



## ANÁLISE GEOMETRICAMENTE NÃO LINEAR POR MÉTODOS BASEADOS NA PARTIÇÃO DA UNIDADE

**Lorena Leocádio Gomes**

**Felício Bruzzi Barros**

**Roque Luiz da Silva Pitangueira**

**Samuel Silva Penna**

lorenaleocadio@gmail.com

felicio@dees.ufmg.br

roque@dees.ufmg.br

spenna@dees.ufmg.br

Programa de Pós-Graduação em Engenharia de Estruturas - Universidade Federal de Minas Gerais (UFMG)

Av. Antônio Carlos, 6627 - Escola de Engenharia - Bloco 1 - 4o andar, sala 4215, Pampulha, CEP 31270-901, MG, Belo Horizonte, Brasil

**Resumo.** *Este artigo apresenta um projeto de implementação computacional para a solução de problemas com não linearidade geométrica por meio do Método dos Elementos Finitos Generalizados (MEFG), um método que pode ser considerado como uma instância do Método da Partição da Unidade (MPU). Na análise com não linearidade geométrica, tem-se uma significativa distorção da malha de elementos devido aos efeitos de grandes deslocamentos e deformações considerados no cálculo estrutural, que penalizam a aproximação da solução feita a partir do MEF. No entanto, constata-se que o MEFG é menos propenso a ser influenciado por esta distorção na malha, o que o torna mais vantajoso para este tipo de análise. Assim, um ambiente computacional existente e desenvolvido no Departamento de Engenharia de Estruturas (DEES) da UFMG, que possui meios que permitem análises estruturais lineares e não lineares, foi expandido de forma a realizar as análises com não linearidade geométrica através do MEFG. Como forma de validar a implementação desta expansão, os resultados de simulações numéricas, para este tipo de análise, são comparados com resultados encontrados na literatura.*

**Palavras chave:** *Mecânica Computacional, Análise Geometricamente Não Linear, JAVA, Método dos Elementos Finitos Generalizados, Métodos da Partição da Unidade*

## 1 INTRODUÇÃO

A proposta deste trabalho é a expansão da plataforma computacional INSANE, permitindo que sejam realizadas análises estruturais geometricamente não lineares por meio de métodos baseados na partição da unidade, em particular o Método dos Elementos Finitos Generalizados (MEFG).

O Método dos Elementos Finitos Generalizados (MEFG), proposto por Strouboulis, Babuška e Copps (2000) e por Duarte, Babuška e Oden (2000), é uma combinação do Método dos Elementos Finitos (MEF) clássico e do Método da Partição da Unidade (MPU) desenvolvido por Melenk e Babuška (1996). Em uma análise via MEFG, o domínio do problema é dividido em malhas de elementos onde são construídas as funções de aproximação, utilizando para isto, a partição da unidade associada aos pontos nodais. Estas funções, que, usualmente, são as mesmas utilizadas em elementos finitos, são então enriquecidas nos nós de forma análoga ao Método das Nuvens-*hp*, método apresentado em Duarte e Oden (1996), e que é uma das variações do método da Partição da Unidade. Desta maneira, segundo Strouboulis, Babuška e Copps (2000), ao se incluir essas funções de enriquecimento, aumenta-se consideravelmente a precisão das aproximações do MEF clássico, com a adição de somente alguns graus de liberdade.

As não linearidades geométricas, segundo Ribeiro (2009), estão relacionadas às não linearidades da estrutura provenientes de variações na geometria, como mudanças de forma e rotações. E essas variações podem levar a estrutura a ter alterações consideráveis na configuração de equilíbrio. Ao se realizar uma análise estrutural geometricamente não linear, os resultados serão tão mais realistas quanto mais relevantes forem estas variações.

Em uma análise com grandes deslocamentos e deformações via MEF, os elementos gerados na estrutura sofrem grandes distorções, e essas distorções penalizam a qualidade de aproximação da solução de elementos finitos. Alguns trabalhos já realizados demonstram que o MEFG é menos propenso a ser influenciado por esta distorção da malha, tornando-se mais adequado a sua aplicação em problemas com não linearidade geométrica (Barros, 2002), (Alves; Barros; Pitangueira, 2013).

Resolver problemas não lineares, é encontrar a posição de equilíbrio do corpo através da solução de um sistema de equações, iterativamente, até que as forças internas sejam, dentro de um determinado critério, iguais às forças externas. Como o processo de solução do sistema deve ser repetido inúmeras vezes, o uso de ferramentas computacionais adequadas torna a solução deste tipo de análise mais viável e rápida.

A plataforma INSANE é um software livre desenvolvido no Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais (UFMG), implementado em linguagem Java. Em sua versão atual, o INSANE permite a realização de análises com não linearidade física e geométrica através do Método dos Elementos Finitos, esta última implementada por Fonseca (2008) e para análise fisicamente não linear pelo MEFG, implementada por Wolenski et al. (2014).

O objetivo é apresentar a expansão feita na plataforma INSANE, para que as análises via MEFG contemplem a não linearidade geométrica. As ferramentas numéricas, já validadas e disponíveis no INSANE para a solução deste tipo de problema, foram modificadas e utilizadas para uma análise via MEFG, que se mostra vantajosa por sofrer menor influência do efeito negativo da distorção da malha. Assim, este artigo é a primeira parte de um trabalho na qual, procura-

se mostrar e validar esta expansão do MEFG para o caso da formulação Lagrangiana Total. Na sequência do trabalho, será também considerada a formulação Lagrangiana Atualizada e a aferição do impacto de elevadas distorções da malha sobre a discretização.

Como forma de apresentar o trabalho realizado, este artigo é dividido da seguinte maneira: o item 2 é feita uma revisão teórica, apresentando os aspectos relevantes sobre os três principais temas a serem desenvolvidos, MEFG, não linearidade geométrica e o INSANE; no item que se segue (item 3), explica-se de forma clara o trabalho realizado e os exemplos utilizados para sua validação; e ao final, no item 4, tem-se uma discussão dos resultados obtidos.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Método dos Elementos Finitos Generalizados

O Método dos Elementos Finitos Generalizados (MEFG) pode ser entendido como uma extensão do Método dos Elementos Finitos (MEF) tradicional, adicionando funções especiais para a aproximação.

O MEFG possui também características importantes dos chamados métodos sem malha. Nestes métodos, segundo Barros (2002), a solução aproximada do problema é construída sem a necessidade de definição de uma malha de elementos, e com base na distribuição dos pontos nodais.

Dentre as variações dos métodos sem malha, existem os métodos baseados na definição da Partição da Unidade. Para este trabalho, destaca-se o Método das Nuvens-*hp*, que tem origem no Método de Galerkin Livre de Elementos (Belytschko; Black, 1999). A flexibilidade destes métodos, advinda da falta de conectividade entre os nós utilizados para discretizar o domínio conduz, contudo, a problemas relacionados à aplicação das condições de contorno e integração numérica.

No método das Nuvens-*hp*, diferentemente do método Galerkin Livre de Elementos, utiliza-se de uma estratégia de enriquecimento da função de aproximação, em que as funções de forma originais são multiplicadas por funções quaisquer, escolhidas especialmente para o problema.

No MEFG, as aproximações são inicialmente construídas como no MEF, ou seja, a partir de funções identificadas com uma malha de elementos. Estas, contudo, são interpretadas como associadas aos nós e seus domínios de influência, as nuvens (conjunto de elementos que contém um mesmo nó). O MEFG herda do método das Nuvens-*hp*, a estratégia de enriquecimento da aproximação. Como resultado, problemas como a integração numérica e imposição das condições de contorno são superadas como ocorre no MEF, a aproximação ganha uma certa flexibilidade, oriunda do método das Nuvens-*hp*, e torna-se menos propensa a sofrer os efeitos negativos da distorção da malha.

Assim, segundo Alves (2012), o MEFG instaura uma ponte entre os dois diferentes métodos, onde, sobre uma malha de elementos finitos faz-se uso de funções da Partição da Unidade (PU), e que tem o mesmo enriquecimento proposto pelo Método das Nuvens-*hp*.

### Formulação

A característica principal do Método dos Elementos Finitos Generalizados (MEFG) é o uso da Partição da Unidade (PU) em um conjunto de elementos finitos.

A partição da unidade pode ser definida como um conjunto de funções cujos valores somam a unidade em cada ponto  $\mathbf{x}_j$  no domínio, e o problema a ser analisado é que definirá o tipo de função de PU a ser utilizada.

O emprego das funções convencionais de MEF (como as funções Lagrangianas), além de facilitar a aplicação do método, verificam diretamente as condições de contorno, ao contrário do que normalmente ocorre no Método das Nuvens-*hp* (Barros, 2002).

Uma das características herdadas do Método das Nuvens-*hp* pelo MEFG é a discretização do domínio do problema em nuvens de elementos, que são definidas como as uniões de elementos finitos que partilham um mesmo nó no domínio.

Além disso, têm-se as funções de enriquecimento, conhecidas como funções de aproximação local e oriundas das estratégias do Método das Nuvens-*hp*. Estas funções, escolhidas a partir de um conhecimento prévio da solução do problema, devem construir um conjunto de  $q_j$  funções linearmente independentes e são definidas para cada ponto nodal  $\mathbf{x}_j$  como:

$$I_j \stackrel{def}{=} \{L_{j1}(\mathbf{x}), L_{j2}(\mathbf{x}), \dots, L_{jq}(\mathbf{x})\} = \{L_{ji}(\mathbf{x})\}_{i=1}^{q_j}, \text{ com } L_{j1}(\mathbf{x}) = 1 \quad (1)$$

Estas funções, multiplicadas pelas funções de PU, ampliam o espaço de aproximação de elementos finitos, que passa a ter como base um conjunto das funções de aproximação  $\phi_{ji}(\mathbf{x})$ :

$$\{\phi_{ji}\}_{i=1}^{q_j} = N_j(\mathbf{x}) \times \{L_{ji}(\mathbf{x})\}_{i=1}^{q_j}, \text{ sem somatório em } j. \quad (2)$$

Assim, segundo Alves (2012), as funções de forma do MEFG herdam o suporte compacto da Partição da Unidade e as características da aproximação da função de aproximação local. Com isso, uma função genérica para a aproximação do problema (no caso deslocamento) é obtida com a combinação linear das funções de forma de MEF e MEFG:

$$\tilde{\mathbf{u}}(\mathbf{x}) = \sum_{j=1}^N N_j(\mathbf{x}) \left\{ \mathbf{u}_j + \sum_{i=2}^{q_j} L_{ji}(\mathbf{x}) \mathbf{b}_{ji} \right\} \quad (3)$$

onde  $\mathbf{u}_j$  é o parâmetro associado aos graus de liberdade do MEF e  $\mathbf{b}_{ji}$  são os novos graus de liberdade, criados pelos enriquecimentos nodais, associados ao MEFG. Quando a PU, se polinomial, é enriquecida por monômios, o conjunto de funções (2) torna-se linearmente dependente e a matriz de rigidez construída com estas funções torna-se positiva semi-definida. Esta dependência, porém, pode ser solucionada ao se utilizar estratégias numéricas propostas em Strouboulis, Babuška e Capps (2000), como o procedimento de Babuška, conforme denominado em Barros (2002).

## 2.2 Análise Geometricamente Não Linear

Em um problema em que o material seja considerado com comportamento elástico linear, e que a estrutura esteja sujeita a pequenos deslocamentos e deformações, o seguinte sistema de equações lineares, pode ser obtido via discretização por elementos finitos e baseado na formulação de Galerkin:

$$\mathbf{K}\mathbf{u} = \mathbf{F} \quad (4)$$

onde os deslocamentos, representados pela variável  $\mathbf{u}$ , variam linearmente com o acréscimo de esforços na estrutura,  $\mathbf{F}$  e a matriz de rigidez  $\mathbf{K}$  é considerada constante durante toda a aplicação do carregamento.

Em descrições mais realistas do comportamento estrutural, o corpo pode estar submetido a grandes deslocamentos e deformações e a relação constitutiva do material não é linear. Estes comportamentos acabam definindo dois tipos diferentes de análises não lineares na mecânica das estruturas, a análise fisicamente não linear, onde apenas a relação *tensão x deformação* do material é não linear, e a análise geometricamente não linear, que considera a possibilidade de grandes deslocamentos e deformações da estrutura. Para este trabalho, será considerada apenas a análise com não linearidade geométrica.

Em uma análise não linear busca-se encontrar o estado de equilíbrio do corpo correspondente às cargas aplicadas e, para isto acontecer, a seguinte equação deve ser satisfeita:

$$\mathbf{R} - \mathbf{F} = \mathbf{0} \quad (5)$$

onde  $\mathbf{R}$  representa o vetor forças internas aplicadas no corpo e  $\mathbf{F}$  representa as forças externas representada na Eq. (4).

O problema da não linearidade é justamente conseguir resolver o sistema representado na Eq. (5), pois a rigidez da estrutura,  $\mathbf{K}$ , passa a ser uma função dos deslocamentos do corpo, que são as incógnitas a serem determinadas no problema da Eq. (4), criando, desta maneira, uma interdependência desses deslocamentos. Assim, para solucionar este problema de interdependência e conseguir determinar o vetor de deslocamentos, usa-se um processo incremental e iterativo, onde a carga é aplicada gradualmente na estrutura, até se chegar ao equilíbrio do sistema e a relação apresentada pela Eq. (5) se aproximar ao máximo de zero.

Existem dois tipos de formulações que se propõem a descrever o movimento dos sólidos deformáveis, e que podem ser empregadas como formas de solucionar problemas com não linearidade geométrica. São elas a formulação Lagrangiana, também chamada de material, onde as grandezas associadas ao corpo deformado são referidas à configuração original (indeformada), e a formulação Euleriana, ou espacial, em que as grandezas são atribuídas à configuração final (deformada).

De acordo com Fonseca (2008), a formulação Lagrangiana é a mais apropriada para a análise não linear incremental iterativa na mecânica dos sólidos, pois, neste caso, o importante é a trajetória de deformações de cada ponto do sólido durante o processo de carregamento. Já a formulação Euleriana vem sendo mais aplicada em problemas da mecânica dos fluidos, pois a atenção é focada no movimento do material através de um volume de controle estacionário.

A formulação adotada neste trabalho é a Lagrangiana, conforme apresentada por Bathe (1996) e será descrita nos parágrafos a seguir.

Na abordagem da análise incremental Lagrangiana, o equilíbrio do corpo em um pseudo-tempo (chamado simplesmente de tempo)  $t + \Delta t$ , onde  $\Delta t$  refere-se a um intervalo de tempo associado ao passo da aplicação do carregamento, é expresso usando o princípio dos deslocamentos virtuais a partir da seguinte equação:

$$\int_{t+\Delta t \mathbf{V}} {}^{t+\Delta t} \tau_{ij} \delta_{t+\Delta t} e_{ij} d{}^{t+\Delta t} \mathbf{V} = {}^{t+\Delta t} \mathfrak{R} \quad (6)$$

onde o lado esquerdo da equação corresponde ao trabalho virtual interno e  ${}^{t+\Delta t} \mathfrak{R}$  ao trabalho virtual externo, na configuração  $t + \Delta t$ , e ainda,  ${}^{t+\Delta t} \tau_{ij}$  representa as componentes cartesianas do tensor de tensões de Cauchy,  $\delta_{t+\Delta t} e_{ij}$  corresponde ao tensor de deformações correspondente aos deslocamentos virtuais impostos,  ${}^{t+\Delta t} \mathbf{V}$  ao volume do corpo no tempo correspondente e

$\delta$  associa à variável que a ele segue designação de virtual. O subscrito à direita se refere à componente das coordenadas cartesianas.

A expressão para o trabalho virtual externo, é dada por:

$${}^{t+\Delta t}\mathfrak{R} = \int_{{}^{t+\Delta t}\mathbf{V}} {}^{t+\Delta t}f_i^B \delta u_i d{}^{t+\Delta t}\mathbf{V} + \int_{{}^{t+\Delta t}S_f} {}^{t+\Delta t}f_i^S \delta u_i d{}^{t+\Delta t}\mathbf{S} \quad (7)$$

onde  ${}^{t+\Delta t}f_i^B$  representa a componente de forças externas aplicada por unidade de volume no tempo  $t + \Delta t$ ,  ${}^{t+\Delta t}f_i^S$  a componente de força de superfície por unidade de área,  ${}^{t+\Delta t}S_f$  é a superfície no tempo  $t + \Delta t$  no qual as forças de superfície são aplicadas e  $\delta u_i^S$  caracteriza  $\delta u_i$  avaliado na superfície  ${}^{t+\Delta t}S_f$ .

A dificuldade fundamental na aplicação geral da Eq. (6) é que se desconhece a configuração do corpo no tempo  $t + \Delta t$ , e a mudança contínua na configuração do corpo implica em consequências para o desenvolvimento de um procedimento de análise incremental. Na formulação Lagrangiana, por se conhecer a trajetória de deformações, pode-se fazer referência a uma configuração de equilíbrio anterior ao tempo  $t + \Delta t$  para a solução do problema. Existem duas diferentes formulações que possibilitam a escolha desta configuração de equilíbrio, são elas a formulação Lagrangiana Total (LT) e a formulação Lagrangiana Atualizada (LA). Na formulação LT todas as variáveis estáticas e cinemáticas são referidas na configuração inicial (no tempo 0). A formulação LA tem procedimento análogo à LT, porém, com as variáveis referidas à última configuração calculada. As duas formulações levam em consideração os efeitos cinemáticos não lineares devido a grandes deslocamentos e deformações, porém, são as relações constitutivas especificadas de *tensão x deformação* que proporcionam um adequado modelo para o comportamento de grandes deformações. A descrição de Bathe (1996) que se está seguindo é genérica o suficiente para abranger tanto a formulação total quanto a atualizada, e será assim apresentada nesta seção, porém neste trabalho foram feitas análises utilizando somente a formulação Lagrangiana Total.

Assim, para análise incremental, certas medidas de tensão e deformação podem ser empregadas para melhor representar a Eq. (6) nas formas Lagrangianas Total e Atualizada, e essas componentes são: o segundo tensor de Piola-Kirchoff,  $S_{ij}$ , e o tensor de deformações de Green-Lagrange,  $\varepsilon_{ij}$ . Portanto, ao se fazer as substituições pertinentes, a Eq. (6) se transforma em:

$$\int_{{}^0\mathbf{V}} {}^{t+\Delta t}S_{ij} \delta_0^{{}^{t+\Delta t}}\varepsilon_{ij} d^0\mathbf{V} = {}^{t+\Delta t}\mathfrak{R} \quad (8)$$

representando a formulação Lagrangiana Total (LT), e:

$$\int_{{}^t\mathbf{V}} {}^{t+\Delta t}S_{ij} \delta_t^{{}^{t+\Delta t}}\varepsilon_{ij} d^t\mathbf{V} = {}^{t+\Delta t}\mathfrak{R} \quad (9)$$

a formulação Lagrangiana Atualizada (LA).

Na equações 8 e 9, o sobrescrito à esquerda indica a configuração do corpo em que a grandeza representada ocorre, e o subscrito à esquerda, em relação a qual configuração do corpo em que a grandeza é medida.

Usando as relações para a decomposição incremental das tensões e deformações e linearizando as equações, obtém-se as seguintes equações de equilíbrio para as formulações LT e LA, respectivamente:

$$\int_{{}^0\mathbf{V}} {}^0C_{ijrs} e_{rs} \delta_0 e_{ij} d^0\mathbf{V} + \int_{{}^0\mathbf{V}} {}^tS_{ij} \delta_0 \eta_{ij} d^0\mathbf{V} = {}^{t+\Delta t}\mathfrak{R} - \int_{{}^0\mathbf{V}} {}^tS_{ij} \delta_0 e_{ij} d^0\mathbf{V} \quad (10)$$

$$\int_{t\mathbf{V}} {}^t C_{ijrs} {}^t e_{rs} \delta {}^t e_{ij} d^t \mathbf{V} + \int_{t\mathbf{V}} {}^t \tau_{ij} \delta {}^t \eta_{ij} d^t \mathbf{V} = {}^{t+\Delta t} \mathfrak{R} - \int_{t\mathbf{V}} {}^t \tau_{ij} \delta {}^t e_{ij} d^t \mathbf{V} \quad (11)$$

onde  ${}_0 C_{ijrs}$  e  ${}_t C_{ijrs}$  são os tensores incrementais das propriedades dos materiais no tempo  $t$  na configuração  $0$  e  $t$ , respectivamente.  ${}_0 S_{ij}$  representa o segundo tensor de Piola-Kirchoff no tempo  $t$  e  ${}^t \tau_{ij}$  o tensor de tensão de Cauchy no tempo  $t$ .  ${}_0 e_{ij}$ ,  ${}_t e_{ij}$ ,  ${}_0 \eta_{ij}$  e  ${}_t \eta_{ij}$  são as componentes de deformações incrementais que se dividem em componentes lineares e não lineares, referidas ao tempo  $0$  e  $t$ , como representado pelas seguintes equações:

$$\begin{aligned} {}_0^{t+\Delta t} \epsilon_{ij} &= {}_0^t \epsilon_{ij} + {}_0 \epsilon_{ij}; & {}_0 \epsilon_{ij} &= {}_0 e_{ij} + {}_0 \eta_{ij} \\ {}_t^{t+\Delta t} \epsilon_{ij} &= {}_t \epsilon_{ij}; & {}_t \epsilon_{ij} &= {}_t e_{ij} + {}_t \eta_{ij} \end{aligned} \quad (12)$$

onde  ${}_t^{t+\Delta t} \epsilon_{ij}$  e  ${}_0^{t+\Delta t} \epsilon_{ij}$  são as componentes de deformação que se dividem nas componentes lineares:

$$\begin{aligned} {}_0 e_{ij} &= \frac{1}{2} ({}_0 u_{i,j} + {}_0 u_{j,i} + {}_0^t u_{k,i} {}_0 u_{k,j} + {}_0 u_{k,i} {}_0^t u_{k,j}) \\ {}_t e_{ij} &= \frac{1}{2} ({}_t u_{i,j} + {}_t u_{j,i}) \end{aligned} \quad (13)$$

e nas componentes não lineares:

$$\begin{aligned} {}_0 \eta_{ij} &= \frac{1}{2} {}_0 u_{k,i} {}_0 u_{k,j} \\ {}_t \eta_{ij} &= \frac{1}{2} {}_t u_{k,i} {}_t u_{k,j} \end{aligned} \quad (14)$$

As equações 10 e 11 são empregadas no cálculo do incremento de deslocamentos que irão atualizar os valores das aproximações dos deslocamentos, tensões e deformações correspondentes ao tempo  $t + \Delta t$ .

O princípio básico da análise não linear utilizando a formulação paramétrica do MEF, assim como no MEEG, é o mesmo da análise linear. Deve-se obter as funções de interpolação (funções de forma) para o elemento finito desejado e, a partir delas, calculam-se as matrizes necessárias para a análise (Fonseca, 2008).

Discretizando-se o problema via MEF e ou MEEG, a Eq. (10) pode ser substituída pela seguinte equação matricial para a formulação LT:

$$\left( {}_0^t \mathbf{K}_L + {}_0^t \mathbf{K}_{NL} \right) \Delta \mathbf{U}^{(j)} = {}^{t+\Delta t} \mathbf{R} - {}_0^t \mathbf{F}^{(j-1)} \quad (15)$$

Já a Eq. (11) assume a seguinte forma matricial para a formulação LA:

$$\left( {}_t^t \mathbf{K}_L + {}_t^t \mathbf{K}_{NL} \right) \Delta \mathbf{U}^{(j)} = {}^{t+\Delta t} \mathbf{R} - {}_t^t \mathbf{F}^{(j-1)} \quad (16)$$

Onde:

${}_0^t \mathbf{K}_L, {}_t^t \mathbf{K}_L$  = Matrizes de rigidez incrementais de deformações lineares

${}_0^t \mathbf{K}_{NL}, {}_t^t \mathbf{K}_{NL}$  = Matrizes de rigidez incrementais de deformações não lineares

${}^{t+\Delta t} \mathbf{R}$  = Vetor de forças externas aplicadas nos pontos nodais no tempo  $t + \Delta t$

${}_0^t \mathbf{F}, {}_t^t \mathbf{F}$  = Vetores de forças equivalentes do ponto nodal às tensões nos elementos (tempo  $t$ )

$\Delta \mathbf{U}^{(j)}$  = Vetor de incrementos de deslocamentos nodais na  $j$ -ésima iteração:  ${}^t \mathbf{U}^{(j)} = {}^t \mathbf{U}^{(j-1)} + \Delta \mathbf{U}^{(j)}$

Na Tabela 1 são reunidas as equações na forma integral e na forma matricial para ambas as formulações Lagrangianas.

**Tabela 1: Matrizes de Elementos Finitos (Bathe, 1996).**

Tipo de Análise	Integral	Matriz Correspondente
Ambas Formulações	${}^{t+\Delta t}{}_{0S_f} \mathcal{R} = \int_{0S_f} {}^{t+\Delta t}{}_{0f_i} \delta u_i^S \delta u_i^S d^0S + \int_{0V} {}^{t+\Delta t}{}_{0f_i} \delta u_i^B \delta u_i^B d^0V$	${}^{t+\Delta t}{}_{0S_f} \mathbf{R} = \int_{0S_f} \mathbf{H}^{ST} {}^{t+\Delta t}{}_{0f_i} \mathbf{f}^S d^0S$
Formulação Lagrangeana Total	$\int_{0V} {}_0C_{ijrs} {}_0e_{rs} \delta_0 e_{ij} d^0V$ $\int_{0V} {}_0^t S_{ij} \delta_0 \eta_{ij} d^0V$ $\int_{0V} {}_0^t S_{ij} \delta_0 e_{ij} d^0V$	${}_0^t \mathbf{K}_L \hat{\mathbf{u}} = \left( \int_{0V} {}_0^t \mathbf{B}_L^T {}_0 \mathbf{C} {}_0^t \mathbf{B}_L d^0V \right) \hat{\mathbf{u}}$ ${}_0^t \mathbf{K}_{NL} \hat{\mathbf{u}} = \left( \int_{0V} {}_0^t \mathbf{B}_{NL}^T {}_0^t \mathbf{S} {}_0^t \mathbf{B}_{NL} d^0V \right) \hat{\mathbf{u}}$ ${}_0^t \mathbf{F} = \int_{0V} {}_0^t \mathbf{B}_L^T {}_0^t \hat{\mathbf{S}} d^0V$
Formulação Lagrangeana Atualizada	$\int_{tV} {}_t C_{ijrs} {}_t e_{rs} \delta_t e_{ij} d^tV$ $\int_{tV} {}_t \tau_{ij} \delta_t \eta_{ij} d^tV$ $\int_{tV} {}_t \tau_{ij} \delta_t e_{ij} d^tV$	${}_t \mathbf{K}_L \hat{\mathbf{u}} = \left( \int_{tV} {}_t \mathbf{B}_L^T {}_t \mathbf{C} {}_t \mathbf{B}_L d^tV \right) \hat{\mathbf{u}}$ ${}_t \mathbf{K}_{NL} \hat{\mathbf{u}} = \left( \int_{tV} {}_t \mathbf{B}_{NL}^T {}_t \boldsymbol{\tau} {}_t \mathbf{B}_{NL} d^tV \right) \hat{\mathbf{u}}$ ${}_t \mathbf{F} = \int_{tV} {}_t \mathbf{B}_L^T {}_t \hat{\boldsymbol{\tau}} d^tV$

## Trajetória de Equilíbrio

Em uma formulação geral do processo incremental-iterativo, para a obtenção da trajetória de equilíbrio de uma análise não linear, as equações de equilíbrio (15) e (16) podem ser transformadas em uma equação que corresponde à iteração  $j$  do passo  $i$ , e pode ser escrita da seguinte maneira:

$$[K]_{j-1}^i \cdot \{\delta U\}_j^i = \delta \lambda_j^i \cdot \{P\} + \{Q\}_{j-1}^i \quad (17)$$

onde,

$[K]_{j-1}^i$  é a matriz de rigidez tangente na iteração  $j - 1$  do passo  $i$ , função do campo de deslocamentos  $\{U\}_{j-1}^i$  e obtida, neste trabalho, a partir da soma da parte linear e não linear das matrizes de rigidez indicadas nas equações (15) e (16);

$\{\delta U\}_j^i$  é o vetor de deslocamentos incrementais da iteração  $j$  do passo  $i$ ;

$\delta \lambda_j^i$  é o incremento do fator de cargas na iteração  $j$  do passo  $i$ ;

$\{P\}$  é o vetor de cargas de referência, representado pela soma dos vetores  $\mathbf{R}$  e  $\mathbf{F}$  das equações (15) e (16);

$\{Q\}_{j-1}^i$  é o vetor de forças residuais da iteração  $j - 1$  do passo  $i$ .

Para a resolução da Eq. (17), a primeira providência a ser realizada, é estipular o incremento de fator de carga,  $\delta \lambda_j^i$ , e o seu cálculo varia em função do método de controle escolhido. Com o valor estabelecido, pode-se obter o incremento do vetor de deslocamentos,  $\{\delta U\}_j$ , que pode

ser decomposto nas parcelas associadas à carga de referência  $\{\delta U\}_j^P$  e à carga residual  $\{\delta U\}_j^Q$ , da seguinte forma:

$$\{\delta U\}_j = \delta \lambda_j \cdot \{\delta U\}_j^P + \{\delta U\}_j^Q \quad (18)$$

com

$$[K]_{j-1} \cdot \{\delta U\}_j^P = \{P\} \quad (19)$$

e

$$[K]_{j-1} \cdot \{\delta U\}_j^Q = \{Q\}_{j-1} \quad (20)$$

Com os valores encontrados para  $\delta \lambda_j^i$  e  $\{\delta U\}_j$ , as variáveis do problema podem ser atualizadas, através das seguintes equações:

$$\lambda_j = \lambda_{j-1} + \delta \lambda_j \quad (21)$$

$$\{U\}_j = \{U\}_{j-1} + \{\delta U\}_j \quad (22)$$

Com a atualização dos deslocamentos, as tensões internas são, também, atualizadas. Ao final de cada iteração, o equilíbrio do sistema é verificado pelo critério das forças, a partir do cálculo do vetor de forças residuais,  $\{Q\}_j$ , ou então pelo critério dos deslocamentos, mediante a magnitude do vetor de deslocamentos incrementais  $\{\delta U\}_j$ . O processo iterativo só termina quando se alcança determinado critério de convergência. Se a convergência não for atingida, será necessário o cálculo de uma nova iteração, a partir do cálculo dos valores de  $\{\delta U\}_j^P$  e  $\{\delta U\}_j^Q$  pelas equações 19 e 20, podendo-se assim obter o valor de  $\delta \lambda_j$  com uma equação de restrição que envolve combinações das grandezas do problema.

O vetor de cargas residuais da iteração  $j$  é dado por:

$$\{Q\}_j = \lambda_j \cdot \{P\} - \{F\}_j, \quad (23)$$

onde  $\{F\}_j$  é o vetor de forças equivalentes às tensões internas ao final da iteração  $j$ .

Destaca-se ainda, que na primeira iteração de cada passo, o vetor de cargas residuais  $\{Q\}_{j-1}$  é nulo.

Por ser uma formulação aplicável a vários tipos de métodos de controle, Yang e Shieh (1990) apresenta um diagrama com os principais passos para um algoritmo genérico de resolução do processo incremental-iterativo, como mostra a Fig. 1. Este algoritmo foi implementado no INSANE em Fuina (2004).

## 2.3 Sistema INSANE

O sistema INSANE (*INterative Structural ANalysis Environment*) é um ambiente computacional implementado em linguagem de programação Java, utilizando a Programação Orientada a Objetos (POO) e desenvolvido no Departamento de Engenharia de Estruturas (DEES) da Universidade Federal de Minas Gerais (UFMG).

A POO é baseada em classes e objetos e isto permite, segundo Fonseca (2008), que programas desenvolvidos desta maneira sejam divididos em módulos independentes, proporcionando o seu desenvolvimento por várias pessoas, concomitantemente. Já a linguagem Java, de acordo com Fonseca (2008), além de ser uma linguagem orientada a objetos, tem a característica de

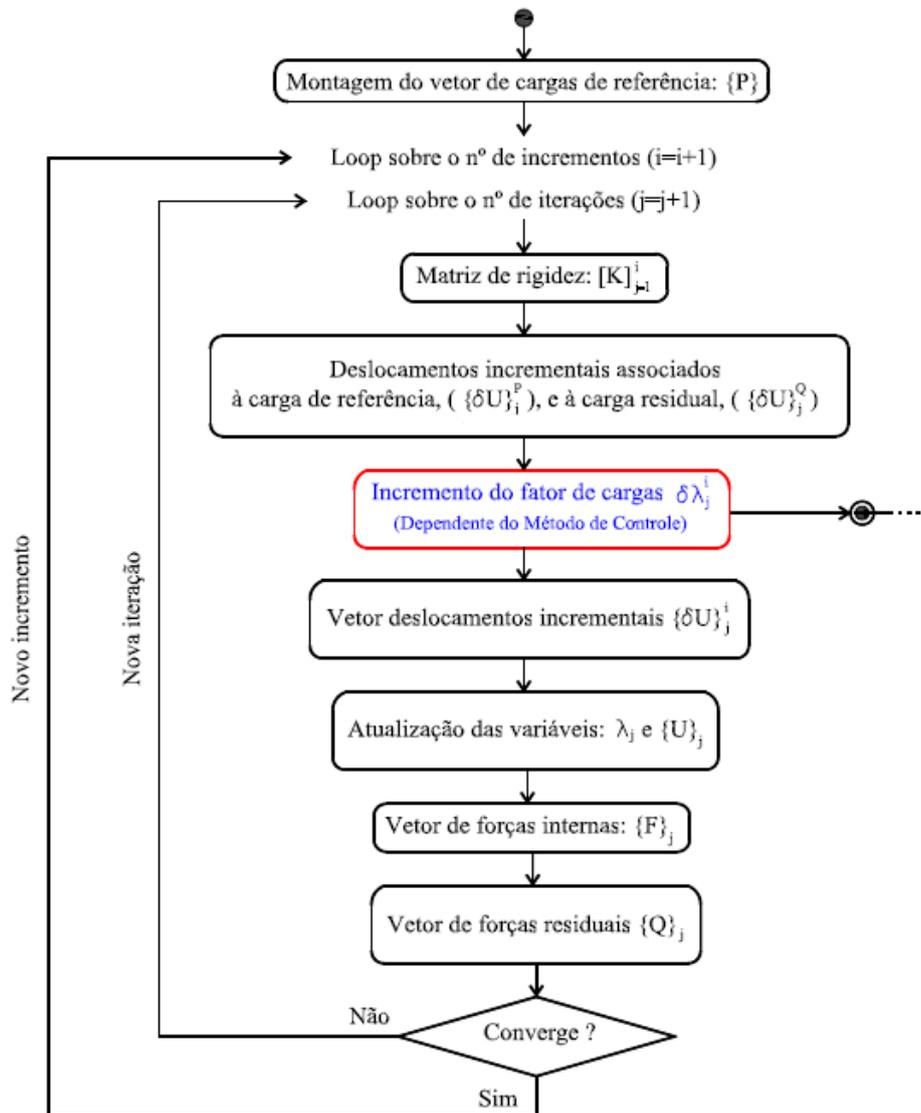


Figura 1: Algoritmo para métodos de controle (Fuina, 2004).

ter portabilidade, por ser desenvolvida de forma a ser independente de plataforma, fazendo com que o código possa ser compilado em um sistema operacional e executado em outro. Com isso, a plataforma INSANE tem como princípio ser um ambiente segmentado, amigável a mudanças e, assim, ser melhorado e ampliado de forma progressiva, sem necessidade de muitas modificações.

O INSANE pode ser dividido em três grandes segmentos: o pré-processador, que é a aplicação gráfica interativa que oferece ferramentas para inserir dados do modelo de diferentes formas; o processador, como sendo o núcleo numérico do sistema, responsável pela leitura dos dados e obtenção dos resultados das análises dos modelos discretos; e por fim o pós-processador, que permite ao usuário a visualização dos resultados ao fim da análise.

Para a solução dos problemas na plataforma, a estrutura do núcleo numérico é concentrada nas relações entre duas interfaces, *Assembler* e *Persistence*, e duas classes abstratas, as classes *Solution* e *Model*, como pode-se observar na Fig. 2.

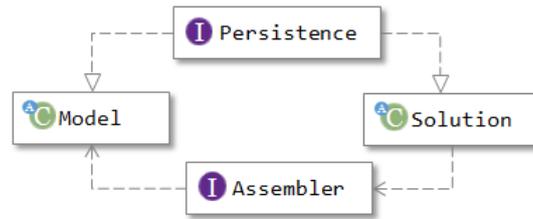


Figura 2: Organização do núcleo numérico do INSANE

A interface *Persistence* é a responsável por receber os arquivos de entrada gerados no pré-processamento do programa. Assim, os dados necessários para descrição do modelo podem ser organizados, para, então, preencherem as informações do modelo. Após a classe *Solution* obter os resultados das equações, *Persistence* também é responsável por gerar os arquivos com os dados de saída para uma posterior utilização dos resultados.

Após a leitura de dados por *Persistence*, são as classes de *Model* que recebem os dados armazenados, podendo, assim, representar o modelo discreto a ser analisado. A partir da classe abstrata *Model* que se derivam algumas classes, necessárias para poder se distinguir atributos específicos para determinados tipos de modelos. Dentre estas classe tem-se a classe *FemModel*, que é a responsável pelos modelos discretos de MEF. Ela possui listas de nós, elementos, modelos de análise, funções de forma, degenerações, materiais e carregamentos, necessários para a descrição da estrutura.

Derivando de *FemModel*, tem-se a classe *GFemModel*, que sendo uma classe filha de *FemModel*, herda todas as características presentes nesta classe e, além disso, possui comportamento e atributos específicos do Método dos Elementos Finitos Generalizados (MEFG), como, por exemplo, uma lista de objetos das classes da herança de *EnrichmentType*. A classe *EnrichmentType* é bem específica para o MEFG, pois é ela que armazena e manipula as informações relacionadas aos tipos de enriquecimento nodal. *EnrichmentType* é uma classe abstrata, pois dela derivam classes que implementam tipos especiais de enriquecimento.

Além disso, para a construção da classe *Model*, tem-se implementado as classes *Node*, *Element*, *Shape* e *ProblemDriver*. A classe *Node* é a encarregada de armazenar todas as informações discretas associadas a cada nó do problema. Especificamente para a análise via MEFG, existem as variáveis estáticas CLOUD-SIZE, que representa o tamanho da nuvem deste nó, e ENRICHMENT-TYPE, que referencia um dos tipos de enriquecimento que foram armazenados na lista de *EnrichmentType* em *GFemModel*, pois, desta forma, cada nó tem a ele associado uma nuvem,  $\omega_j$ , e uma ou mais funções de enriquecimento  $L_{ji}(x)$ .

*Shape* é uma classe abstrata para representar as funções de forma, e derivadas de diversos tipos de aproximações diferentes. A hierarquia desta classe é dividida, primeiramente, no tipo de coordenada utilizada (real ou natural), e logo depois, as coordenadas naturais são divididas segundo a geometria dos elementos finitos (unidimensional, triangular, quadrilateral tetraédrica ou hexagonal).

A classe *Element* é a responsável em representar o elemento do modelo discreto, possuindo todos os atributos necessários para sua discretização, como por exemplo, a função de forma, a lista de pontos de integração e lista de nós para formar o elemento. É *Element* que informa à interface *Assembler* todas as grandezas fundamentais para montagem das equações finais

do modelo. Dentre os atributos, existe também um objeto do tipo *ProblemDriver*, que faz referência a um tipo de problema a ser modelado.

Derivando de *ProblemDriver*, existem diferentes tipos de problemas e formulações de modelos discretos, e dentre eles, a classe *SolidMech*, que representa as formulações para um modelo da mecânica dos sólidos. As diferentes formulações de MEF estão implementadas no programa como classes filhas de *SolidMech*, e dentre elas, a classe *Parametric* e a classe *GFemParametric*, que são responsáveis por fornecer informações relevantes aos elementos paramétricos. A diferença entre essas duas classes é que *GFemParametric* usa as coordenadas físicas e naturais do elemento para o cálculo das funções de forma e derivadas, enquanto a classe *Parametric* utiliza apenas as coordenadas naturais nos pontos de integração, fazendo com que as aproximações pelo MEF dependam menos da malha, e assim sejam menos penalizadas com distorções no elemento, por exemplo.

Dependendo do tipo do modelo de análise, é a classe abstrata *AnalysisModel* que fornece as informações dos modelos matemáticos necessárias aos elementos finitos, como, por exemplo, o cálculo da matriz de deformação. Sua implementação é realizada pelas diversas classes que representam diferentes modelos de análise, como os modelos em estado plano de tensão (*PlaneStress*), estado plano de deformação (*PlaneStrain*), modelos lineares (*Line*) e sólidos (*Solid*). Neste mesmo contexto, existe a interface *GFemAnalysisModel*, que possui métodos com características específicas para os modelos de elementos finitos generalizados. Assim, uma classe para um modelo de análise de um problema via MEF, como por exemplo as classes *GFemPlaneStress* e *GFemSolid*, implementam a interface *GFemAnalysisModel* e também são derivadas da classe abstrata *AnalysisModel*. Com isto, elas possuem métodos gerais dos modelos matemáticos e métodos mais específicos para o MEF.

Após armazenadas as informações do modelo nas classes relativas a *Model*, elas são recuperadas pela interface *Assembler*, que possui os métodos necessários para montar as matrizes e vetores do modelo, que, no caso deste trabalho, é representado pela Eq. (17) para os processos incrementais-iterativos.

*Assembler* é implementada pela classe *FemAssembler*, por ser conveniente a diferentes tipos de problemas que podem ser modelados pelo Método dos Elementos Finitos. Derivando de *FemAssembler* tem-se a classe *GFemAssembler*, responsável por montar a equação para modelos que serão resolvidos via MEF.

Após a montagem do sistema de equações a ser resolvido, é a classe *Solution* que soluciona o sistema de equação. Por ser uma classe abstrata, possui classes derivadas que dispõem de métodos para a solução de diferentes tipos de problemas. Dentre essas subclasses tem-se a *SteadyState*, que representa a solução de problemas lineares estáticos e, por isso, é a mais simples delas. *SteadyState* tem como um de seus atributos, um objeto da classe *LinearEquationSystem*, que é a responsável por resolver o sistema de equações lineares.

No enriquecimento da partição da unidade para a construção das funções de forma do MEF, a matriz de rigidez do sistema pode se tornar semidefinida positiva, quando a PU e as funções de enriquecimento são polinomiais. Para se resolver este sistema, na classe *LinearEquationSystem* implementou-se o método de processo iterativo, proposto por Strouboulis, Babuška e Copps (2000).

Para a solução de problemas não lineares, a classe *EquilibriumPath* é a classe herdeira de

*Solution* que generaliza a solução de problemas que necessitam determinar as trajetórias de equilíbrio. De *EquilibriumPath* surgem mais duas subclasses, a *DynamicEquilibriumPath* para o cálculo de problemas dinâmicos, e a *StaticEquilibriumPath*, que representa a solução para uma análise não linear estática.

A classe *StaticEquilibriumPath* possui como atributos objetos do tipo *Step* e *IterativeStrategy*. A interface *IterativeStrategy* define o método de controle a ser utilizado para obtenção das trajetórias de equilíbrio, e é algo informado pelo usuário. Em sua hierarquia estão representados os diferentes métodos de controle: o controle de carga, o controle de deslocamento, o controle por trabalho, o controle de comprimento de arco, e controle de arco por variação de energia. Todas as classes possuem os métodos *getPredictor()* e *getCorrector()*, que calculam o fator de carga de cada iteração, e também possuem um objeto do tipo *Step* que terá todas as informações sobre a iteração anterior para o cálculo do fator de carga.

A interface *Step* é a que representa o tipo de processo incremental iterativo utilizado na análise. São implementadas a partir dela as classes *StandardNewtonRaphson* e *ModifiedNewtonRaphson*, que possuem, respectivamente, métodos incrementais iterativos da formulação Newton Raphson padrão e modificado. Estas classes possuem como atributos um objeto do tipo *Assembler*, que informa as matrizes e vetores da equação a ser resolvida em cada iteração do processo, um objeto da classe *LinearEquationSystem*, para resolver o sistema de equações algébricas lineares na iteração, e um objeto *IterativeStrategy*, como forma de representar qual estratégia de iteração está sendo utilizada. Dentre os métodos presentes nas classes, existe o método *setConvergence()*, que cria um objeto do tipo *Convergence*, podendo, desta maneira, calcular a convergência baseada em força, deslocamento ou ambos, e também o método *getConvergence()*, responsável por verificar a convergência da iteração corrente.

### Expansão da plataforma para a análise com não linearidade geométrica

Para este trabalho, o objetivo foi que a plataforma solucionasse problemas com não linearidade geométrica via MEFG através da formulação Lagrangiana Total.

A formulação paramétrica é tratada no MEFG na classe *GFemParametric*, a exemplo da *Parametric* usada no MEF, e ambas derivam de *SolidMech* e de *ProblemDriver* (Fig. 3), que possui apenas métodos gerais para a montagem da equação final do modelo. Com isto, criou-se a classe *GFemGeometricallyNonLinearTl*, que é responsável por fornecer informações pertinentes aos elementos paramétricos de MEFG específicas à formulação Lagrangiana Total. No momento que o elemento é acionado por *Assembler* para a montagem da matriz de rigidez, ele consulta o seu *ProblemDriver*, que neste caso será a nova classe *GFemGeometricallyNonLinearTl*, e esta classe fornecerá as matrizes de rigidez linear e não linear e o vetor de forças da Eq. (15), como indicado na Fig. 3.

O método *getLinearIncrementalC* é o responsável pelo cálculo da matriz de rigidez linear ( $\mathbf{K}_L$ ), o método *getNonLinearIncrementalC* pelo cálculo da matriz não linear ( $\mathbf{K}_{NL}$ ), e o método *getIncrementalC* pela soma das duas matrizes, como determinado pela Eq. (15). Já o método *getF* é o responsável pelo cálculo do vetor de forças nodais.

Para o cálculo correto dessas matrizes de rigidez e do vetor de forças, adicionando, para isto, o enriquecimento oriundo do modelo de elementos finitos generalizados, foram necessárias modificações em um grupo de classes que implementam a Interface *GFemAnalysisModel*. Métodos existentes nas classes derivadas de *AnalysisModel* para a análise via Elementos Finitos

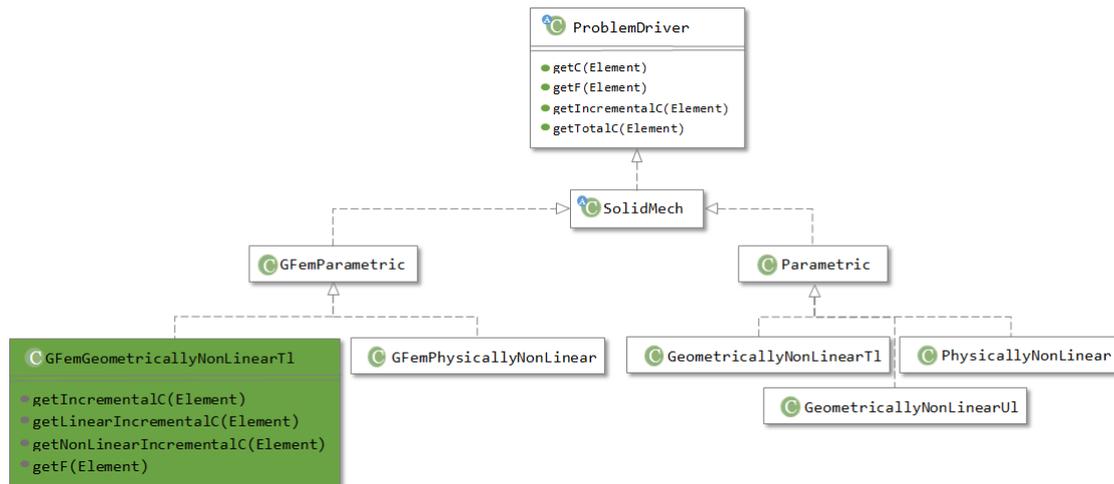


Figura 3: Diagrama UML parcial da classe ProblemDriver

tos, foram adicionados em *GFemAnalysisModel* e modificados de maneira que contemplassem esses enriquecimentos nodais dos modelos do MEFMG. Os métodos criados foram *getInternalVariablesOperator()* para a montagem da matriz  $B_L$  e *getNonLinearInternalVariablesOperator()* para a montagem da matriz  $B_{NL}$ , ambas indicadas nas equações da Tabela 1. Os métodos *getNonLinearDualInternalVariablesMatrix()* e *getGreenInternalVariables* relativos à construção do segundo tensor de Piola-Kirchoff (matriz  $S$  nas equações da Tabela 1), também foram introduzidos na classe, para que o enriquecimento contido nas matrizes  $B$  fossem considerados.

Além dessas modificações nos modelos de análise, introduziu-se dois métodos na classe *GFemElement*, para poder recuperar características específicas dos modelos de elementos finitos generalizados. Dentre estes métodos tem-se o *getPointGreenInternalVariables()*, que retorna o vetor das variáveis internas de Green em um ponto do elemento, e foi criado de maneira que levasse em consideração os enriquecimentos nas funções de forma, e o *getCartesianNodalDisplacementsMatrix*, responsável pela montagem de uma matriz de deslocamentos nodais do elemento, e sua criação foi necessária para que os deslocamentos dos graus de liberdade adicionais, devido ao enriquecimento, fossem contabilizados na matriz final.

### 3 RESULTADOS E DISCUSSÃO

Após a criação da classe *GFemGeometricallyNonLinearTl* e da introdução de métodos em algumas classes na plataforma INSANE, foram feitas análises e comparações de resultados de exemplos propostos por Fonseca (2008), que foram obtidos via Método dos Elementos Finitos, e que estão indicados nos gráficos que se seguem pelas siglas *LT-NRP*, para o problema analisado via método de Newton-Raphson padrão, e *LT-NRM*, para o método modificado. Nas análises via Método dos Elementos Finitos Generalizados as siglas utilizadas nos gráficos foram *LT-NRP-MEFG* e *LT-NRM-MEFG* para os métodos de Newton-Raphson padrão e modificado, respectivamente.

#### 3.1 Coluna Engastada e Livre

Este exemplo consiste em uma coluna engastada em sua extremidade inferior e livre na outra extremidade, onde é aplicado o carregamento pontual vertical e excêntrico, como apresentada na Fig. 4.

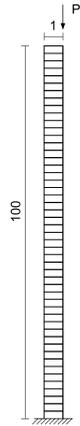


Figura 4: Coluna engastada e livre

O material na análise desta coluna foi considerado linear-elástico isotrópico, cujo módulo de elasticidade é  $E = 12$  e o coeficiente de Poisson  $\nu = 0,3$ . As dimensões usadas para a coluna foram  $1 \times 100 \times 1$ , todas em unidade consistente.

Para as análises via MEF e MEFG, foi utilizado um modelo de análise em estado plano de tensões (*PlaneStress*), e a coluna foi discretizada em 50 elementos finitos quadriláteros de 8 nós para o modelo via Método dos Elementos Finitos e em elementos quadriláteros de 4 nós enriquecidos polinomialmente em  $x$  e  $y$  para se ter uma aproximação quadrática pelo Método dos Elementos Finitos Generalizados. A carga de referência foi  $P_0 = -2,0 \times 10^{-4}uf$ .

No método de Newton-Raphson padrão foi utilizado o método de controle de carga, em que se incrementou  $0,1uc$  a carga aplicada em cada passo, a tolerância à convergência foi de  $0,01$  para as forças e foram necessários 90 passos para a convergência. Já no método modificado, foi necessário utilizar o método de controle de deslocamentos generalizados com um incremento no fator de carga de  $0,5$ , a tolerância foi aumentada para  $0,1$  e foram necessários 1000 passos. A diferença existente entre os dois métodos é devido à natureza da trajetória de cada análise.

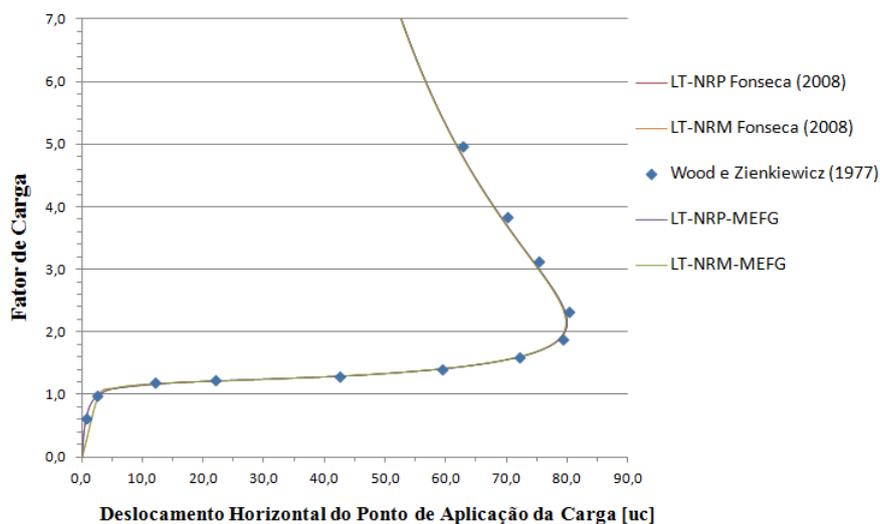


Figura 5: Trajetória de equilíbrio para o deslocamento horizontal do ponto de aplicação de carga.

Os resultados das trajetórias de equilíbrio para o deslocamento horizontal do ponto de aplicação da carga é apresentado na Fig. 5.

Além do trabalho de Fonseca (2008), foi utilizado como referência o resultado analítico apresentado por Wood e Zienkiewicz (1977). Pode-se observar a concordância dos resultados calculados no INSANE para o modelo de Elementos Finitos Generalizados com os valores analíticos e os fornecidos por Fonseca (2008), que também foram obtidos na plataforma INSANE.

### 3.2 Pórtico de Williams

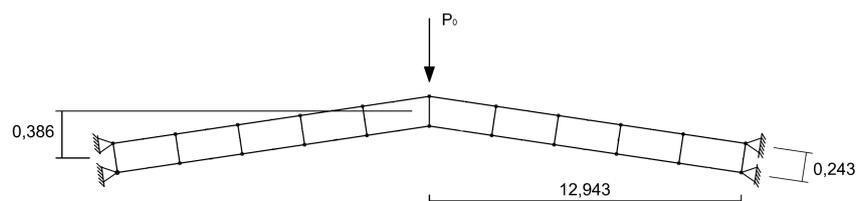


Figura 6: Pórtico de Williams

O Pórtico de Williams, apresentado na Fig. 6, é um pórtico abatido biengastado composto por um material linear-elástico isotrópico com módulo de elasticidade  $E = 10,3 \times 10^6 uf/uc^2$  e coeficiente de Poisson nulo. A espessura é  $0,753uc$ .

O pórtico foi discretizado com 10 elementos quadrilaterais de 8 nós para a análise via MEF no INSANE e com elementos quadrilaterais de 4 nós enriquecidos em x e y para o MEFG. O modelo de análise foi o estado plano de tensões (*PlaneStress*) e a carga de referência considerada foi  $P_0 = -20,0uf$ .

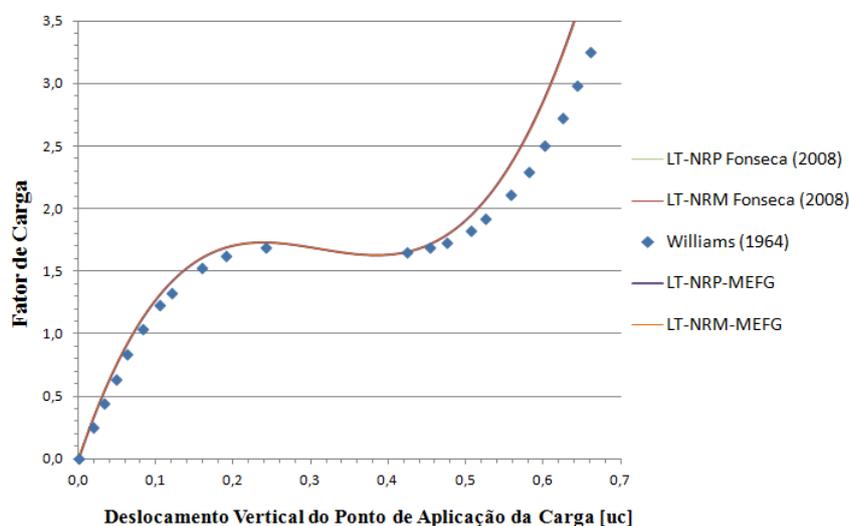
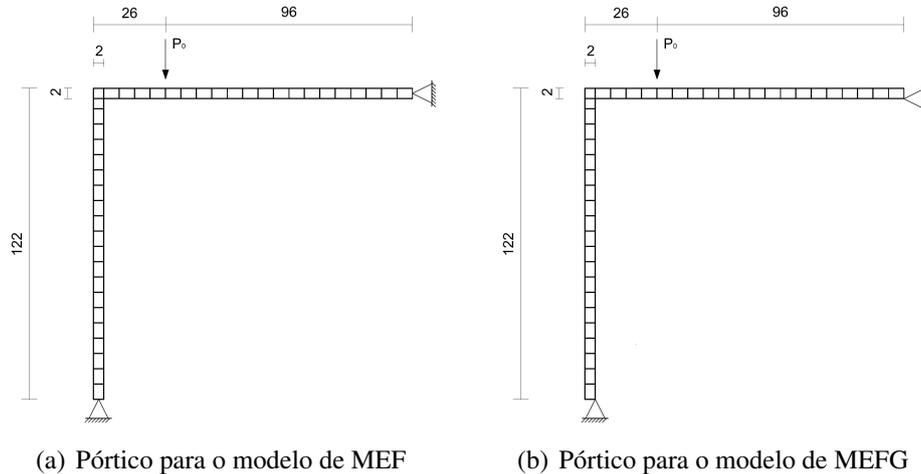


Figura 7: Trajetória de equilíbrio para o deslocamento vertical do ponto de aplicação de carga.

A análise foi feita, tanto para o MEF quanto para o MEFG, utilizando o método de controle de deslocamentos com um fator de carga de  $-0,00325uc$  incrementado no deslocamento vertical do nó de aplicação da carga. Foram realizados 200 passos em cada análise e a tolerância para a convergência foi de  $0,0001$  para as forças.

Observa-se no gráfico da Fig. 7 uma compatibilidade de resultados das análises via MEFG com as análises via MEF feitas por Fonseca (2008), ambas realizadas no INSANE, e a concordância com os valores fornecidos por Williams (1964), que apresentou os resultados analítico e experimental para o pórtico.

### 3.3 Pórtico de Lee



**Figura 8: Pórtico de Lee**

O Pórtico de Lee, apresentado na Fig. 8, é um pórtico em forma de L birrotulado que é amplamente utilizado na validação de formulações não lineares.

Em todas as análises feitas para este pórtico foi utilizado um material linear-elástico isotrópico, com módulo de elasticidade de  $E = 720uf/uc^2$  e coeficiente de Poisson de  $\nu = 0,3$ . A sua seção transversal tem dimensões de  $2uc \times 3uc$ .

A discretização usada foi de 41 elementos finitos quadrilaterais, sendo que para o modelo em Elementos Finitos, esses elementos foram de 8 nós e no modelo de Elementos Finitos Generalizados os elementos de quatro nós são enriquecidos pelos monômios  $x$  e  $y$  para se ter uma aproximação quadrática. As restrições do modelo de MEF foram adicionadas nos nós existentes no meio dos vértices de extremidade dos elementos de 8 nós (Fig. 8(a)). Como os elementos Q4, utilizados no modelo de elementos finitos generalizados, não possuem esses mesmos nós, optou-se por colocar as restrições nos nós de extremidade dos elementos, nos cantos internos do pórtico, como indicado pela Fig. 8(b).

O modelo de análise foi o estado plano de tensões (*PlaneStress*), e a carga de referência considerada foi de  $P_0 = -2,0uf$ . Utilizou-se o método de controle de deslocamento generalizado, com o incremento de fator de carga de 0,01 e foram necessários 1300 passos, com uma tolerância de 0,001 para as forças.

O resultado das trajetórias de equilíbrio para o deslocamento horizontal está indicado na Fig. 9. Pode-se observar que os valores calculados pelo INSANE para a análise via MEFG estão em excelente concordância com os valores fornecidos em Fonseca (2008) e os também apresentados por Galvao (2000), e a pequena diferença existente pode ser devido à mudança na posição das restrições do modelo.

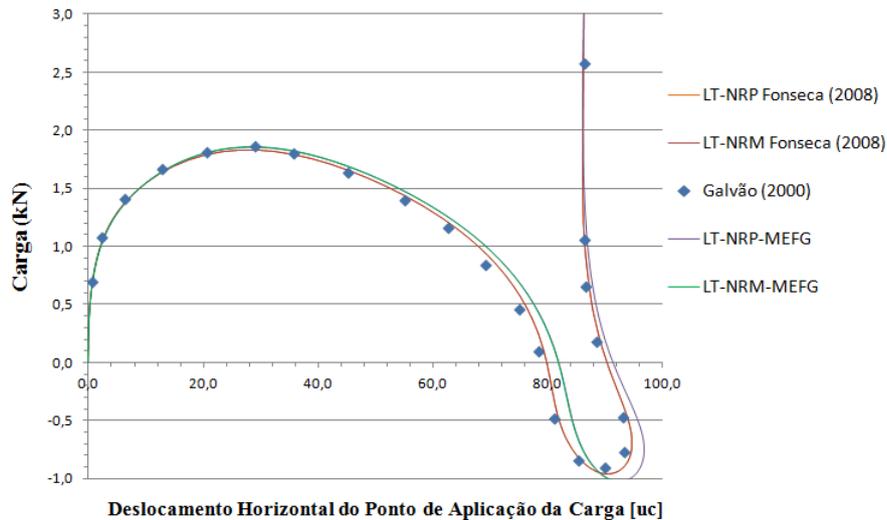


Figura 9: Trajetória de equilíbrio para o deslocamento horizontal do ponto de aplicação da carga

## 4 CONCLUSÃO

Em uma análise com não linearidade geométrica, os grandes deslocamentos e deformações envolvidos podem fazer com que a malha sofra consideráveis distorções, e esta distorção faz com que se percam alguns termos da equação de aproximação do elemento, penalizando a qualidade da solução pelo MEF. Sabe-se que no MEFG, a aproximação é menos prejudicada, se comparada com a do MEF, pois a estratégia de enriquecimento é construída nas coordenadas reais do problema. Tal característica, torna o MEFG bastante adequado para ser utilizado em análises com não linearidade geométrica.

A plataforma INSANE já possui um arcabouço de recursos implementados e validados para a solução de problemas com não linearidade geométrica via Método dos Elementos Finitos. Uma vez já implementado na plataforma o MEFG, e tendo em vista sua característica adequada à solução deste tipo de problema, neste trabalho se propôs a implementação da análise geometricamente não linear pelo MEFG para a formulação Lagrangiana Total. Dado os resultados dos exemplos apresentados no item 3, conseguiu-se validar o sistema de solução proposto e implementado na plataforma INSANE. Esta comparação de resultados foi possível, uma vez que a discretização utilizada nos modelos de MEF (elementos Q8) é equivalente a se utilizar elementos quadriláteros de 4 nós enriquecidos por  $x$  e  $y$  no MEFG. A partir desta primeira etapa de validação da implementação da formulação Lagrangiana Total, os próximos passos do trabalho serão, a implementação da formulação Lagrangiana Atualizada, bem como sua validação, e ao final, a comparação entre o MEF e o MEFG com relação à distorção da malha na descrição do comportamento não linear geométrico.

## AGRADECIMENTOS

Os autores agradecem o apoio financeiro em forma de fomento à pesquisa concedido pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - grants 308932/2016-1 e 308785/2014-2) e pela Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

## REFERÊNCIAS

- Alves, P. D. *Estratégia Global-Local Aplicada ao Método dos Elementos Finitos Generalizados*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, Maio 2012.
- Alves, P. D.; Barros, F. B.; Pitangueira, R. L. S. An object-oriented approach to the generalized finite element method. *Advances in Engineering Software*, v. 59, p. 1–18, 2013.
- Barros, F. B. *Métodos Sem Malha e Métodos dos Elementos Finitos Generalizados em Análise Não-Linear de Estruturas*. Tese (Doutorado) — EESC - USP, São Carlos, SP, Brasil, 2002.
- BATHE, K.-J. *Finite Element Procedures*. [S.l.]: Upper Saddle River, N. J., 1996. (Prentice-Hall civil engineering and engineering mechanics series). ISBN 0133014584.
- Belytschko, T.; Black, T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, v. 45, p. 601–620, 1999.
- Duarte, C. A.; Babuška, I.; Oden, J. T. Generalized finite element methods for three-dimensional structural mechanics problems. *Computers and Structures*, v. 77, p. 215–232, 2000.
- Duarte, C. A.; Oden, J. T. An hp adaptive method using clouds. *Computer methods in applied mechanics and engineering*, v. 139, p. 237–262, 1996.
- Fonseca, F. T. da. *Sistema Computacional para Análise Dinâmica Geometricamente Não-linear através do Método dos Elementos Finitos*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, Agosto 2008.
- Fuina, J. S. *Métodos de controle de deformações para análise não-linear de Estruturas*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, Fevereiro 2004.
- Galvao, A. da S. *Formulações Não-Lineares de Elementos Finitos para Análise de Sistemas Estruturais Metálicos Reticulados Planos*. Dissertação (Mestrado) — Universidade Federal de Ouro Preto, Ouro Preto, MG, Brasil, Março 2000.
- Melenk, J. M.; Babuška, I. The partition of unity finite element method: Basic theory and applications. *Computer methods in applied mechanics and engineering*, v. 39, p. 289–314, 1996.
- Ribeiro, J. C. L. *Desenvolvimento e Aplicação de um Sistema Computacional para Simulação via Método dos Elementos Finitos do Comportamento de Estruturas de Aço e Mistas em Situação de Incêndio*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil, Dezembro 2009.
- Strouboulis, T.; Babuška, I.; Copps, K. The design and analysis of the generalized finite element method. *Computer methods in applied mechanics and engineering*, v. 181, p. 43–69, 2000.
- Williams, F. W. An approach to the non-linear behavior of the members of a rigid jointed plane framework with finite deflections. *The Quarterly Journal of Mechanics and Applied Mathematics*, v. 17, p. 451–469, 1964.
- Wolenski, A. R. V.; Monteiro, A. B.; Barros, F. B.; Penna, S. S.; Pitangueira, R. L. S. Application of the generalized finite element method on the physically nonlinear analysis of structures. In: *XXXV CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering*. [S.l.: s.n.], 2014.

Wood, R. D.; Zienkiewicz, O. C. Geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells. *Computers and Structures*, v. 7, p. 725–735, 1977.

Yang, Y. B.; Shieh, M. S. Solution method for nonlinear problems with multiple critical points. *AIAA Journal*, v. 28, n. 12, p. 2110–2116, 1990.