que, por sua vez, são associados a diferentes regiões do modelo. A forma como são feitas essas atribuições é explicada na seção seguinte.

Para entrar com esses dados foram criados novos campos nos diálogos já existentes para a definição das propriedades das camadas.

#### **5.4.1.3. A geometria do modelo**

Como o *Relax* trabalha integrado ao método dos elementos finitos, foi incorporado ao sistema *Recon* uma biblioteca de geração de malhas bidimensionais que apresenta os elementos dos tipos triangular e quadrilateral, com interpolação linear e quadrática (*T3*, *T6*, *Q4* e *Q8*). É fornecida à biblioteca uma lista de *loops*, sendo o primeiro deles referente ao contorno da região a ser discretizada e os demais representando poligonais fechadas, internas ao *loop* externo. Cada *loop* é representado por uma lista de pares de

coordenadas que, por sua vez, representam os pontos que subdividem os *loops*. O algoritmo de geração de malhas é baseado na combinação de duas técnicas: avanço de fronteira e decomposição hierárquica de domínio. A técnica de avanço de fronteira é tradicional em vários aspectos que são encontrados em outros algoritmos da literatura. Entretanto, o algoritmo utilizado inclui alguns passos adicionais que propiciam geração de malhas de boa qualidade (no que se refere à forma dos triângulos), mesmo para problemas com geometria complicada (incluindo furos ou fraturas), característica comum em seções geológicas, e para problemas onde o tamanho dos elementos tem uma variação muito grande de tamanho, isto é, o algoritmo faz uma boa transição entre um trecho da malha com pequenos elementos e outro trecho com grandes elementos. A decomposição hierárquica do domínio é feita antes da geração de malha pela técnica de avanço de fronteira e utiliza um estrutura de dados de árvore quaternária (*quadtree*). Esta estrutura auxiliar tem a finalidade de definir uma função de espaçamento nodal, isto é, ela define a distribuição do tamanho dos elementos a serem gerados dentro do domínio do modelo [43].

Dependendo do espaçamento entre esses pontos, a malha gerada será ou não de boa qualidade, em outras palavras, a malha apresentará elementos mais uniformes quanto a tamanho e geometria.

O elemento utilizado para discretizar as faces que compõem o módulo de trabalho é o *T3* (triangular linear), com um ponto de integração. A biblioteca gera a malha de acordo com a geometria da fronteira da região e com o número de subdivisões de cada uma das arestas que formam esse contorno. Foi implementado dentro da nova classe, *GeoEdge*, Seção 5.2, métodos responsáveis pela subdivisão das arestas das faces. A figura 5.17 mostra uma face composta por duas arestas, a subdivisão dessas arestas e a geração da malha triangular baseada na geometria e na subdivisão das arestas que compõem a face discretizada.

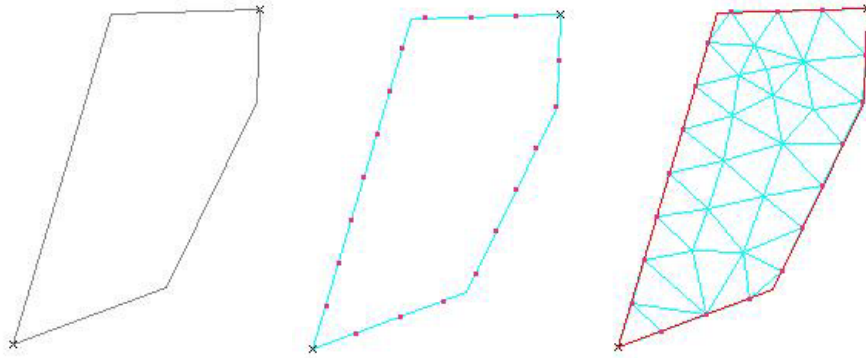
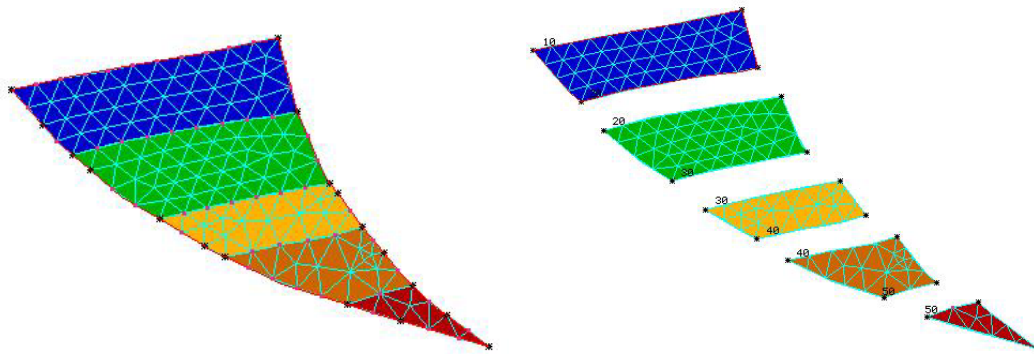


Figura 5.17 – Recon: Geração da malha de Elementos Finitos.

A biblioteca retorna, além da geometria da malha, informações de conectividade entre nós e elementos, importantes para que o módulo de análise e o sistema *Recon* referenciem os dados coerentemente.

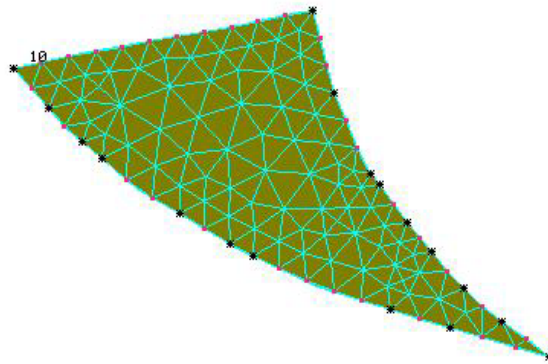
A forma como os atributos são armazenados na estrutura de dados, somado a maneira como é implementado o gerenciamento do módulo de trabalho, causa uma certa dificuldade na hora de exportar essas malhas, que compõem um módulo, para o *Relax*, pois o programa de análise recebe apenas uma malha de elementos finitos, enquanto que um módulo de trabalho do *Recon* pode apresentar várias malhas distintas.

Conforme observado na descrição das classes *GeoFace* e *Layer* (Seções 5.2.1 e 5.2.3), uma face tem pendurada a si, entre outros atributos, a malha de elementos finitos e um apontador para o módulo ao qual esta face pertence. Portanto, o módulo de trabalho obrigatoriamente apresentará mais de uma malha desde que esse apresente igualmente mais de uma camada, como ocorre por exemplo no módulo da *figura 5.18*. A geração da malha, inicialmente individual para cada face que compõe o módulo, apesar de determinar posteriormente a união dessas em uma única malha, se faz necessária já que para cada camada teremos diferentes materiais e conseqüentemente propriedades específicas associadas. Com isso é possível exportar para o *Relax* uma malha com regiões distintas, identificadas pelas diferentes propriedades dos materiais que as definem.



*Figura 5.18 – Recon: Malhas penduradas como atributo de face.*

Para se obter a malha unificada, Figura 5.19, que representa a subdivisão do bloco deformado, é necessário percorrer a lista de faces do módulo e, para cada face, ir construindo a conectividade da nova malha. O objetivo principal é eliminar a duplicidade dos nós de fronteira entre as malhas de duas faces com arestas comuns. Para se identificar esses nós, são necessários testes geométricos.



*Figura 5.19 – Recon: Malha unificada do módulo.*

Criou-se um sólido temporário HED (Figura 5.20) cujos vértices reproduzem os nós da malha e as faces seus elementos, no caso *T3*. Assim, para cada face do módulo, percorre-se os nós da sua respectiva malha e faz-se o teste geométrico com o sólido temporário, construído dinamicamente enquanto todos os nós das malhas são varridos. Para cada nó não encontrado no sólido temporário, lhe é inserido um novo vértice, sendo pendurados

como atributo um novo identificador. Como esses nós das malhas são percorridos elemento a elemento, ao final é possível obter o vetor de conectividades da malha única. Finalmente, para se montar o vetor de coordenadas da malha é bastante percorrer a lista de vértices do sólido temporário, consultar as suas coordenadas e com o seu identificador, pendurado como atributo de vértice, obter a sua posição no vetor de coordenadas da malha.

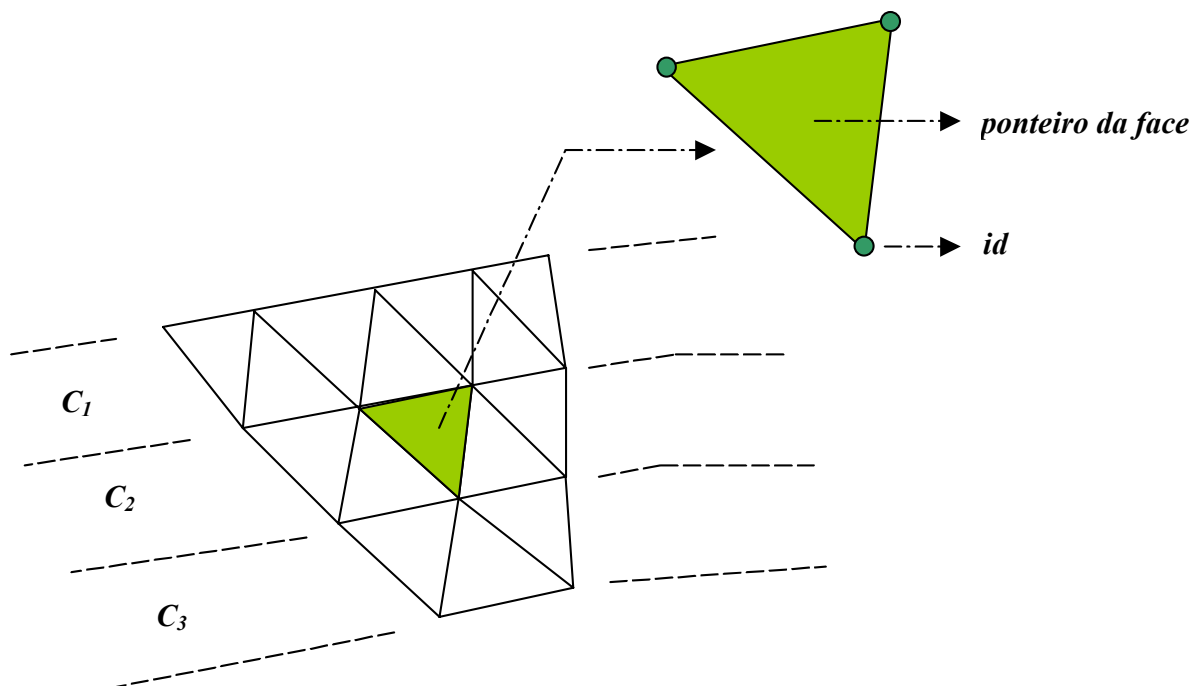


Figura 5.20 – Recon: Sólido temporário HED para armazenar a malha unificada.

Além da lista de nós com as respectivas coordenadas, já obtida através do procedimento acima descrito, também é gerada a lista de elementos contendo, além da conectividade dos nós de cada elemento, os identificadores referentes as propriedades do material, apresentados na Seção 5.4.1.2. O vetor de conectividades entre os elementos da malha unificada já foi montado, mas para exportar a conectividade da malha, é necessário associar para cada elemento da malha os seus atributos. Para tal, usa-se o mesmo sólido temporário utilizado para estruturar os dados da malha unificada. Percorre-se o vetor de conectividades e para cada elemento da malha, é obtida a face do sólido temporário correspondente, consulta-se o seu atributo, no caso o ponteiro para a face do modelo, da

seção geológica onde, por sua vez, encontram-se os atributos de camada que especificam as propriedades de material.

#### **5.4.1.4. Restrições ao Deslocamento**

É necessário informar ao programa de análise a lista de nós da malha unificada que está submetido a restrição de deslocamento sobre a falha. A falha sobre a qual haverá o deslizamento do bloco é selecionada do modelo pelo usuário através de interação com o *mouse*. Para identificar quais são esses nós submetidos à restrição de deslocamento sobre a falha são feitos testes geométricos e novamente utiliza-se o HED para gerenciar esses testes, criando-se mais um sólido temporário para representar essa falha coletada. Basta a seguir percorrer a lista de nós da malha unificada e armazenar os nós que se encontram sobre a falha. Além disso, também é exportada a geometria da falha propriamente dita.

#### **5.4.1.5. Carregamento**

Os tipos de carregamentos previstos para se fazer a análise pelo algoritmo de relaxação dinâmica são ou a aplicação de um campo de deslocamentos prescritos ou a ação da gravidade, que, no caso, atuaria igualmente em todos os nós da malha única.

O campo de deslocamentos prescritos pode ser aplicado a todo o bordo superior do módulo, definido na coleta dos dados para a análise ou aplicado simplesmente no nó que pertence ao bordo superior submetido à restrição de deslocamentos da falha. A escolha de qual das duas opções será utilizada na análise é obtida pelo usuário.

Para se definir e exportar o campo de deslocamentos aplicado a todo o bordo superior são coletadas a geometria origem (o bordo superior) e a geometria destino, tudo via *mouse*. O campo é uniforme, ou seja, todos os vetores que definem os deslocamentos de cada nó individualmente apresentam o mesmo vetor unitário, que por sua vez é definido pela diferença entre o primeiro ponto da poligonal da geometria destino e o nó de interseção entre a falha e o bordo superior, chamado de vetor  $u$  na Figura 5.21.

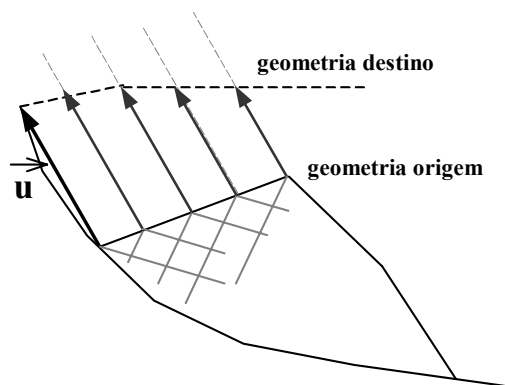


Figura 5.21 – Recon: Definição do campo de deslocamentos prescritos.

Caso seja definido a aplicação de deslocamento prescrito apenas ao nó da falha, este já é o próprio vetor  $u$ .

Com as modificações e rotinas até aqui apresentadas e implementadas no *Recon*, ficam disponibilizados todos esses dados, no caso tanto geometria quanto os atributos.

#### 5.4.2. O Pós-Processador

O objetivo final da integração entre o sistema de balanceamento e o programa de análise é visualizar a nova geometria do bloco conjuntamente com o resto da seção. Além disso também se deseja obter as deformações no módulo após a convergência do algoritmo de relaxação dinâmica. Ao término do processamento, mesmo que o algoritmo não tenha obtido uma nova configuração de equilíbrio, o *Relax* gera um arquivo de pós-processamento contendo, entre outras informações, as novas coordenadas da malha unificada. Esse arquivo apresenta formato similar ao arquivo de entrada de dados, ou seja, o arquivo neutro.

Para que seja possível desenhar dentro do ambiente *Recon* a nova geometria do bloco, é necessário transformar a lista de vértices e arestas que compõem a topologia do bloco. Mais que isso, é fundamental preservar essa topologia após obtidas as novas coordenadas de seus vértices. Ocorre porém que o programa de análise recebe, processa e retorna apenas as novas coordenadas dos nós da malha, conforme já mencionado. A questão



passa a ser então como fazer o mapeamento entre a geometria da nova malha obtida pelo algoritmo de relaxação dinâmica e os vértices e arestas que compõem a lista de faces do módulo de trabalho.

Na realidade esse mapeamento é feito ainda na etapa de pré-processamento, após a geração da malha unificada. O que é feito em tempo de pós-processamento é apenas consultar uma lista que associa vértices e arestas do modelo com a topologia da malha unificada, ou seja, o mapeamento, e com o seu auxílio, partindo-se da nova geometria da malha unificada, reconstruir a lista de faces HED que definem o módulo.

#### 5.4.2.1. Definição da Nova Geometria

O mapeamento é composto de duas listas distintas, uma que associa os vértices topológicos do módulo aos identificadores da malha unificada e outro que associa cada aresta do módulo, a uma lista de identificadores dos nós da malha que estão sobre essa aresta. Ao percorrer a lista de nós da malha unificada define-se, para cada nó, através de testes geométricos se este é ou não referente a um vértice topológico do módulo. Caso seja, são armazenados na lista de vértices o seu ponteiro bem como o seu respectivo nó identificador da malha unificada. Caso o nó da malha não corresponda a um vértice topológico, mas corresponda a uma aresta, então este nó é inserido na lista de identificadores associada a essa aresta. A construção dessa lista, no entanto, requer certos cuidados adicionais. No caso, os nós da malha que são internos das arestas não apresentam correspondência alguma com os vértices não topológicos das arestas, ou seja os vértices internos que definem a geometria da aresta HED, conforme pode ser observado na Figura 5.22 abaixo



Figura 5.22 – Recon – Aresta HED e as suas subdivisões.

Esses nós na realidade são justamente as subdivisões das arestas definidas para a geração da malha. Quando é feito o mapeamento, esses nós passam a ser os novos atributos geométricos das arestas transformadas, o que evidentemente altera a sua geometria. Essa alteração será tanto mais grosseira quanto menos refinada for a subdivisão das arestas. A Figura 5.23 abaixo ilustra o que ocorre com a geometria da malha, quando é redefinida a descrição geométrica de acordo com as suas respectivas subdivisões. As novas arestas apresentam o número de vértices geométricos igual ao número de subdivisões da aresta original acrescido de um.

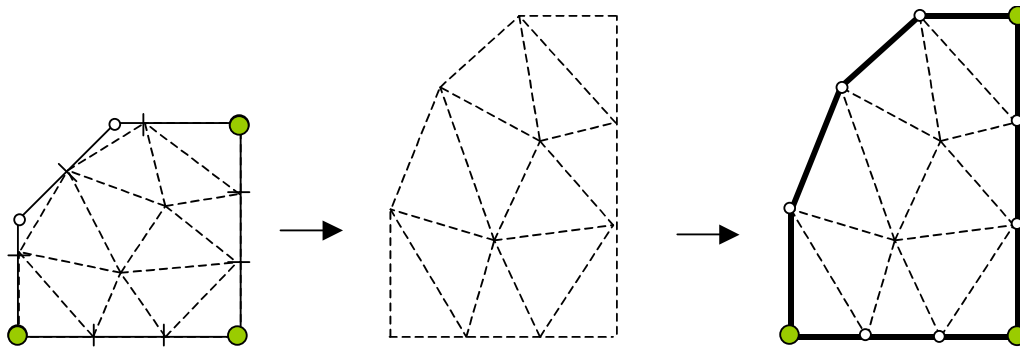


Figura 5.23 – Recon: Mapeamento malha-modelo.

Com o mapeamento montado, é possível ao percorrer as arestas e vértices do módulo obter as novas coordenadas de todos os vértices geométricos que o compõem e com isso remontar a sua geometria, baseada no mapeamento e na malha unificada já com as suas coordenadas atualizadas. A Figura 5.24 mostra como funciona o mapeamento e a reconstrução da aresta  $e_l$ . Na tabela que destaca a aresta  $x$  da lista de arestas mapeadas, pode-se observar que os identificadores estão ordenados de acordo com a topologia original da aresta. Isso é fundamental para que se possa reconstruir corretamente as arestas. Abaixo é descrito o algoritmo que implementa esse mapeamento.

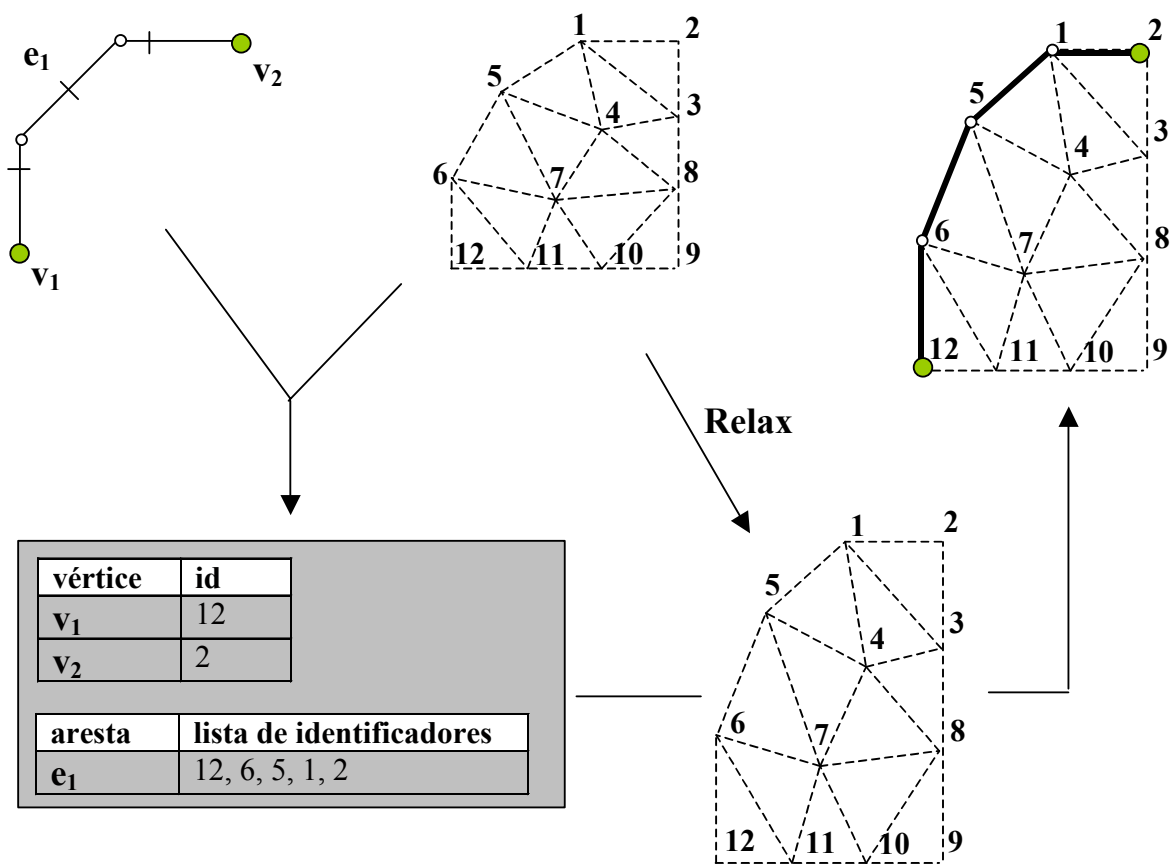


Figura 5.24 – Recon: Resumo da comunicação entre Recon e Relax.

#### 5.4.2.2. Cálculo e Visualização das Deformações Internas

É possível com as ferramentas presentes no Recon atualmente visualizar as deformações introduzidas pelas transformações geométricas existentes no Recon, independentemente da inserção do módulo de análise. O objetivo disso é prover, dentro do ambiente Recon, uma ferramenta para comparação dos resultados obtidos entre os modelos empírico e não empírico.

A idéia é calcular a deformação dos elementos triangulares das faces do módulo deformado utilizando-se do modelo CST (*Constant Strain Triangle*) [19] e, obter a deformação constante de cada elemento triangular. Para suavizar a visualização do campo de deformações introduzido pela transformação, desde que não seja de corpo rígido, é

aferida, para cada nó, a média das deformações dos elementos que contém este nó e a partir de então obter os valores de deformação para cada nó da malha.

A princípio estamos interessados em obter, para um triângulo qualquer, as suas deformações. Considerando deformações infinitesimais com referencial na configuração de origem do elemento (formulação eulleriana) e em notação indicial [4], estes valores podem ser obtidos através da expressão:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} - u_{k,i} u_{k,j}) \quad (5.6)$$

As deformações  $\varepsilon_x$ ,  $\varepsilon_y$  e  $\gamma_{xy}$ , que representam respectivamente as deformações nas direções  $x$ ,  $y$  e a distorção podem portanto ser obtidas de (5.6), gerando as expressões (5.7), (5.8) e (5.9) abaixo.

$$\varepsilon_x = \frac{\partial u}{\partial x} - \frac{1}{2} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 \right] \quad (5.7)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} - \frac{1}{2} \left[ \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 \right] \quad (5.8)$$

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} - \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} \quad (5.9)$$

Onde  $u$  e  $v$  são os deslocamentos expressos de acordo com as funções de forma expressões abaixo e que definem os deslocamentos nodais da Figura 5.25:

$$\begin{cases} u(x, y) = N_1(r, s)u_1 + N_2(r, s)u_2 + N_3(r, s)u_3 \\ v(x, y) = N_1(r, s)v_1 + N_2(r, s)v_2 + N_3(r, s)v_3 \end{cases} \quad (5.9)$$

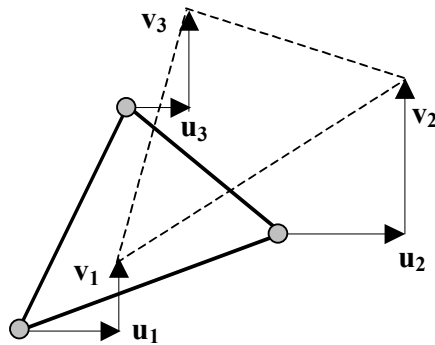


Figura 5.25 – Recon: Cálculo das Deformações No Elemento T3.

Interpolando os deslocamentos nodais e adotando como sistema de coordenadas locais as coordenadas de área do triângulo  $(r,s,t)$ , pode-se obter as seguintes funções de forma:

$$\begin{aligned}
 N_1(r,s) &= r \\
 N_2(r,s) &= s \\
 N_3(r,s) &= 1 - r - s
 \end{aligned}
 \tag{5.10}$$

Com isso, a expressão para a obtenção das deformações em forma matricial é definida abaixo:

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = [B] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}
 \tag{5.11}$$

onde a matriz  $[B]$ , que relaciona deslocamentos com deformações é expressa por:

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix} \quad (5.12)$$

Como as funções de forma são expressas em suas coordenadas locais, não há como obter diretamente as derivadas das funções de forma presentes nos termos da matriz  $\mathbf{B}$ . Para tal, é necessário mapear os termos expressos nas derivadas em relação as coordenadas globais da matriz  $[B]$  para as coordenadas locais:

$$\begin{Bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} \\ \frac{\partial N_1}{\partial s} & \frac{\partial N_2}{\partial s} & \frac{\partial N_3}{\partial s} \end{Bmatrix} \quad (5.13)$$

onde  $[J]^{-1}$  representa a inversa da matriz *Jacobiana*. Ambas as expressões, a matriz *Jacobiana* e a sua inversa, são representadas nas equações (5.14) e (5.15):

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \quad (5.14)$$

$$[J]^{-1} = \frac{1}{\Delta} \begin{bmatrix} \frac{\partial y}{\partial s} & -\frac{\partial x}{\partial s} \\ -\frac{\partial x}{\partial s} & \frac{\partial y}{\partial r} \end{bmatrix} \quad (5.15)$$

onde  $\Delta$  é o determinante da matriz *Jacobiana*. Por outro lado, é possível expressar as coordenadas de um nó através das funções de forma, obtendo as expressões (5.16) abaixo:

$$\begin{aligned}
x &= N_1(r,s)x_1 + N_2(r,s)x_2 + N_3(r,s)x_3 \\
y &= N_1(r,s)y_1 + N_2(r,s)y_2 + N_3(r,s)y_3
\end{aligned}
\tag{5.16}$$

que substituídas em (5.13) e (5.14) determinam (5.17) e (5.18):

$$[J] = \begin{bmatrix} (x_1 - x_3) & (y_1 - y_3) \\ (x_2 - x_3) & (y_2 - y_3) \end{bmatrix}
\tag{5.17}$$

$$[J]^{-1} = \frac{1}{(x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)} \begin{bmatrix} (y_2 - y_3) & (y_3 - y_1) \\ (x_3 - x_2) & (x_1 - x_3) \end{bmatrix}
\tag{5.18}$$

Assim é possível obter os termos da matriz  $[B]$  expressos na equação (5.11):

$$\begin{Bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} \end{Bmatrix} = \frac{1}{\Delta} \begin{Bmatrix} (y_2 - y_3) & (y_3 - y_1) & (y_1 - y_2) \\ (x_3 - x_1) & (x_1 - x_3) & (x_2 - x_3) \end{Bmatrix}
\tag{5.19}$$

Basta agora substituir os termos obtidos matricialmente em (5.18) na equação (5.10) obtendo as deformações  $\varepsilon_x$ ,  $\varepsilon_y$  e  $\gamma_{xy}$ :

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} (y_2 - y_3) & 0 & (y_3 - y_1) & 0 & (y_1 - y_2) & 0 \\ 0 & (x_3 - x_2) & 0 & (x_1 - x_3) & 0 & (x_2 - x_3) \\ (x_3 - x_2) & (y_3 - y_2) & (x_1 - x_3) & (y_3 - y_1) & (x_2 - x_3) & (y_1 - y_2) \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}
\tag{5.20}$$

Uma vez obtidas as deformações  $\varepsilon_x$ ,  $\varepsilon_y$  e  $\gamma_{xy}$  pode-se obter as deformações nas direções principais conforme ilustra a Figura 5.26 e as expressões (5.18) a (5.20):

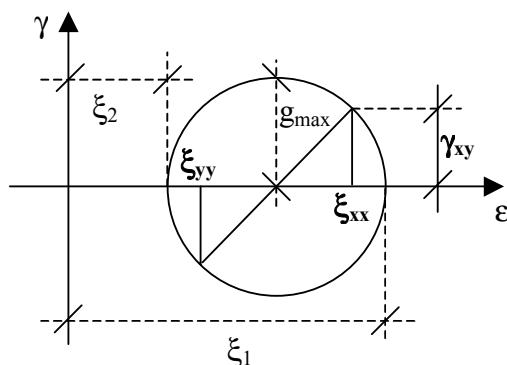


Figura 5.26 – Círculo de Mohr para as direções principais de deformação.

$$g_{\max} = \sqrt{\frac{(\varepsilon_{xx} - \varepsilon_{yy})^2}{4} + \gamma_{xy}^2} \quad (5.21)$$

$$\varepsilon_1 = \frac{\varepsilon_{xx} + \varepsilon_{yy}}{2} + g_{\max} \quad (5.22)$$

$$\varepsilon_2 = \frac{\varepsilon_{xx} + \varepsilon_{yy}}{2} - g_{\max} \quad (5.23)$$

Figura 5.27 – Mudança de base para descontar deslocamentos de corpo rígido.

Podem portanto ser visualizadas as seguintes deformações:

- $\varepsilon_{xx}$ : campo de deformações no eixo x;
- $\varepsilon_{yy}$ : campo de deformações no eixo y;
- $\gamma_{xy}$ : campo de distorções no plano xy;
- $\varepsilon_1$ : direção principal máxima de deformação;
- $\varepsilon_2$ : direção principal mínima de deformação;
- $g_{\max}$ : distorção máxima.

Além disso é possível também visualizar o campo das direções de deformação mínima e máxima do bloco transformado.



A visualização das deformações nodais é feita através de mapas de cores, na realidade são iso-faixas. Para tal lançou-se mão de uma biblioteca, *Contour2D* [41] que implementa um algoritmo para a definição de iso-faixas e iso-linhas com base nos valores associados a malhas de uma forma geral. Cada iso-faixa, representa uma faixa de intensidade de deformação.

Concluindo, independentemente do procedimento adotado para se fazer as transformações no *Recon*, é possível obter o mapa de deformações das seções.

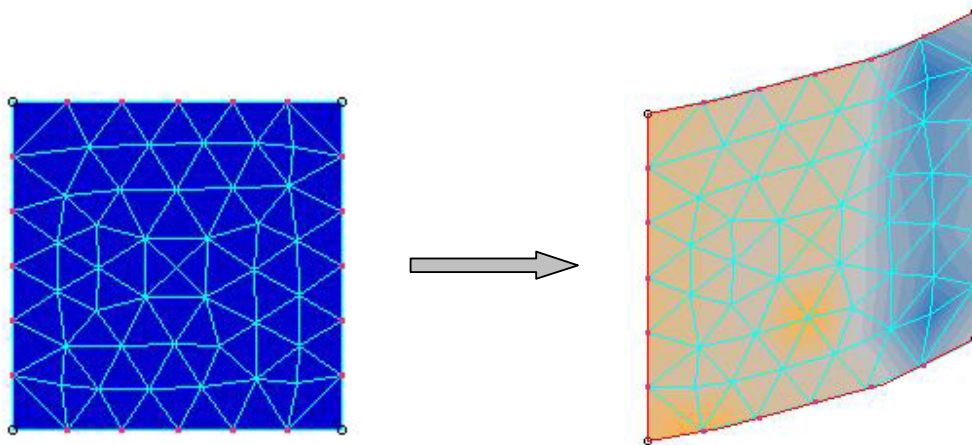


Figura 5.27 – *Recon*: Mapa de cores ilustrando as deformações.

### 5.4.3. Organização das Classes

Com a incorporação de um programa de análise o sistema *Recon* passa a apresentar uma outra categoria de transformações, desta feita associadas a um módulo de análise e não a algoritmos puramente geométricos. No caso das transformações geométricas o procedimento para se aplicá-las é mais simples após a montagem da matriz que rege a transformação, sendo bastante percorrer os vértices e arestas do módulo e obter as suas novas coordenadas através do produto entre os vértices topológicos e não topológicos e a matriz correspondente a transformação corrente. Conforme pôde ser observado na Seção 5.4.2, o mesmo não ocorre com transformações obtidas por métodos numéricos, como o método dos elementos finitos, por exemplo, já que esses algoritmos retornam uma nova

geometria para a malha. Daí a necessidade de se implementar o mapeamento entre malha e modelo anteriormente descrito.

No Capítulo 3 foi descrita a hierarquia de classes que gerenciam as transformações geométricas originais do sistema *Recon*. Para se incorporar esta nova filosofia de transformações foi alterada a hierarquia dessas classes conforme pode-se observar na Figura 5.29.

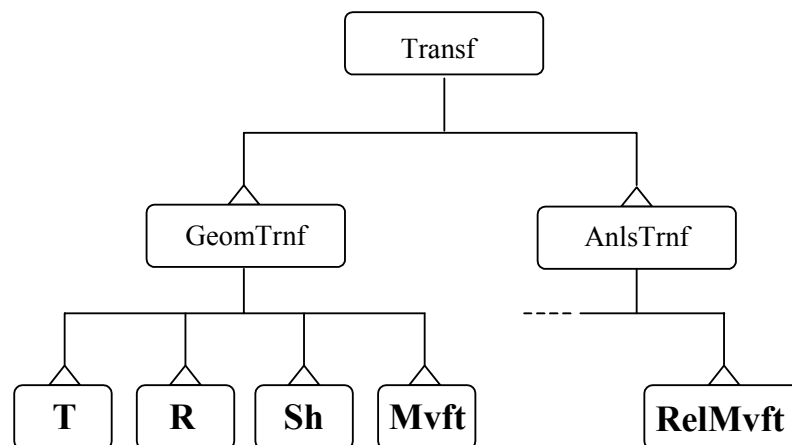


Figura 5.29 – *Recon*: Hierarquia das classes de transformações.

A superclasse *Transf* passou a ser mais genérica e agora é uma classe base para as classes *GeomTrnf* (classe base para as transformações geométricas) e *AnlsTrnf* (classe básica para as transformações providas por métodos numéricos).

Das classes *GeomTrnf* e *AnlsTrnf* é que são derivadas as classes que efetivamente implementam as transformações presentes no *Recon*. Assim a superclasse *Transf* apresenta os métodos comuns a todas as transformações e as suas subclasses servem de base para os dois tipos genéricos de transformações que o sistema passou a dispor.

A principal vantagem que se obtém com essa disciplina de programação orientada a objetos expressa na hierarquia de classes acima apresentada está na facilidade de se incorporar futuramente ao sistema outros tipos de análises numéricas, bastando para isso criar uma nova classe, derivada de *AnlsTrnf*

A Figura 5.30 resume a construção da classe *AnlsTrnf*. As classes derivadas de *AnlsTrnf* herdam todos os seus métodos e atributos, como é o caso da subclasse *RelMvft*, que implementa os métodos responsáveis pela simulação da movimentação de um bloco sobre uma falha lítrica utilizando-se um algoritmo de relaxação dinâmica acoplado ao método dos elementos finitos. Às classes derivadas ainda é permitida a criação de novos métodos e atributos peculiares a uma determinada análise.

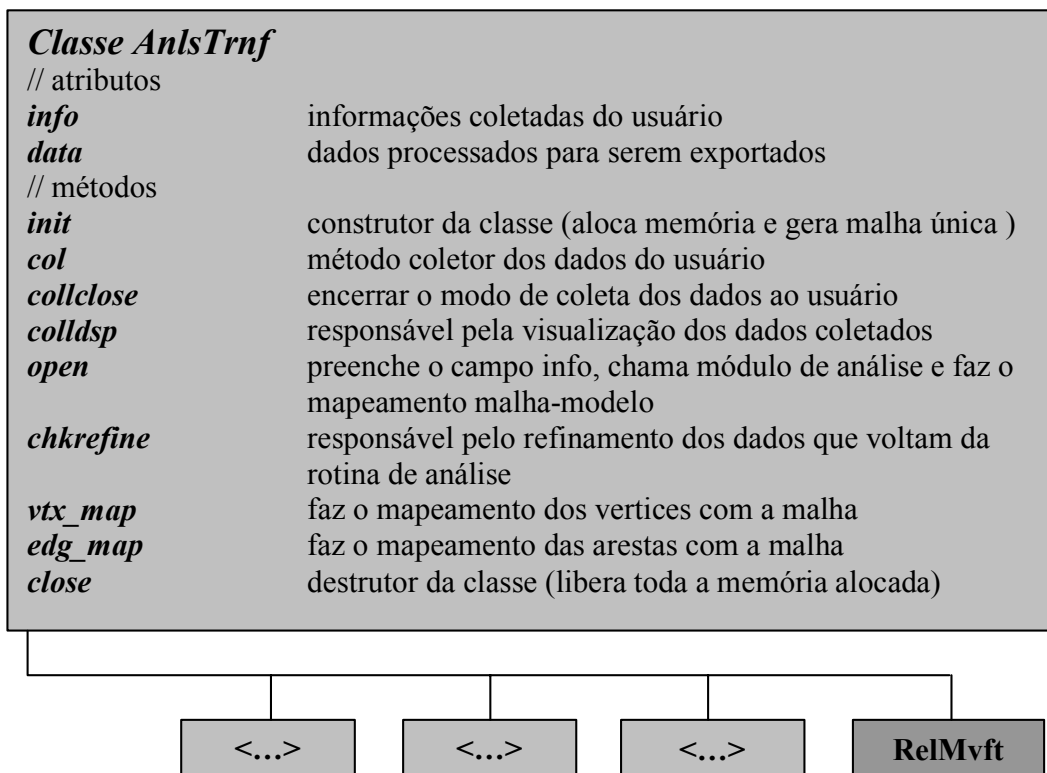


Figura 5.30 – Recon: Definição da classe *AnlsTrnf*.

O atributo *info* armazena as informações coletadas do usuário necessárias para definir o processamento como por exemplo a falha que definirá as restrições ao deslocamento e as poligonais que definem as geometrias origem e destino. O atributo *data* por sua vez armazena os dados processados pelo método *open*. Este método chama as rotinas que calculam a malha unificada, o campo de deslocamentos prescritos e monta todo o mapeamento que será utilizado além de efetivamente gerar o arquivo neutro e chamar as funções da biblioteca *Relax* para rodar o programa de análise. Existem métodos

responsáveis para checar e refazer a consistência topológica do modelo após a transformação. Os métodos *vtx\_map* e *edg\_map* são os métodos que consultam o mapeamento e obtém as novas coordenadas do modelo baseada na nova geometria da malha retornada pelo *Relax*.

#### **5.4.4. Comunicação Entre o Recon e o Relax**

Com todas as mudanças acima descritas no sistema *Recon*, que passou a exportar um arquivo no formato de leitura do programa de análise é possível transformar o *Recon* e o *Relax* em um único sistema. Para isso, o programa de análise foi incorporado ao sistema *Recon* como uma biblioteca com funções que são executadas de dentro do *Recon*. Os sistema se comunica com a biblioteca via arquivo texto simplesmente para não sobrecarregar a memória RAM da máquina que estiver executando o sistema.

Os resultados da análise feita pelo *Relax* são lidos pelo *Recon*, novamente por intermédio de um arquivo texto, sendo a seguir pós-processados conforme já foi explicado.

Resumindo todo o processo, temos que após a subdivisão das arestas e a geração da malha, é feito o pré-processamento do modelo e exportado o arquivo neutro para o *Relax*, que, após atingir a sua configuração de equilíbrio, fornece uma nova geometria para a malha unificada do módulo. Já com a nova malha, são feitas as consultas ao mapeamento definido ainda no pré-processamento, quando é obtida a nova geometria da lista de faces que compõe o módulo de trabalho. Os resultados provenientes do programa de análise são então guardadas na estrutura de dados do sistema que passa a disponibilizar o mapa de cores com as deformações do módulo, bem como as barras com as direções dessas deformações.