



FRAMEWORK COMPUTACIONAL PARA MODELOS DE PLASTICIDADE

Danilo Bento Oliveira

Samuel Silva Penna

Roque Luiz da Silva Pitangueira

dboliveira00@ufmg.br

spenna@dees.ufmg.br

roque@dees.ufmg.br

Universidade Federal de Minas Gerais

Avenida Antônio Carlos, 6627, Pampulha 31270-901, Belo Horizonte, Minas Gerais, Brasil

Resumo. O constante crescimento de pesquisas na área da mecânica computacional contribui para o surgimento de diversos softwares utilizados por engenheiros e pesquisadores, capazes de realizar análises sofisticadas, como os que utilizam a teoria da plasticidade. Neste cenário, o presente trabalho destaca o ambiente computacional INSANE (INteractive Structural ANalysis Environment). Atualmente, o núcleo numérico do sistema possui um framework computacional para modelagem constitutiva, em que dentre as suas diversas características ressalta-se a independência do método numérico utilizado e os diferentes tipos de modelos de análise. Tal framework é propício ao desenvolvimento colaborativo de modelos para a análise fisicamente não linear. Este trabalho abordada o projeto orientado a objetos da implementação dos modelos constitutivos para plasticidade no sistema INSANE, posicionando-o no núcleo numérico e detalhando as classes que o compõe, e, apresenta uma hierarquia de classes para os algoritmos de retorno, necessários para a integração das relações constitutivas que regem o comportamento do material, no contexto de plasticidade computacional. Finalmente, serão apresentadas simulações numéricas do comportamento fisicamente não linear de modelos estruturais via Método dos Elementos Finitos, para exemplificar os modelos constitutivos e os algoritmos de retorno implementados.

Palavras Chave: POO, JAVA, Plasticidade Computacional, MEF, Modelos Constitutivos

1 INTRODUÇÃO

A teoria da plasticidade surgiu da necessidade de estudar equações constitutivas que permitissem descrever matematicamente deformações imediatas e não reversíveis que ocorrem em um corpo sólido, ou seja, as deformações que não desaparecem totalmente ao serem removidas as forças que lhe deram origem (Chen e Han, 1988; Souza Neto et al., 2008). O desenvolvimento da teoria matemática da plasticidade está intimamente ligado ao avanço das técnicas computacionais baseadas em modelos numéricos, como o Método dos Elementos Finitos, e alimentado pelo aumento constante da capacidade de processamento dos computadores.

O constante crescimento de pesquisas na área da mecânica computacional contribui para o desenvolvimento de diversos softwares, utilizados por engenheiros e pesquisadores, capazes de realizar análises sofisticadas, tendo como pano de fundo modelos constitutivos bastante complexos, como os que utilizam a teoria da plasticidade. Uma enorme gama de problemas práticos de engenharia, como a análise de tensões em estruturas metálicas e de concreto, em solos e rochas, pode ser resolvida a partir da aplicação destas técnicas, obtendo um nível de precisão aceitável. É neste contexto que o ambiente computacional INSANE (INteractive Structural ANalysis Environment) vem sendo desenvolvido no Departamento de Estruturas da Universidade Federal de Minas Gerais.

O INSANE é um sistema computacional que segue o paradigma de programação orientado a objetos (POO), sendo implementado em JAVA, de código aberto e de fácil expansão. O sistema tem sua estrutura dividida em três grandes aplicações: Os pré e pós processadores, que são aplicações gráficas interativas e um núcleo numérico, responsável pelo processamento, que contém as implementações dos modelos numéricos e das técnicas de solução apropriadas.

Em seu trabalho, Penna (2011) propôs a criação de um Ambiente Unificado capaz de contemplar diversos modelos constitutivos, ampliando as possibilidades da análise fisicamente não linear de estruturas. Este artigo trata do projeto orientado a objetos relativo à implementação de modelos constitutivos elastoplásticos segundo aludido ambiente, que foi incorporado ao núcleo numérico do INSANE, mais especificamente na segmentação relativa aos modelos constitutivos.

2 FORMULAÇÃO DOS MODELOS CONSTITUTIVOS

Os modelos constitutivos em geral apresentam uma notação própria e, embora em muitos casos guardem semelhanças, a falta de uma unidade das formulações impede uma implementação computacional genérica e objetiva. O Ambiente Unificado proposto por Penna (2011) apresenta uma expansão da estrutura teórica proposta por Carol et al. (1994), sendo capaz de contemplar vários modelos constitutivos - elastoplásticos ou de degradação elástica; isotrópico, ortotrópico ou anisotrópico - formulados com uma ou várias funções de carregamento.

Na formulação dos modelos elastoplásticos as equações das variações de tensão podem ser escritas como

$$\dot{\sigma}_{ij} = E_{ijkl}^0(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^p) \quad (1)$$

com

$$\dot{\varepsilon}_{kl}^p = \dot{\lambda} m_{kl} \quad (2)$$

onde E_{ijkl}^0 são as componentes do tensor de rigidez elástico, m_{kl} é o gradiente das funções de potenciais plásticos (regra de fluxo) e $\dot{\varepsilon}_{kl}^p$ é a variação das deformações plásticas.

A definição da superfície de plastificação segundo modelos fenomenológicos é realizada através do uso de um critério de escoamento. Ao longo dos anos, muitos critérios foram sendo propostos à medida que se estudavam as diversas classes de materiais. A Equação (3) representa a forma linearizada da condição de consistência a partir da superfície de escoamento $F(\boldsymbol{\sigma}, \mathbf{p})$ em termos do tensor de tensões, $\boldsymbol{\sigma}$, e de um vetor de variáveis internas, $\mathbf{p}$.

$$\dot{F} = \left. \frac{\partial F}{\partial \sigma_{ij}} \right|_p \dot{\sigma}_{ij} + \left. \frac{\partial F}{\partial p_k} \right|_\sigma \dot{p}_k = 0 \quad (3)$$

Assumindo que as variáveis internas, p_i , são funções das deformações plásticas e que $\dot{p}_i = \partial p_i / \partial \varepsilon_{kl}^p$, pode-se reescrever a Eq. (3) da seguinte forma

$$n_{ij} \dot{\sigma}_{ij} - H \dot{\lambda} = 0 \quad (4)$$

com

$$n_{ij} = \left. \frac{\partial F}{\partial \sigma_{ij}} \right|_p \quad (5)$$

Comparando as Eqs. (3) e (4), obtém-se

$$H = - \left. \frac{\partial F}{\partial \lambda} \right|_\sigma = - \left. \frac{\partial F}{\partial p_i} \right|_\sigma \frac{\partial p_i}{\partial \varepsilon_{kl}^p} m_{kl} \quad (6)$$

Nas Equações (4) e (5), n_{ij} representa as derivadas de F para valores constantes do multiplicador plástico λ (ou seja $\dot{\lambda} = \dot{\varepsilon}_{kl}^p = 0$, não ocorrendo nenhuma mudança das deformações plásticas e variáveis internas p_i), e geometricamente, representa um vetor normal à superfície de carregamento atual $F = 0$ no espaço de tensões. O módulo inelástico H é obtido a partir de derivadas de F para valores constantes de σ , sendo positivo para o caso de endurecimento (expansão da superfície de escoamento), zero para o caso de plasticidade perfeita (sem alteração da superfície de escoamento) e negativo para o amolecimento (contração da superfície de escoamento).

Substituindo as Eqs. (1) e (2) na Eq. (4), obtém-se uma expressão para o multiplicador plástico ($\dot{\lambda}$)

$$\dot{\lambda} = \frac{n_{cd} E_{cdkl}^0 \dot{\varepsilon}_{kl}}{H + n_{pq} E_{pqrs}^0 m_{rs}} \quad (7)$$

No numerador da Eq. (7), identifica-se o incremento da tensão e o seu produto escalar com o vetor normal n_{cd} cuja resultante, se positiva, aponta para fora da superfície de carga (carregamento plástico), e, em caso de resultante negativa, aponta para o interior

da superfície (descarregamento elástico). Para a consistência dos sinais, é necessário que o denominador da Eq. (7) seja sempre positivo.

O operador tangente relaciona as variações de deformação com variações de tensão na forma

$$\dot{\sigma}_{ij} = E_{ijkl}^t \dot{\epsilon}_{kl} \quad (8)$$

Substituindo as Eqs. (1) e (7) em (8), obtém-se

$$E_{ijkl}^t = E_{ijkl}^0 - \frac{E_{ijab}^0 m_{ab} n_{cd} E_{cdkl}^0}{H + n_{pq} E_{pqrs}^0 m_{rs}} \quad (9)$$

3 IMPLEMENTAÇÕES COMPUTACIONAIS

3.1 Núcleo Numérico

O núcleo numérico do INSANE é responsável pelo processo de solução e obtenção de resultados da análise estrutural. O atual estágio organizacional do núcleo numérico é resultado de diversos trabalhos dentre os quais destacam-se Almeida (2005), Fonseca (2006), Saliba (2007), Fonseca (2008), Ajeje (2009), Fuina (2009) e Penna (2011). Sua estrutura é composta pela interface **Assembler** e pelas classes abstratas **Model** e **Solution**, conforme ilustrado na Fig. 1.

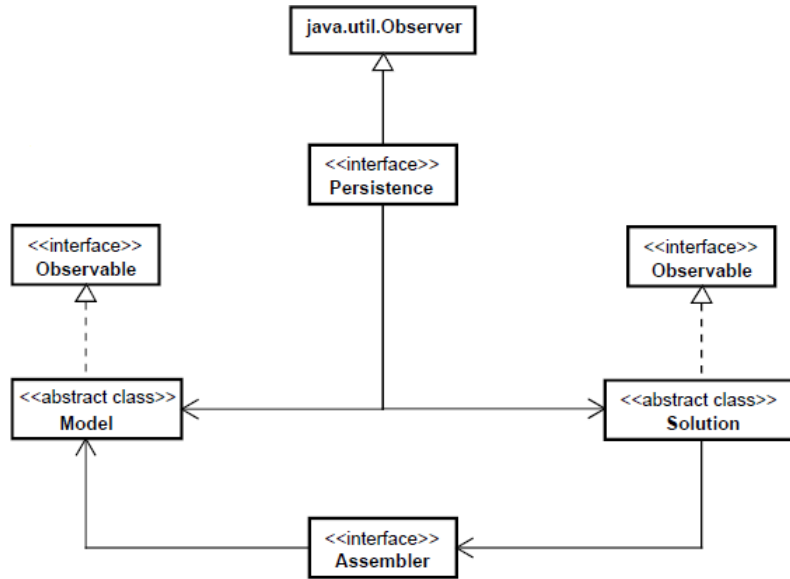


Figura 1: Organização do Núcleo Numérico.

A interface **Assembler** é a responsável pela montagem do sistema matricial de segunda ordem com o qual é possível representar diversos tipos de modelos, a classe abstrata **Solution** desencadeia o processo de solução, possuindo os recursos necessários para resolver esse sistema matricial, seja ele linear ou não linear, e a classe abstrata **Model** possui os dados relativos ao modelo discreto a ser analisado e fornece para **Assembler**

todas as informações necessárias para montar a equação do modelo, que será resolvida por **Solution**.

As classes abstratas **Model** e **Solution** se comunicam com a classe **Persistence**, que é a interface responsável pelo tratamento dos dados de entrada e pela persistência dos dados de saída, sempre que observada alguma alteração no estado dos modelos. Esta persistência é realizada por meio de arquivos XML (eXtensible Markup Language), que possibilitam a criação de dados estruturados com base em arquivo texto.

A comunicação entre as classes **Model** e **Solution** e a interface **Persistence** ocorre através de um mecanismo de propagação de mudanças proporcionado pelo padrão de projeto *Observer*. **Persistence** é um componente observador, enquanto os componentes observados são as classes abstratas **Model** e **Solution**. Quando alguma mudança ocorre no estado de um objeto observado, o mecanismo de propagação de mudanças é disparado e se encarrega de notificar os objetos observadores para se atualizarem, assegurando a consistência e a comunicação entre o componente observador e os componentes observados.

A Figura 2 mostra a hierarquia da interface **Model**. As classes que a implementam são constituídas por listas de objetos inerentes ao modelo discreto a ser analisado, como por exemplo **FemModel**, que representa um modelo de elementos finitos propriamente dito e contém métodos de acesso e manipulação de seus dados.

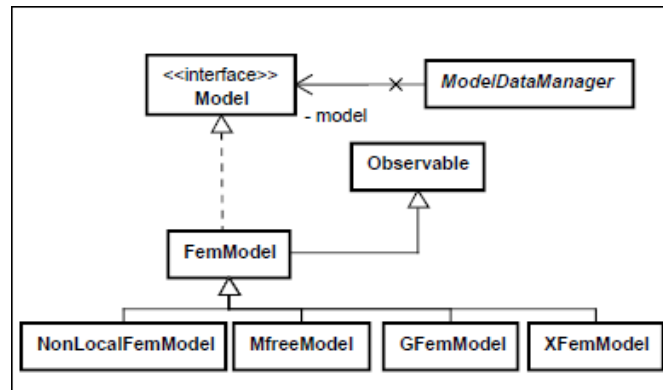


Figura 2: Hierarchy da classe Model (Penna, 2011).

Na mecânica dos sólidos computacional, para o caso de análise estática fisicamente não linear, as soluções são obtidas por meio de um processo incremental iterativo dado por:

$$K_t \cdot \delta U = \delta \lambda \cdot P + Q \quad (10)$$

onde K_t é a matriz de rigidez tangente, δU é o vetor de deslocamentos incrementais, $\delta \lambda$ é um incremento do fator de carga, e os vetores de cargas de referência e de cargas residuais do modelo são respectivamente, P e Q , sendo que, o último é obtido pela diferença entre o vetor de cargas externas e o vetor de forças equivalentes às tensões internas F_i .

No INSANE, o processo incremental iterativo segue o processo de solução do algoritmo genérico proposto por Yang e Shieh (1990), e teve sua inclusão no sistema a partir dos trabalhos de Fuina (2004), Fonseca (2006), Fonseca (2008) e Penna (2011).

Durante a solução é solicitada a montagem do tensor constitutivo necessário para a obtenção da matriz de rigidez incremental. A classe **ConstitutiveModel** é especialmente importante, pois uma vez que o modelo constitutivo recebe as informações dos elementos, do formato da integração, dos dados dos pontos materiais, das propriedades físicas do meio e do tipo de análise, todas as operações necessárias para o cálculo do operador tangente e das forças internas podem ser realizadas, sem se remeter diretamente aos elementos finitos, aos pontos de integração e nem mesmo ao modelo discreto, seja este do Método dos Elementos Finitos, ou de outras abordagens tais como o Método dos Elementos Finitos Generalizado (Alves, 2012), Método dos Elementos de Contorno (Anacleto et al., 2013) ou Métodos Sem Malha (Silva, 2012).

3.2 Implementação dos Modelos Elastoplásticos

A partir do criação do ambiente computacional proposto por Penna (2011), o modelo constitutivo passou a ser descrito por funções potenciais e seus respectivos gradientes, que resultam em um conjunto de operações tensoriais capazes de lidar com os tensores constitutivos, os gradientes das funções e as condições de carregamento e descarregamento. As operações da estrutura teórica unificada e as particularidades de cada modelo são então atribuídas a um conjunto de classes, denominadas "Filters", criadas para responder pelas parcelas que compõem as operações comuns. As implementações são tais que cada "Filter" contém as informações do respectivo modelo constitutivo que ele quer representar e as hipóteses do modelo de análise. O modelo constitutivo é tratado pelo "Filter" no formato tensorial, possibilitando ao modelo de análise fornecer as componentes de cada tensor (de rigidez, de flexibilidade, de tensão e de deformação) necessárias para sua montagem.

A Figura 3 mostra a parte da herança de classes de **ConstitutiveModel**. Observa-se que o modelo constitutivo geral, **UnifiedConstitutiveModel** possui uma instância de **UnifiedConstitutiveModelFilter**, entendida como a superclasse do conjunto de classes que formam a camada "Filter".

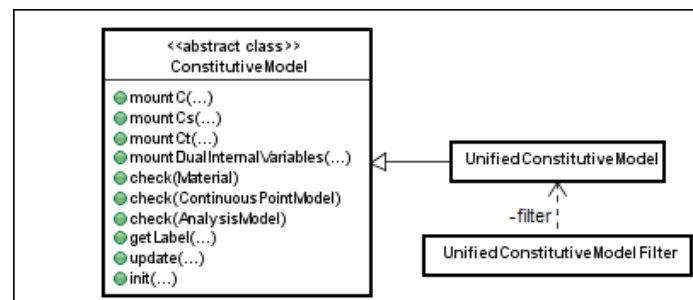


Figura 3: Diagrama de classes para **ConstitutiveModel**.

A classe de **UnifiedConstitutiveModel** se auxilia dos respectivos "Filters" para a montagem das parcelas que compõem as operações gerais da estrutura teórica unificada, tais como a obtenção do tensor de rigidez elástico ($mountC(...)$), da aproximação secante do tensor de rigidez incremental ($mountCs(...)$), a atualização das variáveis constitutivas ($update(...)$), o cálculo do tensor de tensões (realizado pelo método $mountDualInternalVariable(...)$) e do operador tangente ($mountCt(...)$). A figura 4 apresenta as atividades mais gerais do modelo constitutivo unificado.

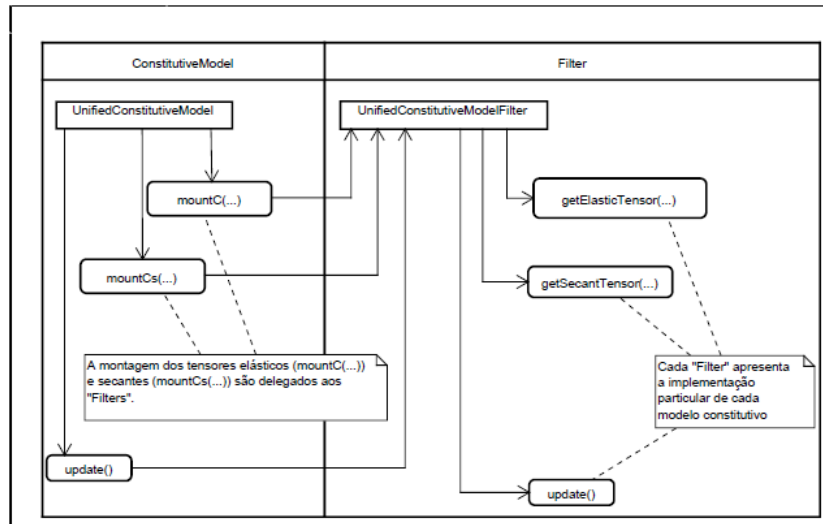


Figura 4: Obtenção dos tensores de rigidez elástico e secante e atualização das variáveis (Penna, 2011).

A herança de **UnifiedConstitutiveModelFilter** é apresentada na Fig. 5, onde estão destacadas as classes referentes aos modelos elastoplásticos com os critérios de von Mises, Rankine, Drucker-Prager e Ottosen.

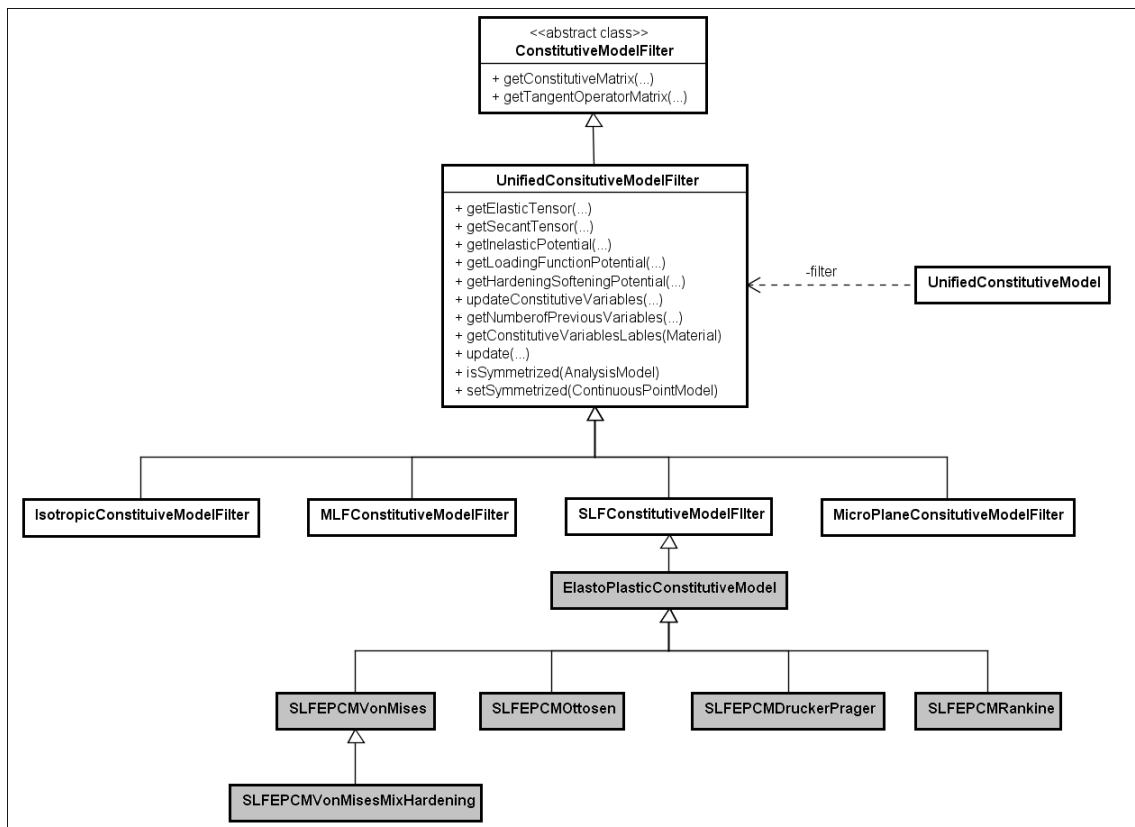


Figura 5: Diagrama de classes para UnifiedConstitutiveModelFilter.

Como pode ser visto na declaração dos métodos da Fig. 5, a classe **UnifiedConsti-**

tutiveModelFilter é responsável por efetuar as operações referentes aos modelos constitutivos, delegando as especificidades de cada modelo aos respectivos "Filters" através da solicitação das parcelas do tensor secante (método *getSecantTensor(...)*), do tensor de módulo inelástico (método *getHardeningSofteningPotential(...)*), do tensor relativo ao gradiente das funções de potenciais plásticos (método *getInelasticPotential(...)*) e do tensor dos gradientes das funções de carregamento (método *getLoadingFunctionPotential(...)*), que são utilizados no cálculo do tensor tangente.

A interação entre o modelo constitutivo e os "Filters" é descrita nas Figs. 6 e 7, através de diagramas de atividade, que apresentam o processo de obtenção do operador tangente e o cálculo das tensões, para o modelo elastoplástico de Drucker-Prager.

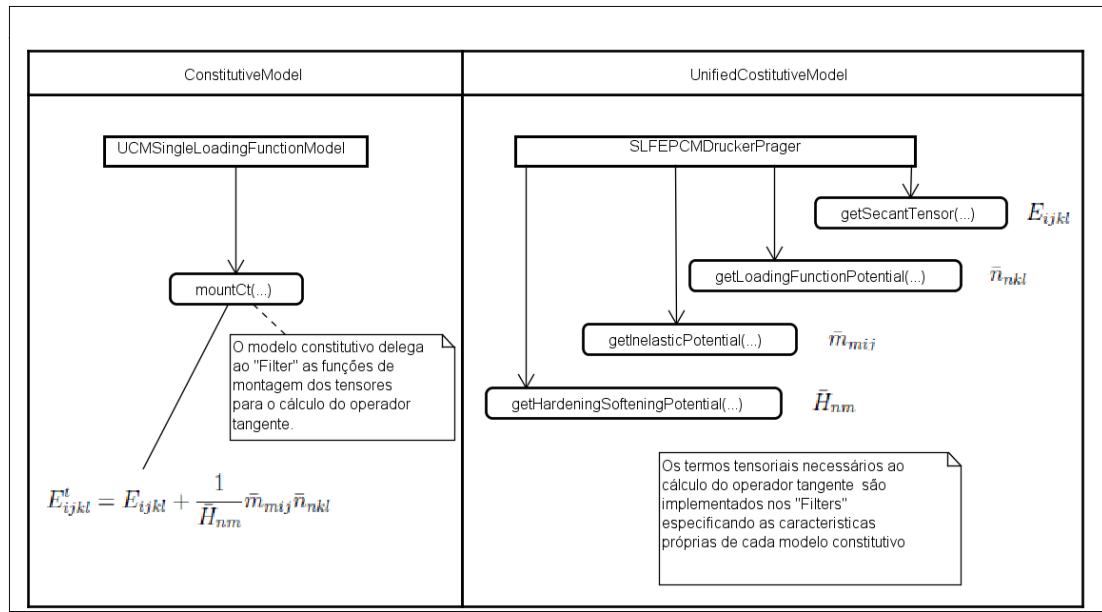


Figura 6: Cálculo do Operador Tangente. Adaptado de Penna (2011).

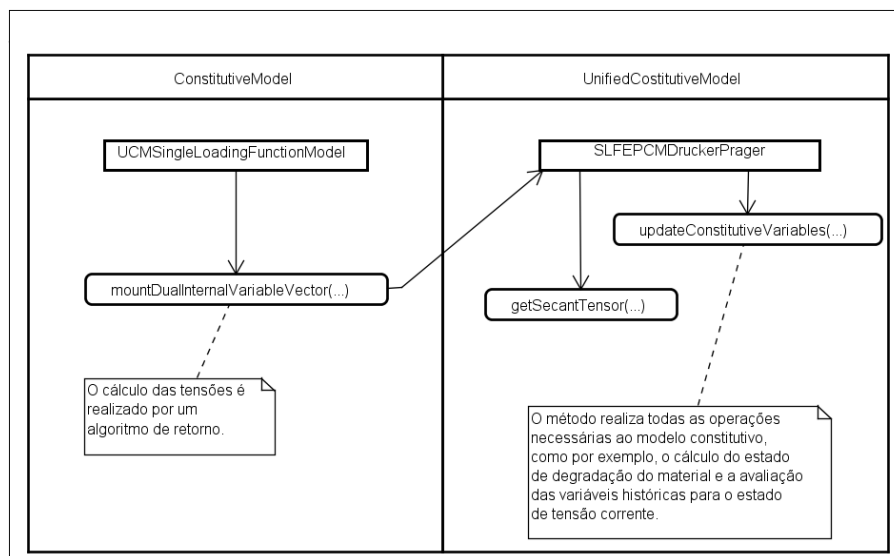


Figura 7: Cálculo das tensões. Adaptado de Penna (2011).

A Figura 8 evidencia a relação de herança entre as classes relativas aos modelos elastoplásticos e a classe abstrata **ElastoPlasticConstitutiveModelFilter**, que ocorre através da utilização de métodos como *getElasticTensor(...)* e *getComplianceTensor(...)*.

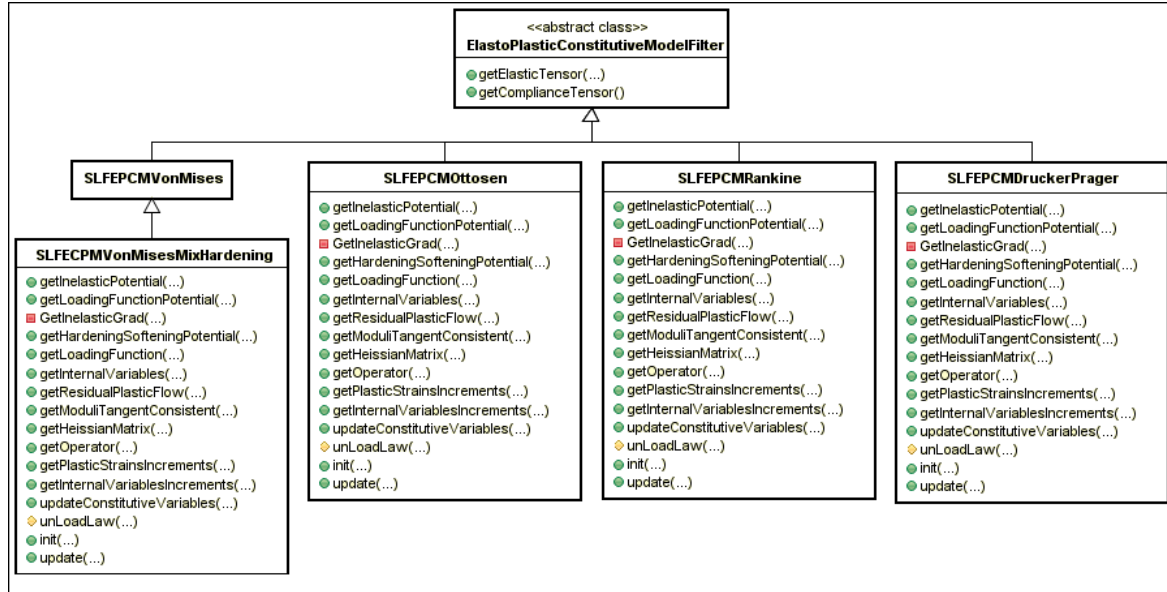


Figura 8: Diagrama de classes para **ElastoPlasticConstitutiveModelFilter**.

Na Figura 8 também observa-se os métodos implementados nos filtros, seguindo a formulação apresentada na seção 2, como por exemplo métodos *getSecantTensor(...)* (E_{ijkl}^t), método *getHardeningSofteningPotential(...)* (H), método *getInelasticPotential(...)* (m_{ij}), método *getLoadingFunctionPotential(...)* (n_{kl}). De maneira geral, os filtros são bastante parecidos, contendo também métodos com as derivadas da função de carregamento e da função de potencial plástico utilizadas no algoritmo de mapeamento de retorno, necessários para a integração numérica das relações constitutivas que regem o comportamento do material.

3.3 Classe Material

Os materiais são representados pelas classes que implementam a classe abstrata **Material** (Fig. 9) e seus atributos são um identificador e um mapa (HashMap), contendo os valores necessários para caracterizar os distintos materiais como por exemplo, módulo de elasticidade e coeficiente de Poisson.

Conforme a Fig. 9 a classe **ElastoPlasticMaterial** implementa a classe abstrata **Material**, herdando seus atributos e particularizando sua representação para materiais empregados para os modelos elastoplásticos. De maneira geral, cada modelo constitutivo está associado a um tipo de material que o represente de forma mais adequada, entretanto, alguns modelos como o de Rankine possuem os mesmos parâmetros de entrada que o modelo de von Mises e, portanto, utilizam a classe **VonMises** para representação do material.

O comportamento não linear dos materiais elastoplásticos é definido por leis de endurecimento ou amolecimento. Estas leis são tratadas pela classe abstrata **InelasticLaw** e

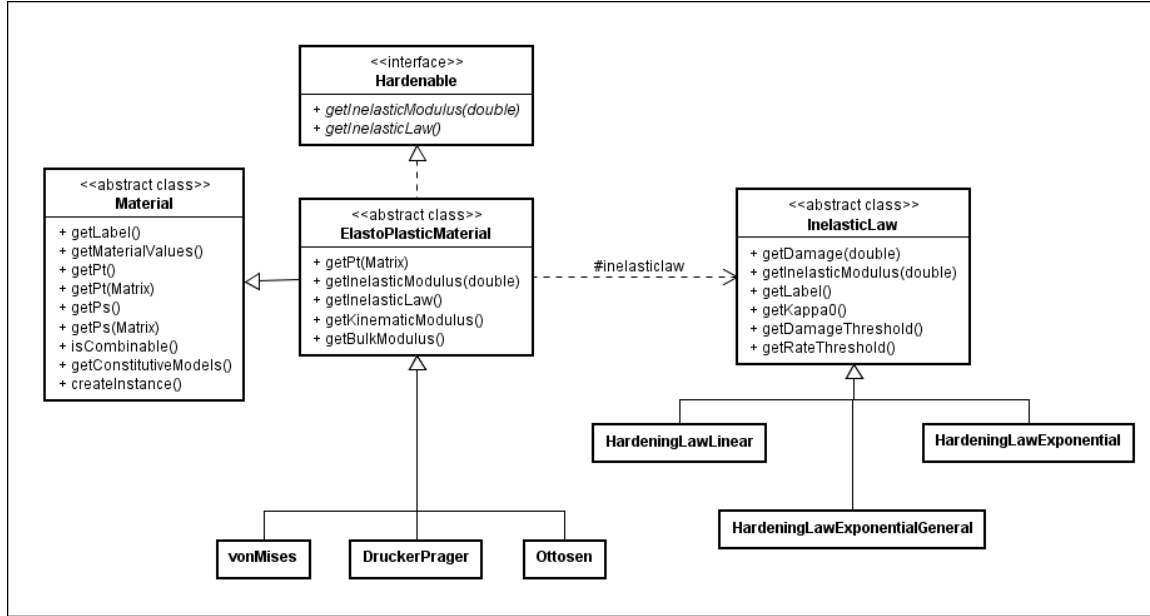


Figura 9: Diagrama de classes para Material.

suas instâncias, que podem assumir diversas formas para a evolução do módulo inelástico. Um aspecto importante a ser observado na Fig. 9 é que a classe abstrata **ElastoPlasticMaterial**, implementa a interface **Hardenable**, ao incorporar os métodos *getInelasticModulus(...)* e *getInelasticLaw(...)*, responsáveis, respectivamente, pelo cálculo do módulo inelástico específico do material e por indicar a lei inelástica utilizada na análise. A instância de **InelasticLaw** pode assumir várias formas, para as diferentes leis de endurecimento ou amolecimento implementadas por **InelasticLaw**, conforme mostra a hierarquia de classes, também ilustrada na Fig. 9.

Diversas funções de evolução do módulo inelástico têm sido propostas. Para a representação do comportamento inelástico exponencial, a função de endurecimento ou amolecimento utilizada é dada na Eq. (11) abaixo, e que foi obtida a partir da derivada da funções de evolução do módulo inelástico apresentada por Benallal et al. (1987) apud de Souza Neto et al. (2008).

$$\mathcal{H}(\kappa) = b.a.e^{-b.\kappa} \quad (11)$$

sendo a e b parâmetros do material e k a variável interna associada ao fenômeno de endurecimento ou amolecimento.

Além da função exponencial, o INSANE conta também com a função potencial apresentada por Bouchard et al. (2011) e por Vaz Jr. et al. (2011), cuja lei de endurecimento ou amolecimento é definida por:

$$\mathcal{H}(\kappa) = \frac{b.a.n}{[K + (n.\kappa)]^{1-b}} \quad (12)$$

sendo a e b parâmetros relativos ao material, K uma deformação inicial relativa ao limite elástico, κ a variável interna associada ao endurecimento ou amolecimento, n o coeficiente responsável por ajustar os valores de κ , a e b .

3.4 Algoritmo de integração das tensões

O principal problema de uma análise elastoplástica reside na determinação isolada das parcelas elástica e plástica do incremento de deformação total, resultando na necessidade de se realizar a integração da lei constitutiva, devido ao fato do módulo elastoplástico ser função da deformação plástica, decorrendo assim um processo iterativo. Pelo caráter incremental dos modelos numéricos de plasticidade, deve-se utilizar um algoritmo da classe de mapeamento de retorno capaz de obter a atualização das tensões necessária ao equilíbrio das forças internas na análise não linear.

Ortiz e Popov (1985) apontam que algoritmos eficientes devem: garantir a consistência das equações constitutivas a serem integradas; possuir estabilidade numérica; e garantir a consistência do modelo incremental. A escolha de um algoritmo impróprio pode levar não só a uma solução imprecisa para as tensões, como também pode atrasar a convergência das iterações de equilíbrio ou até mesmo levar à divergência da iteração. Atualmente o sistema conta com dois algoritmos de retorno, o método da projeção do ponto mais próximo ("closest-point projection") e o método do plano de corte ("cutting-plane").

A Figura 10 apresenta a associação entre **ElastoPlasticConstitutiveModelFilter** e as classes dos algoritmos de retorno. Tal associação é feita a partir de uma instância da classe **ReturnMappingAlgorithm** que deverá ser informada na inicialização do modelo. Assim, todas as opções de algoritmos estarão disponíveis para qualquer um dos modelos implementados.

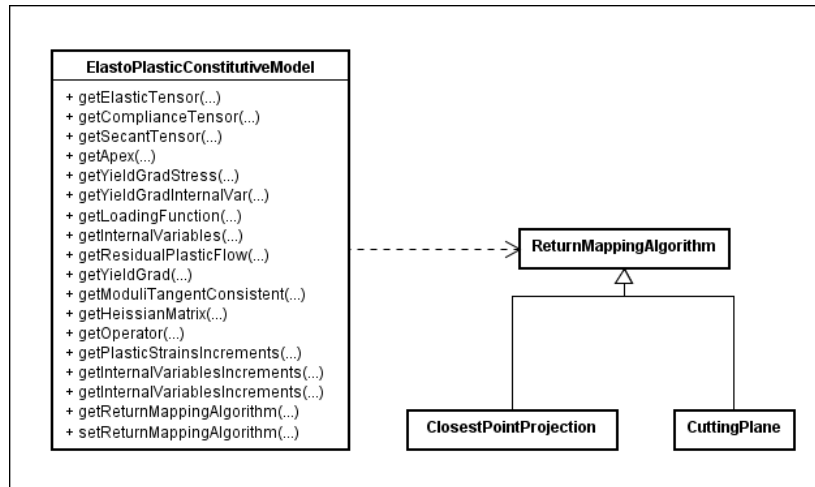


Figura 10: Diagrama de classes para ReturnMappingAlgorithm.

Os algoritmos de mapeamento de retorno foram implementados no sistema segundo a metodologia proposta por Simo e Hughes (1998). As equações para o algoritmo de integração das tensões são dadas por:

- (1) Inicialização das variáveis.

$$R_{n+1}^{(k)} := \begin{Bmatrix} -\varepsilon_{n+1}^{p(k)} + \varepsilon_n^p \\ -\alpha_{n+1}^{(k)} + \alpha_n \end{Bmatrix} + \Delta\gamma_{n+1}^{(k)} \begin{Bmatrix} \partial_{\sigma} f_{n+1} \\ \partial_q f_{n+1} \end{Bmatrix} \quad \text{e} \quad f_{n+1}^{(k)} := f(\alpha_{n+1}^{(k)}, q_{n+1}^{(k)}) \quad (13)$$

onde $\sigma_{n+1}^{(k)} := \nabla W(\varepsilon_{n+1} - \varepsilon_{n+1}^{p(k)})$, $q_{n+1}^{(k)} := -\nabla H(\alpha_{n+1}^{(k)})$ e W representa a energia elástica acumulada.

- (2) Verificação do critério de escoamento e avaliação da regra de fluxo/lei de endurecimento residuais:

$$C_{n+1}^{(k)} := \nabla^2 W(\varepsilon_{n+1} - \varepsilon_{n+1}^{p(k)}) \quad \text{e} \quad D_{n+1}^{(k)} := -\nabla^2 H(\alpha_{n+1}^{(k)}) \quad (14)$$

$$[A_{n+1}^{(k)}]^{-1} := \begin{bmatrix} [C_{n+1}^{(-1)} + \Delta\gamma_{n+1}\partial_{\sigma\sigma}^2 f_{n+1}] & \Delta\gamma_{n+1}\partial_{\sigma q}^2 f_{n+1} \\ \Delta\gamma_{n+1}\partial_{q\sigma}^2 f_{n+1} & [D_{n+1}^{(-1)} + \Delta\gamma_{n+1}\partial_{qq}^2 f_{n+1}] \end{bmatrix}^k \quad (15)$$

onde $[A_{n+1}^{(k)}]^{-1}$ é uma matriz cujos argumentos referem-se aos módulos de elasticidade e tangente.

- (3) Cálculo do módulo elástico e do módulo tangente consistente

$$\Delta^2 \gamma_{n+1}^{(k)} := \frac{f_{n+1}^{(k)} - [\partial_{\sigma} f_{n+1} \partial_q f_{n+1}]^T A_{n+1}^{(k)} R_{n+1}^{(k)}}{[\partial_{\sigma} f_{n+1} \partial_q f_{n+1}]^T A_{n+1}^{(k)} \left\{ \begin{array}{c} \partial_{\sigma} f_{n+1} \\ \partial_q f_{n+1} \end{array} \right\}^k} \quad (16)$$

- (4) Obter incrementos da deformação plástica e das variáveis internas

$$\left\{ \begin{array}{c} \Delta \varepsilon_{n+1}^{p(k)} \\ \Delta \alpha_{n+1}^{(k)} \end{array} \right\} = \left[\begin{array}{cc} [C_{n+1}^{(-1)} + \Delta\gamma_{n+1}\partial_{\sigma\sigma}^2 f_{n+1}] & 0 \\ 0 & [D_{n+1}^{(-1)} + \Delta\gamma_{n+1}\partial_{qq}^2 f_{n+1}] \end{array} \right]^k \cdot A_{n+1}^k \left[R_{n+1}^{(k)} + \Delta^2 \gamma_{n+1}^{(k)} \left\{ \begin{array}{c} \partial_{\sigma} f_{n+1} \\ \partial_q f_{n+1} \end{array} \right\}^{(k)} \right] \quad (17)$$

- (5) Atualizar as variáveis de estado e os parâmetros consistentes

$$\varepsilon_{n+1}^{p(k+1)} = \varepsilon_n^{p(k)} + \Delta \varepsilon_{n+1}^{p(k)} \quad , \quad \alpha_{n+1}^{p(k+1)} = \alpha_n^{p(k)} + \Delta \alpha_{n+1}^{p(k)} \quad \text{e} \quad \Delta \gamma_{n+1}^{p(k+1)} = \Delta \gamma_n^{p(k)} + \Delta^2 \gamma_{n+1}^{p(k)} \quad (18)$$

O processo iterativo Closest-Point Projection tem como principal desvantagem a necessidade de se computar os gradientes da regra de fluxo e da lei de encruamento (Eq. 19), pois para modelos de plasticidade complexos esta tarefa pode ser extremamente trabalhosa.

$$\left[\begin{array}{cc} \partial_{\sigma\sigma}^2 f(\sigma, q) & \partial_{\sigma q}^2 f(\sigma, q) \\ \partial_{q\sigma}^2 f(\sigma, q) & \partial_{qq}^2 f(\sigma, q) \end{array} \right] \quad (19)$$

A Figura 11 mostra um fluxograma para o algoritmo implementado na classe **ClosestPointProjection** proposto por Simo e Hughes (1998) visando ilustrar a estrutura dos algoritmos de mapeamento de retorno implementados.

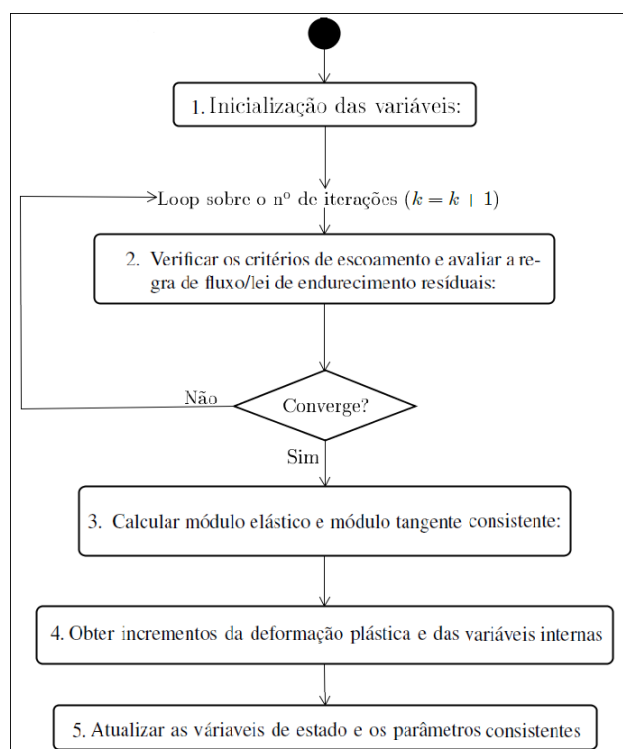


Figura 11: Diagrama de classes para ReturnMappingAlgorithm.

4 EXEMPLOS NUMÉRICOS

Nesta seção, os modelos constitutivos implementados são apresentados em um exemplo de tração, cujo objetivo é ilustrar cada modelo, ressaltando as características individuais de cada um e visando mostrar algumas das possíveis aplicações.

As simulações estão organizadas de maneira que se possa observar a variação da lei inelástica do material com situações de endurecimento, amolecimento e plasticidade perfeita, adequando-se os parâmetros das funções para estes comportamentos e adotando as funções inelásticas linear, exponencial e potencial implementadas no sistema INSANE.

Será considerada uma chapa retangular em estado plano de tensão, adotando-se a malha com 64 elementos quadrilaterais de 4 nós de tamanho 0,125m x 0,125m, cuja configuração de geometria, de carga, as condições contorno estão apresentadas na Fig. 12.

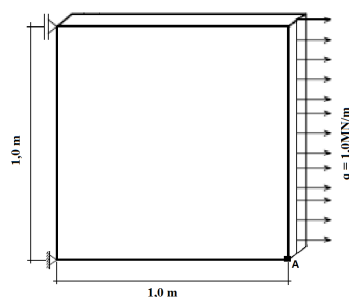


Figura 12: Condições de contorno, geometria e carga para simulação de tração uniaxial.

Os dados apresentados para as leis inelásticas variam em cada modelo, sendo válido ressaltar que, durante esta análise não se teve a preocupação de simular dados reais, pois o objetivo é apenas apresentar o funcionamento dos modelos. Os parâmetros de cada modelo constitutivo são listados abaixo:

Critério de Rankine: $E = 2,0 \times 10^4 MPa$, $\nu = 0,2$, $\sigma_y = 2MPa$ e lei inelástica linear;

Critério de Drucker-Prager: $E = 2,0 \times 10^4 MPa$, $\nu = 0,2$, $\sigma_y = 2MPa$ e lei inelástica exponencial;

Critério de von Mises com Encruamento isotrópico: $E = 1,0 \times 10^5 MPa$, $\nu = 0,2$, $\sigma_y = 50MPa$, $H = 500MPa$ e lei inelástica linear;

Critério de von Mises com Encruamento Cinemático: $E = 1,0 \times 10^5 MPa$, $\nu = 0,2$, $\sigma_y = 50MPa$, $H = 1000MPa$ e lei inelástica linear;

Critério de von Mises com Encruamento Misto: $E = 1,0 \times 10^5 MPa$, $\nu = 0,2$, $\sigma_y = 50MPa$, $H = 1500MPa$ e lei inelástica linear;

Critério de Ottosen: $E = 32129MPa$, $\nu = 0,2$, $\sigma_y = 2MPa$;

A análise foi realizada com aproximação tangente e tensor constitutivo contínuo. Para solução das equações não lineares foi adotado método de controle direto de deslocamento, em que o deslocamento horizontal do ponto A foi controlado com incremento de $0,0005m$ por passo, com tolerância em deslocamento de 1×10^{-4} e para a integração das tensões foi utilizado o algoritmo de retorno Closest Point Projection.

As trajetórias de equilíbrio foram obtidas e, para cada modelo, as curvas "Fator de carga $\times$ Deslocamento Horizontal" da extremidade da chapa estão apresentadas. A Figura 13 reproduz o comportamento de endurecimento (módulo inelástico $H = 2000MPa$), amolecimento (módulo inelástico $H = -2000MPa$) e plasticidade perfeita (módulo inelástico $H = 0MPa$) para o modelo de Rankine.

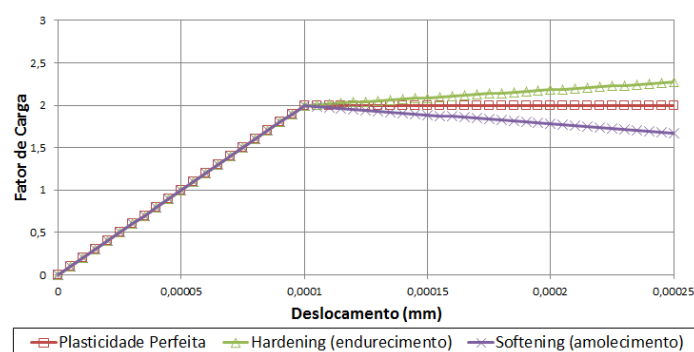


Figura 13: Curvas Fator de Carga x Deslocamento para variação do módulo inelástico.

A resposta do modelo está diretamente associada à lei inelástica adotada, com as funções dependendo de um conjunto de parâmetros que devem ser prescritos para descrever o comportamento do material. Através do modelo de Drucker-Prager buscou-se ilustrar as leis inelásticas exponencial implementada no sistema. Os resultados estão apresentadas na Fig. 14.

A partir da Fig. 14, podemos observar diferentes curvas obtidas através da variação dos parâmetros a e b da Eq. (11). Com estas variações buscou-se ilustrar tanto o compor-

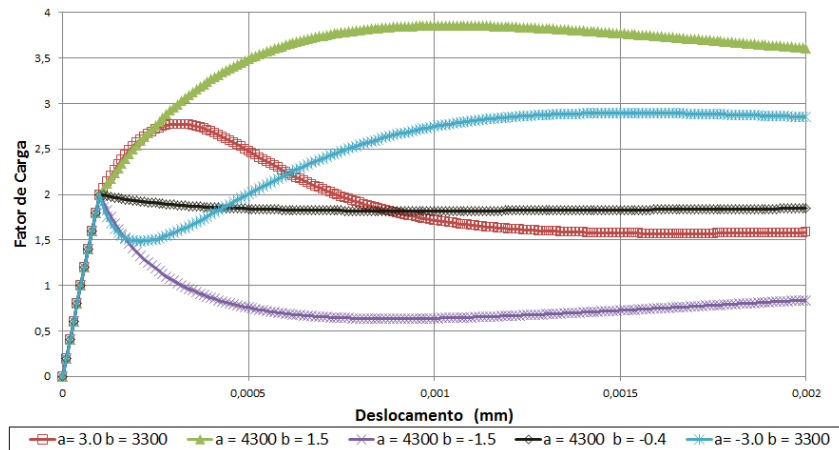


Figura 14: Curvas Fator de Carga x Deslocamento para variações dos parâmetros da lei inelástica exponencial.

tamento de amolecimento quanto o de endurecimento, além de apresentar a sensibilidade das curvas aos parâmetros.

Os modelos constitutivos para função de von Mises são apresentados com o objetivo de ilustrar o comportamento da chapa quando submetida aos variados tipos de encruamento (misto, isotrópico e cinemático). As trajetórias de equilíbrio obtidas estão apresentadas através das curvas "Fator de carga x Deslocamento Horizontal" da extremidade da chapa na Fig. 15.

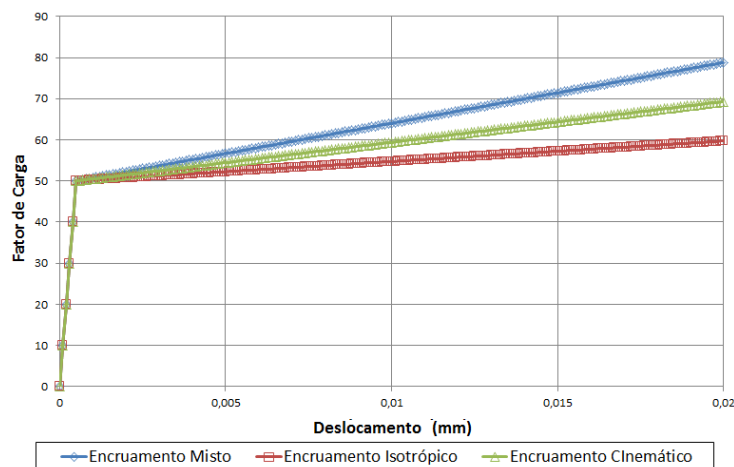


Figura 15: Curvas Fator de Carga x Deslocamento para encruamento isotrópico, cinemático e misto

As superfícies de escoamento da chapa foram geradas a partir dos valores das tensões obtidos. Na Figura 16 pode-se observar claramente a superfície elíptica de escoamento inicial com critério de von Mises, antes que qualquer carregamento fosse aplicado. Após a aplicação do carregamento tem-se três outras superfícies, uma para cada tipo de encruamento (isotrópico, cinemático e misto) geradas a partir da implementação.

Os resultados obtidos mostraram que, quando aplicado o encruamento isotrópico a superfície sofreu expansão uniforme, mantendo a sua posição de centro inalterada. Quando

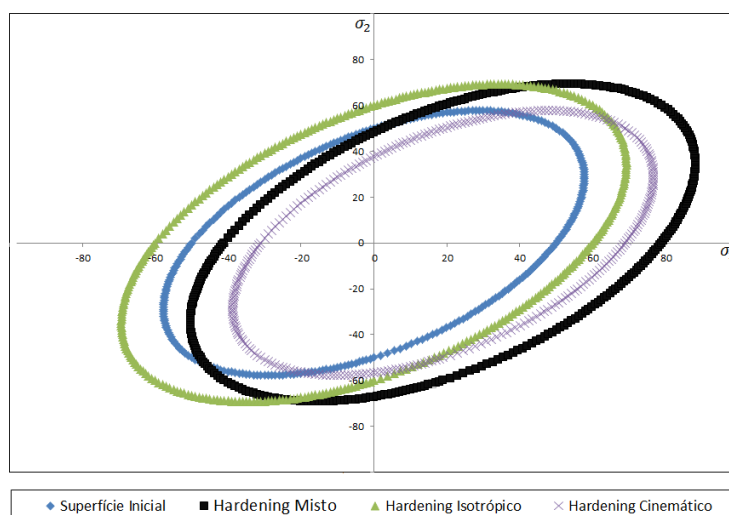


Figura 16: Superfície de escoamento de von Mises.

aplicado o encruamento cinemático, a superfície manteve sua forma e foi translada na direção de aplicação da carga. A partir da consideração do módulo de encruamento misto no problema, a superfície a sua forma e tamanho alterados (conforme encruamento isotrópico) e também sofreu a translação (conforme encruamento cinemático).

A Figura 17 apresenta os resultados obtidos com o modelo de Ottosen, um critério de falha de quatro parâmetros válido para uma grande variedade de combinação de tensões, e desenvolvido para reproduzir matematicamente as relações tensão-deformação para o concreto. Neste modelo, buscou-se apresentar o comportamento das leis inelásticas em um ensaio de compressão uniaxial.

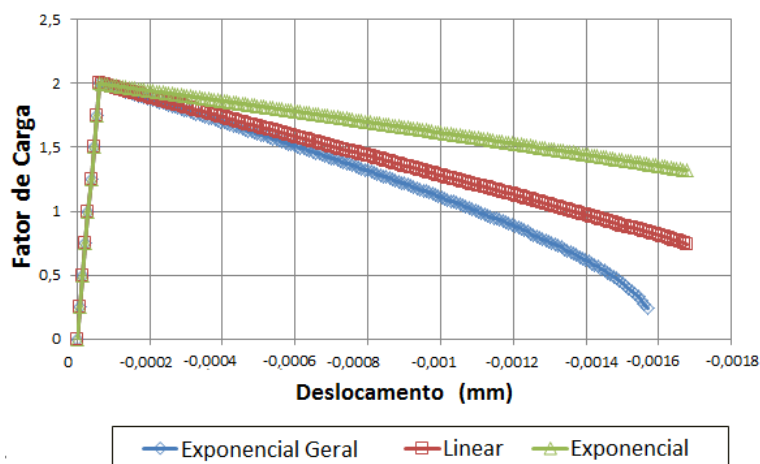


Figura 17: Curvas Fator de Carga x Deslocamento para variação da lei inelástica em ensaio de compressão uniaxial.

Ressalta-se que as condições de geometria e contorno apresentadas anteriormente foram mantidas, alterando o sentido da carga (passando para compressão). Outro aspecto alterado foi o algoritmo de mapeamento de retorno utilizando-se o Cutting Plane.

5 CONCLUSÃO

Este trabalho apresentou o projeto orientado a objetos relativo à implementação dos modelos constitutivos elastoplásticos, segundo o Ambiente Teórico-Computacional para Modelos Constitutivos desenvolvido por Penna (2011). Pode-se observar o grande potencial da biblioteca de modelos constitutivos do sistema computacional INSANE que se dá em virtude da generalidade das implementações dos modelos e dos diversos recursos para modelagem constitutiva existentes nesta plataforma.

Os Modelos Elastoplásticos de von Mises com encruamento misto, Drucker-Prager, Rankine e Ottosen foram utilizados nas simulações numéricas apresentadas. Em geral, as análises numéricas buscaram apresentar os modelos constitutivos, além de apresentar a variação das leis inelásticas com funções linear, exponencial. Pelos exemplos numéricos apresentados pode-se concluir que os modelos implementados conseguem simular ensaios de tração e compressão para variados valores de endurecimento e amolecimento.

A continuidade dessa pesquisa refere-se à implementação dos modelos constitutivos elastoplásticos clássicos de Tresca e Mohr-Coulomb, além de novos modelos para simulação do concreto que possam representar o material de forma coerente com os resultados experimentais. Outra implementação importante a ser realizada está relacionada ao carregamento cíclico, que é de suma importância para análises mais elaboradas além de ilustrar de forma mais clara os encruamentos cinemático e misto.

AGRADECIMENTOS

Os autores agradecem o apoio importante das agências de pesquisa brasileira CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico - Processo: 308785 / 2014-2) e FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais - Processo: PPM-00669-15)

REFERÊNCIAS

- Ajeje, F. H., 2009. *Abordagem Orientada a Objetos para Implementação Computacional de Elementos Finitos de Cascas Planas*. Dissertação (Mestrado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Almeida, M. L., 2005. *Elementos Finitos Paramétricos Implementados em JAVA*. Dissertação (Mestrado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Alves, P. D., 2012. *Estratégia Global-Local Aplicada ao Método dos Elementos Finitos Generalizados*. Dissertação (Mestrado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Anacleto, F. E. S., Peixoto, R. G., Penna, S. S., Ribeiro, G. O., Pitangueira, R. L. S., Ribeiro, T. S. A., 2013. *Non-Linear Solution Algorithm for the Implicit Boundary Element Method Concerning Various Control Methods*. XXXIV Iberian Latin-American Congress on Computational Methods in Engineering.
- Benallal, A., Billardon, R., Daghri, I. e Moret-Bailly, L., 1987. *Crack initiation and propagation analyses taking into account initial strain hardening and damage fields*, Fourth Int. Conf. of Numerical Methods in Fracture Mechanics, Texas, EUA.

- Bouchard, P.-O., Bourgeon, L., Fayolle, S. e Mocellin, K., 2011. *An enhanced Lemaitre model formulation for materials processing damage computation*. Int J Mater Form, vol. 4, pp. 299 - 315.
- Carol, I., Rizzi, E. e Willam, K., 1994. *A unified theory of elastic degradation and damage based on a loading surface*. International Journal of Solids Structures, vol. 31, pp. 2835-2865.
- Chen, W. F., Han, D. J., 1988. *Plasticity for Structural Engineers*. Nova York, EUA: Springer-Verlag.
- Fonseca, F. T., 2008. *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.
- Fonseca, M. T., 2006. *Aplicação Orientada a Objetos para Análise Fisicamente Não-Linear com Modelos Reticulados de Seções Transversais Compostas*. Dissertação (Mestrado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Fuina, J. S., 2004. *Métodos de Controle de Deformações para Análise Não-Linear de Estruturas*. Dissertação (Mestrado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Fuina, J. S., 2009. *Formulações de Modelos Constitutivos de Microplanos para Contínuos Generalizados*. Tese (Doutorado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Ortiz, M., Popov, E. P, 1985. Accuracy and stability of integration algorithms for elasto-plastic constitutive relations. International Journal of Numerical Methods in Engineering, v. 21, p. 1561–1576, 1985.
- Penna, S. S., 2011. *Formulação Multipotencial para Modelos de Degradação Elástica: Unificação Teórica, Proposta de Novo Modelo, Implementação Computacional e Modelagem de Estruturas de Concreto*. Tese (Doutorado), Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Saliba, S. S., 2007. *Implementação Computacional e Análise Crítica de Elementos Finitos de Placas*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Silva, R. P., 2012. *Análise Não-Linear de Estruturas de Concreto por meio do Método Element Free Galerkin*. Tese (Doutorado) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Simo, J. C., Hughes, T. J. R., 1998. *Computational Inelasticity*. Nova York, EUA: Springer-Verlag.
- Souza Neto, E. A., Peric, D., Owen, D. R. J., 2008. *Computational Methods For Plasticity: Theory and Application*. Swansea, EUA: Wiley.
- Vaz Jr., M., Muñoz-Rojas, P. A., Langes, M. R., 2011. *Damage evolution and thermal coupled effects in inelastic solids*. International Journal of Mechanical Sciences, vol. 53, pp. 387-398.