



**UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

**ESTUDO DE MODELOS DE CAMPO DE FASE PARA ANÁLISE DE
FRATURA**

Rafael Gollner Bayão

Belo Horizonte
2021

Rafael Gollner Bayão

**ESTUDO DE MODELOS DE CAMPO DE FASE PARA
ANÁLISE DE FRATURA**

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Mecânica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do grau de bacharel em Engenharia Mecânica.

Orientador: Roque da Silva Pitangueira

Belo Horizonte
2021

Dedicado à minha família que sempre me apoiou

AGRADECIMENTOS

Ao Professor Roque da Silva Pitangueira que me abriu portas ao projeto INSANE, me guiou no desenvolvimento desse trabalho e introduziu ao campo de mecânica de falhas computacional.

A todos integrantes do grupo de desenvolvimento do INSANE que tornaram possível a realização desse trabalho.

RESUMO

O desenvolvimento de modelos que utilizam o Método dos Elementos Finitos para a análise computacional do processo de fratura é um campo de pesquisa muito vasto. Podem-se citar diversas formulações, publicadas ao longo das últimas décadas, que possibilitam descrever e prever o comportamento de estruturas com variados materiais, condições e propriedades físicas. Nesse sentido, os modelos de Degradação Elástica possuem uma extensa bibliografia e formulações com resultados excelentes para várias situações de análises. Apesar disso, recentemente, os modelos de Campo de Fase vêm ganhando destaque por sua capacidade de prever as trajetórias de trinca de forma mais precisa e alcançarem resultados igualmente satisfatórios. Dessa forma, em vista a publicações recentes no desenvolvimento de modelos de Campo de Fase, esse trabalho tem como objetivo fazer uma revisão bibliográfica e implementações de formulações teóricas, junto a métodos de solução, modelos constitutivos e funções características desse método para análises quase-estáticas. Desse modo, incorporando novas ferramentas a um programa de simulação numérica, tal que seja possível realizar comparações de resultados entre modelos de Campo de Fase com os de Degradação Elástica.

Palavras-chave: Método dos elementos finitos. Desenvolvimento de software. Modelos constitutivos. Campo de fase. Degradação elástica;

ABSTRACT

The development of models that use the Finite Element Method formulation to computational fracture analysis is a wide research field. It can be cited several formulations, which were published throughout the last decades, that allow the description and prediction of the behavior from structures with various materials, conditions, and physical properties. In this sense, the Elastic Degradation models have vast literature and formulations with exceptional results for many analysis situations. Besides that, recently, the Phase-field models have been gaining a spotlight for their capacity to predict crack paths precisely and achieving also great results. Therefore, due to recent publications in Phase-field models development, this work propose is making a literature review and implementation of theoretic formulations of this method, along with solution methods, constitutive models, and characteristic functions for quasi-static analysis. Hence, embodying new tools to a numerical simulation program, so it may be possible to perform a comparative analysis between the results from both Phase-field and Elastic Degradations models.

Keywords: Finite Element Methods. Software development. Constitutive models. Phase-field; Elastic degradation;

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 2.1 – Relações de tensão/deformação para diferentes modelos. a) Modelo linear elástico. b) Modelo idealmente plástico. c) Modelo com amolecimento. | 14 |
| Figura 2.2 – Curvas de carregamento. a) Carregamento elástico. b) Carregamento inelástico. c) i. Descarregamento; ii. Recarregamento. | 15 |
| Figura 2.3 – Trinca discreta e trinca difusa com seus respectivos domínios | 18 |
| Figura 2.4 – Região de degradação em domínio unidimensional com $l_0 = 1$ | 19 |
| Figura 2.5 – Valores da função $\alpha(\phi)$ para diferentes ξ | 20 |
| Figura 3.1 – O formato “.msh” | 34 |
| Figura 3.2 – Exemplificação do rotulamento de blocos para nós | 35 |
| Figura 3.3 – Composição de uma interface implementada em SWING | 36 |
| Figura 3.4 – A estrutura de semi-arestas | 37 |
| Figura 3.5 – Fluxograma de Projeto | 39 |
| Figura 3.6 – A janela <i>GmshAttributesAssistantDialog</i> . Alguns botões permanecem inativos enquanto não possuem entidades selecionadas. | 40 |
| Figura 3.7 – A janela <i>SelectionAssistant</i> | 41 |
| Figura 3.8 – Interface gráfica para funções de campo de fase: (a) funções sem valores a serem informados; (b) funções com valores a ser informados | 43 |
| Figura 3.9 – Hierarquia das funções de degradação energéticas de Wu (2017) | 43 |
| Figura 4.1 – Teste uniaxial de tração: (a) modelo de análise; (b) malha; | 47 |
| Figura 4.2 – Teste de tração com lei de amolecimento linear | 48 |
| Figura 4.3 – Teste de tração com lei de amolecimento de Cornelissen | 49 |
| Figura 4.4 – ϕ em teste uniaxial de tração: (a) lei de Cornelissen; (b) Lei linear; | 49 |
| Figura 4.5 – Painel em L de Wu (2017): (a) Esquema de montagem; (b) Caminho da trinca; (c) Curvas de resposta para nó de controle | 50 |
| Figura 4.6 – Flexão de 3 pontos: (a) modelo; (b) malha de Penna (2011); (c) malha utilizada. | 52 |
| Figura 4.7 – Parametrização flexão de 3 pontos | 53 |
| Figura 4.8 – Resultados para análise de flexão de 3 pontos | 53 |
| Figura 4.9 – Campo de fase para lei (a) de Cornelissen; (b) exponencial. | 54 |
| Figura 4.10–Comparação de resultados para diferentes valores de l_0 | 54 |
| Figura 4.11–Cisalhamento 4 pontos: montagem da análise. | 55 |
| Figura 4.12–Cisalhamento 4 pontos: calibração do valor de l_0 | 55 |
| Figura 4.13–Cisalhamento 4 pontos: malha utilizada, nós em rosa com restrição $\phi = 0$ | 56 |
| Figura 4.14–Resultados Cisalhamento 4 pontos | 56 |
| Figura 4.15–Cisalhamento 4 Pontos: (a) distribuição da variável de campo; (b) tensões de cisalhamento na estrutura. | 57 |

| | |
|---|----|
| Figura 4.16–Cisalhamento 4 pontos com outras propriedades | 57 |
| Figura 4.17–Cisalhamento 4 pontos com outras propriedades | 58 |
| Figura 4.18–Painel em L de Penna (2011): (a) esquema de montagem; (b) malha para fissuração distribuída; (c) malha para campo de fase. | 59 |
| Figura 4.19–Resultados para análise do Painel em L | 59 |
| Figura 4.20–Outros resultados para análise do Painel em L | 60 |
| Figura 4.21–Destaque da distribuição da variável ϕ no Painel em L: (a) lei de amoleci- mento de Cornelissen; (b) Lei de amolecimento Exponencial. | 61 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Funções de geometria da trinca e suas propriedades | 19 |
| Tabela 2 – Funções de degradação energética | 21 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|---------------------------------|
| HEDS | <i>Half-Edge Data Structure</i> |
| MEF | Método dos Elementos Finitos |
| POO | Programação Orientada a Objetos |

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 12 |
| 1.1 | OBJETIVO GERAL | 13 |
| 1.2 | OBJETIVOS ESPECÍFICOS | 13 |
| 2 | REVISÃO BIBLIOGRÁFICA | 14 |
| 2.1 | MODELOS DE DEGRADAÇÃO ELÁSTICA | 14 |
| 2.1.1 | Modelos de Dano | 15 |
| 2.1.2 | Modelos de Fissuração Distribuída | 16 |
| 2.1.3 | Relações Tensão-Deformação | 16 |
| 2.2 | MODELOS DE CAMPO DE FASES | 17 |
| 2.2.1 | A Teoria de Griffith | 17 |
| 2.2.2 | Modelagem de trincas difusas | 18 |
| 2.2.2.1 | <i>Função de geometria da trinca</i> | 18 |
| 2.2.3 | Degradação da energia no campo | 20 |
| 2.2.3.1 | <i>Funções de degradação energética</i> | 20 |
| 2.2.3.2 | <i>Degradação da energia baseada em propriedades do material</i> | 21 |
| 2.2.3.3 | <i>Leis de amolecimento</i> | 22 |
| 2.2.4 | Montagem do modelo em Elementos Finitos | 24 |
| 2.2.4.1 | <i>Equações na Forma Fraca</i> | 24 |
| 2.2.4.2 | <i>Discretização em MEF 2D</i> | 24 |
| 2.2.5 | Métodos de Solução | 26 |
| 2.2.5.1 | <i>Método monolítico</i> | 27 |
| 2.2.5.2 | <i>Método alternado com variável histórica</i> | 27 |
| 2.2.5.3 | <i>Método alternado com restrição de contorno</i> | 28 |
| 2.2.6 | Modelos Constitutivos | 29 |
| 2.2.6.1 | <i>Modelos sem separação de esforços</i> | 29 |
| 2.2.6.2 | <i>Modelos com separação de esforços</i> | 29 |
| 2.2.6.3 | <i>Modelos híbridos</i> | 30 |
| 2.3 | CONCEITOS BÁSICOS DE PROGRAMAÇÃO | 31 |
| 2.3.1 | Programação Orientada a Objetos | 31 |
| 2.3.2 | Hereditariedade | 31 |
| 2.3.3 | Interfaces | 32 |
| 2.4 | PONTOS RELEVANTES A RESPEITO DO TEMA | 32 |
| 3 | METODOLOGIA | 33 |
| 3.1 | IMPORTAÇÃO DE MALHAS E PARÂMETROS DO MODELO | 33 |
| 3.1.1 | Processo de implementação | 33 |
| 3.1.1.1 | <i>O programa GMSH</i> | 33 |

| | | |
|----------|---|-----------|
| 3.1.1.2 | <i>O formato “.msh”</i> | 33 |
| 3.1.1.3 | <i>Inserção de atributos</i> | 34 |
| 3.1.2 | Pontos específicos sobre código no INSANE | 35 |
| 3.1.2.1 | <i>Padrão “Command”</i> | 35 |
| 3.1.2.2 | <i>A biblioteca SWING</i> | 36 |
| 3.1.2.3 | <i>A Estrutura de Dados de Geometria</i> | 37 |
| 3.1.2.4 | <i>Classes essenciais à aplicação</i> | 37 |
| 3.1.3 | Código implementado | 38 |
| 3.1.3.1 | <i>Classes implementadas para importação de malhas</i> | 38 |
| 3.1.3.2 | <i>Classes implementadas para inserir atributos do modelo</i> | 40 |
| 3.1.3.3 | <i>A janela “Attributes Assistant”</i> | 40 |
| 3.1.3.4 | <i>A janela “Selection Assistant”</i> | 41 |
| 3.2 | IMPLEMENTAÇÃO DE MODELOS DE CAMPO DE FASE | 42 |
| 3.2.1 | Funções com valores definidos pelo usuário | 42 |
| 3.2.2 | Implementação das leis de amolecimento | 43 |
| 3.2.2.1 | Implementação dos métodos de solução | 43 |
| 3.2.2.2 | Implementação de modelos constitutivos | 46 |
| 4 | ANÁLISES E RESULTADOS | 47 |
| 4.1 | VALIDAÇÃO DAS IMPLEMENTAÇÕES | 47 |
| 4.1.1 | Validação com tração unidimensional | 47 |
| 4.1.2 | Validação com Painel em L | 48 |
| 4.2 | EXEMPLOS NUMÉRICOS | 51 |
| 4.2.1 | Flexão em 3 Pontos | 51 |
| 4.2.2 | Cisalhamento em 4 Pontos | 54 |
| 4.2.3 | Painel em L | 58 |
| 5 | CONCLUSÃO | 62 |
| | Referências | 64 |

1 INTRODUÇÃO

O Método dos Elementos Finitos (MEF) é um método de análise numérica que possibilita encontrar resultados aproximados para problemas de engenharia baseados na solução de equações diferenciais parciais. Segundo Logan (2015), o MEF teve início na década de 40 com o desenvolvimento de métodos para cálculo de esforços em vigas e barras unidimensionais, mas ao longo dos anos foi sendo aperfeiçoado e hoje abrange diversas áreas da engenharia, como flambagem, análise de vibrações, impacto, transferência de calor, escoamento de fluidos, campos elétricos e magnéticos. Na área de mecânica de falhas, o MEF é aplicado em modelos como os de Fissuração Discreta ou Distribuída, de Dano, de Degradação Elástica, Campo de Fase, entre outros.

Nesse aspecto, há uma crescente especialização e exploração de teorias cada vez mais complexas, no intuito de encontrar resultados mais precisos. Logan (2015) afirma que os programas que oferecem recursos do MEF possuem duas linhas de abordagem: ser generalistas, focando em métodos já consagrados, ou especiais, focando em áreas específicas. Entretanto, segundo Wu et al. (2020), alguns modelos que demandam grande processamento têm seu uso restrito ao estudo acadêmico, pois, mesmo com o avanço rápido do processamento de computadores, ainda são muito custosos para grandes aplicações. Dessa forma, são raros os softwares comerciais que disponibilizam novos modelos de falha ou que permitam implementações de novos recursos diretamente no código.

Para entender os modelos de mecânica de falhas computacional, deve-se conhecer os princípios da Mecânica do Dano e Mecânica da Fratura, teorias que estudam o processo de falha do material. Na primeira, apresentada pela primeira vez por Kachanov (1958), o material é visto como contínuo e o processo de falha é interpretado pela degradação das propriedades mecânicas geradas por micro trincas em sua extensão. Já na Mecânica da Fratura é considerada a existência de uma trinca inicial e são estudadas as condições que levam ao crescimento dessa. De acordo com Wu et al. (2020), os dois modelos mais conhecidos da área são o *Linear Elastic Fracture Mechanics* (Griffith (1921)), que calcula a energia de fratura pela relação da energia de separação atômica e de geração de superfície de fratura, e o *Cohesive Zone Model*, (Dugdale (1960)), que define uma zona de forças coesivas na extremidade da trinca.

O estudo das condições que levam estruturas e materiais a falharem tem grande importância no contexto de engenharia, pois conhecer as características e limites em cada aplicação é o que permite o desenvolvimento de projetos mais complexos e que suprem as demandas cada vez mais exigentes do mercado. Nessa linha, torna-se possível projetar estruturas de forma mais eficiente, reduzindo a dependência de testes experimentais que além de demandar tempo podem ser caros e gerar lixo, e permite que se atenda condições mais rígidas de segurança e confiabilidade. Para alcançar esse objetivo, buscam-se modelos e recursos computacionais que possibilitem o usuário a alcançar resultados mais precisos e que diminuam o tempo de processamento. No

quesito da redução do custo computacional, o uso de malhas adaptativas tem grande destaque. Esse método permite um refinamento dinâmico da malha durante o processo de cálculo nas regiões mais críticas, permitindo o uso de malhas inicialmente grosseiras. No que se refere à busca por resultados mais precisos, os modelos têm se aprimorado com a inclusão de diferentes fenômenos, podendo-se citar modelos que consideram o comportamento desigual sob tensões volumétricas e deviatórias ou a assimetria das deformações quando aplicadas tensões de tração ou compressão. Miehe et al. (2010), por exemplo, desenvolve um modelo constitutivo que permite eliminar o desenvolvimento de trincas no modelo quando sujeito à compressão.

Este trabalho dedica-se aos modelos de Campo de Fase e a visualização de seus resultados em comparação a modelos de Degradação Elástica. Ambos utilizam do MEF para tentar prever o comportamento de um corpo em processo de falha, mas utilizam de abordagens distintas. Os modelos de Degradação Elástica se respaldam na Mecânica do Dano, o que permite representar a redução das propriedades elásticas do meio material, como o módulo de elasticidade e, consequentemente, a redução da rigidez do sistema estrutural dele constituído. Os modelos de Campos de Fase baseiam-se na Teoria de Griffith, bastante aplicada em problemas da Mecânica da Fratura Linear Elástica. A formulação variacional dessa teoria permite a representação do processo de degradação do contínuo por meio da introdução de campos de fase representativos dos estados degradados nas diferentes regiões dos sólidos. Neste trabalho, será utilizado o software de código aberto INSANE (*Interactive Structural Analysis Environment*) para desenvolver um estudo dos modelos de campo de fase.

1.1 OBJETIVO GERAL

Desenvolver aplicações e métodos para o sistema INSANE que viabilize seu uso em simulações numéricas em MEF baseadas nas teorias de Campos de Fase para análise de fratura e estudar o resultado para diferentes formulações presentes na literatura.

1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver uma aplicação que possibilite a importação de malhas de software externo e com código livre para o sistema INSANE.
- Desenvolver métodos para que a aplicação permita introduzir os atributos de modelo disponibilizados pelo sistema INSANE nas malhas importadas.
- Desenvolver um novo algoritmo de solução numérica que permita o uso de novos modelos de Campo de Fase no INSANE.
- Implementar modelos constitutivos e funções características para modelos de Campo de Fase no sistema INSANE que incluam os limites de resistência do material.
- Utilizar a aplicação para comparar resultados dos modelos implementados com modelos de fissuração distribuída presentes na literatura.

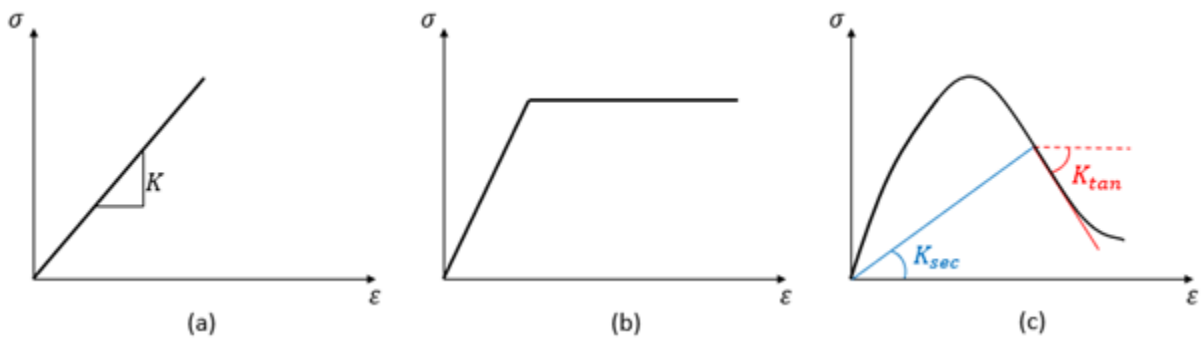
2 REVISÃO BIBLIOGRÁFICA

O MEF em sua abordagem clássica para problemas de mecânica dos sólidos busca relacionar os deslocamentos nodais a partir dos carregamentos aplicados à estrutura. Dessa forma, para problemas elástico lineares, a Equação 2.1 descreve o problema a ser solucionado:

$$\mathbf{K}\bar{u} = \bar{f} \quad (2.1)$$

Sendo \mathbf{K} a matriz de rigidez, \bar{u} o vetor de deslocamentos nodais e \bar{f} o vetor de carregamentos aplicados na estrutura. Porém, enquanto a matriz de rigidez é constante para modelos elástico lineares, ao considerar fenômenos como plasticidade e nucleação de trincas é esperado um comportamento não linear, conforme mostrado na Figura 2.1, sendo necessário a formulação de modelos mais complexos para simular seu comportamento.

Figura 2.1 – Relações de tensão/deformação para diferentes modelos. a) Modelo linear elástico. b) Modelo idealmente plástico. c) Modelo com amolecimento.



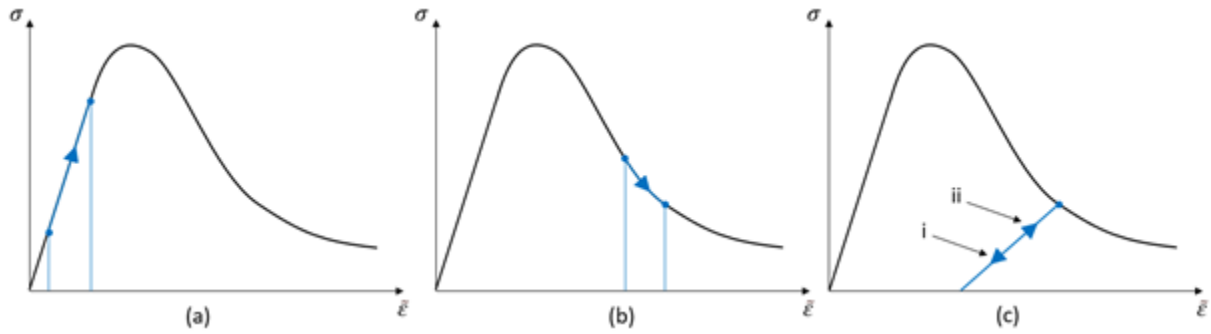
Fonte: O autor.

2.1 MODELOS DE DEGRADAÇÃO ELÁSTICA

Nos modelos de degradação elástica, o processo de nucleação de trincas e formação de vazios no material é trabalhado através da deterioração das propriedades mecânicas. Nesse aspecto, geralmente o material é considerado inicialmente isotrópico e uniforme, e conforme ocorre a deformação são utilizados de relações constitutivas para determinar a variação da elasticidade em cada nó. Além disso, para a preservar a degradação entre as iterações, é necessário determinar funções de carregamento, de descarregamento e recarregamento. A Figura 2.2 demonstra o comportamento dessas funções para um gráfico de tensão (σ) por deformação equivalente ($\tilde{\epsilon}$), cada modelo apresenta uma maneira específica para calcular $\tilde{\epsilon}$. Dessa forma, a função de carregamento é normalmente definida por uma relação entre a solicitação do material e uma variável histórica, ambas definidas pelo modelo de análise escolhido.

Os modelos de degradação elástica, em geral, se diferem dos de plasticidade por não se considerar a formação de deformações residuais no processo de descarregamento. Assim, a função de

Figura 2.2 – Curvas de carregamento. a) Carregamento elástico. b) Carregamento inelástico. c) i. Descarregamento; ii. Recarregamento.



Fonte: O autor.

descarregamento é atribuída, de forma geral, à rigidez secante. Mesmo assim, a inexistência de deformação residual não está atrelada a recuperação da trinca, tal fenômeno não é considerado para os modelos mais comuns.

2.1.1 Modelos de Dano

Para os modelos de dano, em geral define-se uma variável que determina o grau de deterioração do material, podendo ser determinada pela densidade de vazios na seção transversal ou pela redução do módulo de elasticidade, Equação 2.2. Dessa forma, a variável Dano (D) pode variar de 0, correspondendo a um material intacto, a 1, que seria um vazio ou uma trinca.

$$D = 1 - \frac{E_D}{E_0} \quad (2.2)$$

Sendo D a variável de dano, E_D o módulo de elasticidade do material danificado e E_0 o módulo de elasticidade inicial. Na generalização dos modelos de degradação elástica apresentada em Penna (2011) o autor aponta que os modelos constitutivos de dano se diferenciam pela definição individual da função de carregamento. O autor descreve uma formulação unificada para essa de acordo com a Equação 2.3.

$$F = f(\mathcal{A}) - h(k) \quad (2.3)$$

Onde, $f(\mathcal{A})$ seria uma função baseada nas condições do material em relação ao estado das tensões, deformações, variáveis termodinâmicas, entre outros, e $h(k)$ um parâmetro histórico relacionado à mesma função. Para o modelo de Mazars e Lemaitre (1985), por exemplo, é descrita uma função de carregamento conforme a Equação 2.4:

$$F(\tilde{\epsilon}, D) = \tilde{\epsilon} - k(D), \quad \tilde{\epsilon} = \sqrt{\epsilon_{ij}\epsilon_{ij}} \quad (2.4)$$

Sendo $k(D)$ o maior valor de $\tilde{\epsilon}$ atingido durante a simulação. Ademais, Penna (2011) inclui demonstrações para relações constitutivas, formulações teóricas e desenvolvimento de tensores secantes e tangentes para modelos de dano isotrópico de Mazars e Lemaitre (1985), Simo e Ju (1987), Ju (1989), Lemaitre e Chaboche (1990), de Vree et al. (1995) e Mazars (1984), e de dano ortotrópico por de Borst e Gutiérrez (1999).

2.1.2 Modelos de Fissuração Distribuída

Nos modelos de fissuração distribuída, através da formação de fissuras no material, é considerado que ele gradativamente muda seu comportamento de isotrópico para ortotrópico. Dessa forma, as propriedades são definidas a partir de um sistema local de coordenadas alinhado com as tensões principais e normal à região de fissuração. Entretanto, apesar de permitirem a determinação de uma trajetória para a trinca sem a necessidade da modelagem discreta dela, o que elimina o custo computacional de refazer a malha a cada iteração, Jirásek (2011) aponta a possibilidade de ocorrência de *Stress Locking*, caso a malha não esteja devidamente alinhada com a direção de fissuração. Portanto, é necessário ao usuário conhecer previamente a trajetória da trinca esperada para o modelo.

Penna (2011) descreve uma linha do tempo no desenvolvimento desses modelos, iniciando com Rashid (1968) que considerava a direção normal à fissura com rigidez nula, sendo seguido por uma série de modelos que tentavam aproximar mais dos dados experimentais através da determinação de coeficientes de retenção – Suidan e Schnobrich (1973), Băzant e Oh (1983), Rots et al. (1985), entre outros. Outra consideração importante na classificação desses modelos é a direção de fissuração, apesar de algumas formulações trabalharem com direção fixa, em outros é possível encontrar resultados com rotação do sentido da fissuração ao longo da análise. Em seu trabalho, Penna (2011) generaliza os modelos de fissuração como modelos de dano ortotrópico utilizando múltiplas funções de carregamento para representar seu comportamento. Desse modo, os modelos apresentariam funções de carregamento diferentes para cada uma das direções principais de tensão.

2.1.3 Relações Tensão-Deformação

Nas Seções 2.1.1 e 2.1.2 foi explicado como a evolução do dano ocorre em cada modelo de degradação elástica. Apesar disso, é necessário desenvolver relações para possibilitar descrever o comportamento do material sob carregamento. Essas são importantes para entender como será a evolução da degradação ao longo da análise.

De Penna (2011), podem ser citadas variadas relações para o comportamento. Boone et al. (1986) introduz uma relação exponencial para a degradação, que incorpora um comportamento linear-elástico seguida pela Equação 2.5a. Enquanto, Carreira e Chu (1985) propõem uma relação polinomial, Equação 2.5b, que não incorporam um comportamento inicialmente elástico.

$$\sigma = f_t \exp(-k(\epsilon - \epsilon_t)), \quad \text{sendo: } k = \frac{f_t}{G_f} \quad (2.5a)$$

$$\sigma_v = \frac{k \left(\frac{\epsilon}{\epsilon_v} \right)}{k - 1 + \left(\frac{\epsilon}{\epsilon_v} \right)^k}, \quad \text{sendo: } k = \frac{1}{1 - \left(\frac{f_v}{\epsilon_v E_0} \right)} \quad v = t, c \quad (2.5b)$$

Com f_t e f_c sendo a resistência a tração e compressão, G_f a energia de fratura, ϵ_t e ϵ_c a deformação máxima para tração e compressão e E_0 o módulo de elasticidade inicial.

Outras abordagens mais simplificadas utilizam apenas de relações lineares, como a aproximação bilinear, Equação 2.6a para tração e Equação 2.6b para compressão, com E_2 representando a inclinação da linha de redução da elasticidade.

$$\sigma = \frac{\epsilon_{t,cr} - \epsilon}{\epsilon_{t,cr} - \epsilon_t}, \quad \text{sendo: } \epsilon_{t,cr} = \epsilon_t + \frac{2G_f}{f_t} \quad (2.6a)$$

$$\sigma = \frac{\epsilon_{c,cr} - \epsilon}{\epsilon_{c,cr} - \epsilon_c}, \quad \text{sendo: } \epsilon_{c,cr} = \epsilon_c + \frac{f_c}{E_2} \quad (2.6b)$$

As relações entre a degradação e a deformação são abordadas com diferentes expressões na literatura para cada uma das relações citadas. Se essas forem de interesse do leitor, um compilado pode ser encontrado no trabalho de Penna (2011).

2.2 MODELOS DE CAMPO DE FASES

2.2.1 A Teoria de Griffith

O critério de Griffith, Equação 2.7a, é uma formulação matemática proposta para tentar prever as condições que levam à propagação de trincas. Dessa forma, o critério propõe uma relação entre a energia crítica, a energia de deformação (Ψ_s), Equação 2.7b, e a energia relacionada com a formação de superfície de trinca (Ψ_c), Equação 2.7c. Assim, seguindo as definições da Equação 2.7a, pode-se reformular o variacional de energia de acordo com a Equação 2.7d. Nessa formulação, uma trinca só teria condições para se propagar caso a energia gerada na formação de superfície fosse maior ou igual a consumida para a deformação do material.

$$E_t(\bar{u}, \Gamma) = \int_{\Omega} \psi_0(\underline{\epsilon}(\bar{u})) dV + \int_{\Gamma} G_f dA - P_{ext} \quad (2.7a)$$

$$\Psi_s = \int_{\Omega} \psi_0(\underline{\epsilon}(\bar{u})) dV \quad (2.7b)$$

$$\Psi_c = \int_{\Gamma} G_f dA \quad (2.7c)$$

$$E_t = \Psi_s + \Psi_c - P_{ext} \quad (2.7d)$$

Dessa forma, para composição do modelo variacional, são impostas as condições de Karush-Kuhn-Tucker, que atendem diretamente às condições físicas do problema. São essas:

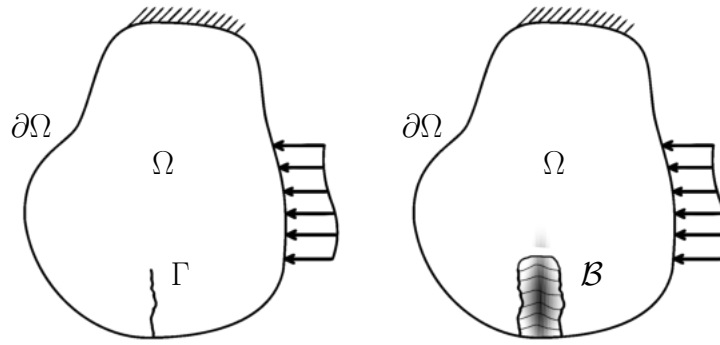
- $\dot{\Gamma} \geq 0$, a trinca não se recupera.
- $\mathcal{G} - G_f \leq 0$, a energia de fratura não excede a crítica G_f
- $(\mathcal{G} - G_f)\dot{\Gamma} = 0$, a trinca cresce ao atingir a condição de $\mathcal{G} = G_f$

2.2.2 Modelagem de trincas difusas

Apesar do critério de Griffith conseguir alcançar bons resultados analíticos, a formulação tradicional não atende às propriedades dos MEF de malha fixa. Inserir uma trinca aguda no modelo poderia gerar descontinuidades numéricas nas fronteiras dos elementos e causaria instabilidade da solução. Para resolver isso, os modelos de Campo de Fase introduzem o conceito de trinca difusa. Dessa forma, como mostrado a seguir, na Equação 2.8, o termo Ψ_c é reformulado, inserindo o valor de γ , denominado como densidade da superfície de trinca. O domínio da trinca deixa de ser uma região discreta Γ e passa a ser um domínio difuso \mathcal{B} . Como mostrado na Figura 2.3.

$$\Psi_c = \int_{\Gamma} G_f dA \approx \int_{\mathcal{B}} G_f \gamma(\phi, \nabla\phi) dV \quad (2.8)$$

Figura 2.3 – Trinca discreta e trinca difusa com seus respectivos domínios



Fonte: O Autor

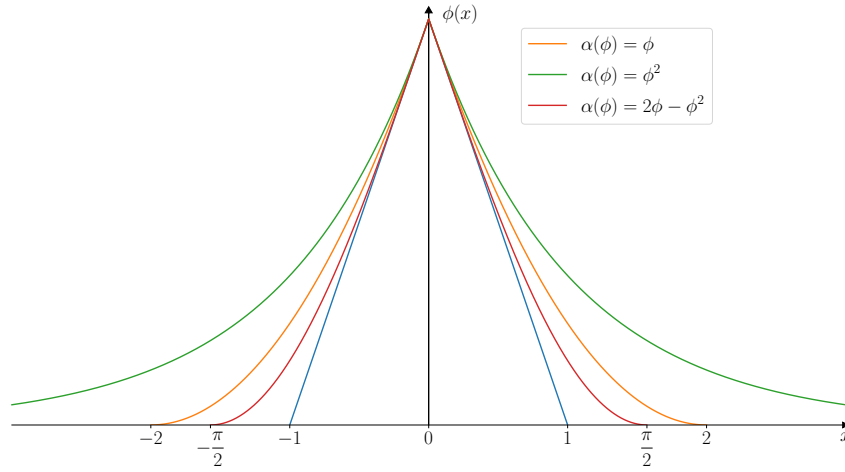
2.2.2.1 Função de geometria da trinca

Para a expressão de suavização da trinca, $\gamma(\phi, \nabla\phi)$, Wu (2017) propôs uma generalização do termo, conforme a Equação 2.9:

$$\gamma(\phi, \nabla\phi) = \frac{1}{C_0} \left[\frac{1}{l_0} \alpha(\phi) + l_0 |\nabla\phi|^2 \right] \quad (2.9)$$

Nesse formato temos l_0 como o comprimento característico da trinca, $\alpha(\phi)$ a função de geometria da trinca e a constante $C_0 = 4 \int_0^1 \alpha^{\frac{1}{2}}(\phi) d\phi$. Para definição da função de geometria da trinca, é necessário que $\alpha(\phi) \in [0, 1]$ atenda às seguintes propriedades: $\alpha(0) = 0$, $\alpha(1) = 1$ e que ela seja monotonicamente crescente.

Em modelos baseados na formação de trincas em materiais frágeis as funções mais adotadas na literatura são as de Bourdin et al. (2000), $\alpha(\phi) = \phi^2$, e Pham et al. (2011), $\alpha(\phi) = \phi$. Ademais, no trabalho de Wu (2017) o autor sugere o uso da função $\alpha(\phi) = 2\phi - \phi^2$ e introduz

Figura 2.4 – Região de degradação em domínio unidimensional com $l_0 = 1$ 

Fonte: Wu et al. (2020) adaptado.

uma formulação geral, Equação 2.10, que engloba as três citadas em função do parâmetro ξ .

$$\alpha(\phi) = \xi\phi + (1 - \xi)\phi^2 \quad \xi \in [0, 2] \quad (2.10)$$

Cada valor de ξ produz uma largura diferente para a região de degradação. Na Tabela 1 pode-se encontrar o valor final da largura da fase e o formato que essa fase forma em um domínio unidimensional. Essas curvas são destacadas na Figura 2.4 que as projetam em um gráfico para $l_0 = 1$. Ademais, o intervalo $\xi \in [0, 2]$ serve como uma garantia que a função de geometria da

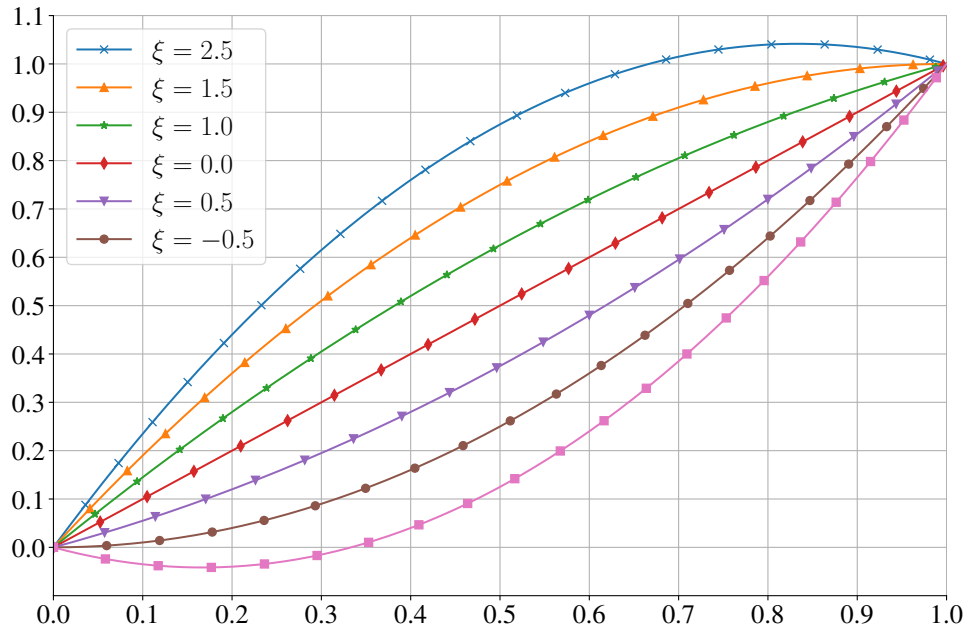
Tabela 1 – Funções de geometria da trinca e suas propriedades

| $\alpha(\phi)$ | C_0 | $\phi(x)$ | Largura da fase |
|------------------|---------------|---------------------------------------|-----------------|
| ϕ | $\frac{8}{3}$ | $\left(1 - \frac{ x }{2l_0}\right)^2$ | $4l_0$ |
| ϕ^2 | 2 | $\exp\left(-\frac{ x }{l_0}\right)$ | ∞ |
| $2\phi - \phi^2$ | π | $1 - \sin \frac{ x }{l_0}$ | πl_0 |

Fonte: Wu et al. (2020) adaptado.

trinca de Wu (2017) não apresentará recuperação, uma das condições de Karush-Kuhn-Tucker. Valores de $\xi < 0$ levariam a uma redução inicial de $\alpha(\phi)$, enquanto valores de $\xi > 2$ gerariam redução do valor da função para ϕ próximo de 1. Isso é representado na Figura 2.5 que mostra a evolução do valor da função ao longo de ϕ para diferentes valores de ξ .

A Função de Geometria da Trinca de Wu (2017) foi implementada no sistema INSANE como parte desse trabalho. Detalhes dessa implementação serão abordados na Seção 3.2.1.

Figura 2.5 – Valores da função $\alpha(\phi)$ para diferentes ξ 

Fonte: Wu et al. (2020) adaptado.

2.2.3 Degradação da energia no campo

Com a determinação de uma região de trinca difusa, torna-se necessário a reformulação da densidade de energia de deformação, conforme mostrado na Equação 2.11a. Sendo assim, o termo $\psi(\underline{\epsilon}(\bar{u}), \phi)$ passa a ser formado pela multiplicação da função $g(\phi)$, também conhecida como Função de Degradação Energética, com o valor de ψ_0 , Equação 2.11b. Essa função possui grande influência no comportamento do estrutural do material, pois determinará como o funcional de energia vai variar com a formação das trincas. Nesse sentido, como otimização desse fator, a separação da variável ψ_0 é adotada por alguns modelos constitutivos para evitar a formação de trincas em regiões sob compressão. A separação da variável é feita a partir de definições de componentes ativas (ψ_0^+) e inativas (ψ_0^-), Equação 2.11c, uma outra abordagem eficiente para esse propósito é a formulação híbrida que será tratada na Seção 2.2.6.3.

$$\Psi_s = \int_{\Omega} \psi_0(\underline{\epsilon}) dV \approx \int_{\Omega} \psi(\underline{\epsilon}, \phi) dV \quad (2.11a)$$

$$\psi(\underline{\epsilon}, \phi) = g(\phi) \psi_0(\underline{\epsilon}) \quad (2.11b)$$

$$\psi(\underline{\epsilon}, \phi) = g(\phi) \psi_0^+(\underline{\epsilon}) + \psi_0^-(\underline{\epsilon}) \quad (2.11c)$$

2.2.3.1 Funções de degradação energética

Para as funções de degradação energética, de acordo com Wu et al. (2020), são necessárias que as seguintes condições sejam atendidas:

- $g(0) = 1$, material intacto, logo, não há degradação.

- $g(1) = 0$, material completamente degradado, logo, não há energia residual.
- $\frac{dg}{d\phi} = g'(\phi) < 0$, a função é monotonicamente decrescente.
- $\left. \frac{dg}{d\phi} \right|_{\phi=1} = g'(1) = 0$, sem variação súbita ao fim da degradação.

Leão (2021) informa que para funções com $g'(0) = 0$ ocorre um comportamento inicialmente linear-elástico. Entretanto, o mesmo não ocorre para funções com $g'(0) < 0$, que a degradação inicia no mesmo instante que o carregamento. A Tabela 2 apresenta algumas funções de degradação utilizadas na literatura. Das funções apresentadas na Tabela 2 ape-

Tabela 2 – Funções de degradação energética

| $g(\phi)$ | $Q(\phi)$ | autores |
|---|--|-----------------------|
| $(1 - \phi)^2$ | — | Bourdin et al. (2000) |
| $3(1 - \phi)^2 - 2(1 - \phi)^3$ | — | Karma et al. (2001) |
| $4(1 - \phi)^3 - 3(1 - \phi)^4$ | — | Kuhn et al. (2015) |
| $\frac{(1 - \phi)^p}{(1 - \phi)^p + Q(\phi)}$ | $a_1\phi + a_1a_2\phi^2 + a_1a_2a_3\phi^3$ | Wu (2017) |

Fonte: Wu et al. (2020) adaptado.

nas a função de Wu (2017) não foi implementada no INSANE até o momento. Como será abordado a seguir, essa função apresenta características promissoras, pois permite realizar uma correlação entre seus parâmetros e propriedades do material, portanto, como parte deste trabalho, a implementação dela será abordada na Seção 3.2.1.

2.2.3.2 Degradação da energia baseada em propriedades do material

A Função de Degradação Energética de Wu (2017) permite definir diretamente parâmetros a_1 , a_2 , a_3 e p que interferem nos valores da função $g(\phi)$. A partir disso, é possível demonstrar uma relação entre as constantes citadas e propriedades físicas do material.

De Wu (2017) são apresentadas as Equações 2.12a, 2.12b e 2.12c, derivadas da análise dessa equação em um domínio unidimensional.

$$a_1 = \frac{2E_0G_f}{f_t^2} \cdot \frac{\xi}{C_0l_0} \quad (2.12a)$$

$$a_2 = \frac{1}{\xi} \left[\left(-\frac{4\pi\xi^2G_f}{C_0f_t}k_0 \right)^{\frac{2}{3}} + 1 \right] - (p + 1) \quad (2.12b)$$

$$a_3 = \begin{cases} 0 & p > 2 \\ \frac{1}{a_2} \left[\frac{1}{\xi} \left(\frac{C_0w_cf_t}{2\pi G_f} \right)^2 - (1 + a_2) \right] & p = 2 \end{cases} \quad (2.12c)$$

Sendo ξ o valor usado na Função de Geometria da Trinca, Equação 2.10, C_0 a constante correspondente à mesma função, conforme a Tabela 1, k_0 e w_c são, respectivamente, a inclinação inicial da curva de degradação e a abertura máxima alcançada com tensão residual, determinadas de acordo com a lei de amolecimento (mais detalhes na Seção 2.2.3.3), G_f a energia crítica de fratura do material e E_0 o módulo de elasticidade inicial.

Ademais, Wu (2017) ressalta que para garantir a condição $\dot{\phi} \geq 0$ (que corresponde a primeira condição de Karush-Kuhn-Tucker) o uso de $\xi = 2$ é recomendado, pois isso permite o uso de inclinações iniciais mais negativas, logo, um uso de leis de amolecimento mais variadas. Além disso, novamente para impedir a regeneração da região degradada, deve-se aplicar $p \geq 2$, sendo que para $p > 2$ temos $a_3 = 0$ e w_c tendendo ao infinito.

2.2.3.3 Leis de amolecimento

Como tratado anteriormente, as leis de amolecimento visam descrever como será o comportamento da curva de tensão-deformação no processo de fratura de materiais frágeis, mais especificamente em ensaios de tração uniaxial. Para a Função de Degradação Energética de Wu (2017), o autor foi capaz de determinar valores ótimos de a_2 , a_3 e p para diferentes curvas trabalhadas na literatura.

Conforme explicado anteriormente, essas leis determinam condições de inclinação inicial, abertura máxima da trinca e formato da curva. Para as leis de amolecimento linear e de Cornelissen, que possuem abertura de trinca máxima definida, é possível determinar os parâmetros analiticamente, derivando a expressão de $\sigma(w)$ para encontrar k_0 em $\sigma'(0) = k_0$ e integrando de 0 a w_c para G_f . Apesar disso, para as leis exponencial e hiperbólica, que não possuem abertura máxima de trinca definida, valores ótimos, encontrados por ajuste de dados, são apresentados em Wu (2017) e serão utilizados conforme o trabalho do referido autor. Assim, pode-se definir a_2 , a_3 e p :

- Lei de amolecimento linear:

- forma da curva dada pela Equação 2.13:

$$k_0 = -\frac{f_t^2}{2G_f}, \quad w_c = \frac{2G_f}{f_t} \implies p = 2 \quad (2.13)$$

- Valores da função dados pela Equação 2.14:

$$a_2 = \frac{1}{\xi} \left[\left(\frac{2\pi\xi^2}{C_0} \right)^{\frac{2}{3}} + 1 \right] - 3, \quad a_3 = 0 \quad (2.14)$$

- Lei de amolecimento de Cornelissen, Equação 2.15:

$$\sigma(w) = f_t \left[(1 + \eta_1^3 r^3) \exp(-\eta_2 r) - r (1 + \eta_1^3) \exp(-\eta_2) \right] \quad r = \frac{w}{w_c} \quad (2.15)$$

- forma da curva dada pela Equação 2.16:

$$k_0 = n_k \frac{f_t}{w_c}, \quad w_c = \frac{1}{n_w} \frac{G_f}{f_t} \implies p = 2 \quad (2.16)$$

Com n_k dado pela Equação 2.17:

$$n_k = -\eta_2 - (1 + \eta_1^3) + e^{-\eta_2} \quad (2.17)$$

e n_w dado pela Equação 2.18:

$$n_w = - \left[(1 + \eta_1^3) \left(0.5 + \frac{1}{\eta_2} \right) + \frac{3\eta_1^3}{\eta_2^2} \left(1 + \frac{2}{\eta_2} + \frac{2}{\eta_2^2} \right) \right] e^{-\eta_2} + \frac{1}{\eta_2} + 6 \frac{\eta_1^3}{\eta_2^4} \quad (2.18)$$

- Valores da função dados pela Equação 2.19:

$$a_2 = \frac{1}{\xi} \left[\left(-\frac{4\pi\xi^2 n_k}{C_0} \right)^{\frac{2}{3}} + 1 \right] - 3, \quad a_3 = \frac{1}{a_2} \left[\frac{1}{\xi} \left(\frac{C_0}{2\pi n_w} \right)^2 - (1 + a_2) \right] \quad (2.19)$$

- Lei de amolecimento exponencial, Equação 2.20:

$$\sigma(w) = f_t \exp \left(-\frac{f_t}{G_f} w \right) \quad (2.20)$$

- forma da curva dada pela Equação 2.21:

$$k_0 = -\frac{f_t^2}{G_f}, \quad w_c = +\infty \implies a_3 = 0 \quad (2.21)$$

- Valores da função (p obtido por ajuste de dados em Wu (2017)), Equação 2.22:

$$p = 2.5, \quad a_2 = \frac{1}{\xi} \left[\left(\frac{4\pi\xi^2}{C_0} \right)^{\frac{2}{3}} + 1 \right] - (p + 1) \quad (2.22)$$

- Lei de amolecimento hiperbólica, Equação 2.23:

$$\sigma(w) = f_t \left(1 + \frac{f_t}{G_f} w \right)^{-2} \quad (2.23)$$

- forma da curva dada pela Equação 2.24:

$$k_0 = -\frac{2f_t^2}{G_f}, \quad w_c = +\infty \implies a_3 = 0 \quad (2.24)$$

- Valores da função (p obtido por ajuste de dados em Wu (2017)), Equação 2.25:

$$p = 4, \quad a_2 = \frac{1}{\xi} \left[\left(\frac{8\pi\xi^2}{C_0} \right)^{\frac{2}{3}} + 1 \right] - (p + 1) \quad (2.25)$$

2.2.4 Montagem do modelo em Elementos Finitos

2.2.4.1 Equações na Forma Fraca

De Leão (2021) pode-se reproduzir as equações do modelo de Campo de Fases para a forma integral do problema, também conhecida como forma fraca. Assim, as Equações 2.26a e 2.26b apresentam o sistema variacional a ser solucionado.

$$\left\{ \begin{array}{l} \int_{\Omega} \underline{\sigma} : \delta \underline{\epsilon} dV = \delta P_{ext} \\ \int_{\mathcal{B}} g'(\phi) \bar{Y} \delta \phi + G_f \delta \gamma dV \geq 0 \end{array} \right. \quad (2.26a)$$

$$\left\{ \int_{\mathcal{B}} g'(\phi) \bar{Y} \delta \phi + G_f \delta \gamma dV \geq 0 \right. \quad (2.26b)$$

Sendo $\delta \gamma$ definido pela Equação 2.27:

$$\delta \gamma = \frac{1}{G_0} \left[\frac{1}{l_0} \alpha'(\phi) \delta \phi + 2l_0 \nabla \phi \cdot \nabla \delta \phi \right] \quad (2.27)$$

E \bar{Y} a variável conhecida como força motriz efetiva da trinca, calculada de acordo com a Equação 2.28a e que se relaciona com a força motriz da trinca (Y), Equação 2.28b, pela relação dada na Equação 2.28c.

$$\bar{Y} = \frac{\partial \psi}{\partial g} \quad (2.28a)$$

$$Y = -\frac{\partial \psi}{\partial \phi} = \frac{\partial \psi}{\partial g} \frac{\partial g}{\partial \phi} \quad (2.28b)$$

$$Y = -g'(\phi) \bar{Y} \quad (2.28c)$$

2.2.4.2 Discretização em MEF 2D

Para discretização em MEF para modelos 2D é necessário definir a função de forma, que relaciona as variáveis de contínuas do modelo com as condições nodais da malha. Definem-se assim, uma função N_I^u , também conhecida como função de forma, para os deslocamentos e, seguindo um mesmo procedimento, N_I^ϕ , para a variável ϕ . Essas funções devem ser definidas para cada nó do elemento formando uma matriz de interpolação, \mathbf{N}_I^u e \mathbf{N}_I^ϕ , conforme mostrado nas Equações 2.29a e 2.29b, respectivamente.

$$\begin{pmatrix} u(x, y) \\ v(x, y) \end{pmatrix} = \begin{bmatrix} N_I^u & 0 \\ 0 & N_I^u \end{bmatrix} \begin{pmatrix} u_I \\ v_I \end{pmatrix} = \mathbf{N}_I^u \bar{d}_I \quad I = [1, \text{nós do elemento}] \quad (2.29a)$$

$$\begin{pmatrix} \phi(x, y) \end{pmatrix} = \begin{bmatrix} N_I^\phi \end{bmatrix} \begin{pmatrix} a_I \end{pmatrix} = \mathbf{N}_I^\phi \bar{a}_I \quad I = [1, \text{nós do elemento}] \quad (2.29b)$$

Sendo u e v os deslocamentos em x e y respectivamente, u_I e v_I os deslocamento nodais, \bar{d}_I o vetor de deslocamentos nodais para cada nó, a_I a variável nodal de campo de fase e \bar{a} o vetor das variáveis nodais de campo de fase do elemento. De forma similar define-se as matrizes de interpolação de deformação e de gradiente de campo, \mathbf{B}_I^u e \mathbf{B}_I^ϕ , respectivamente. As Equações

2.30a e 2.30b descrevem essas matrizes.

$$\bar{\epsilon} = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{x,y} \end{pmatrix} = \begin{bmatrix} N_{i,x}^u & 0 \\ 0 & N_{I,y}^u \\ N_{i,y}^u & N_{i,x}^u \end{bmatrix} \begin{pmatrix} u_I \\ v_I \end{pmatrix} = \mathbf{B}_I^u \bar{d}_I \quad I = [1, \text{nós do elemento}] \quad (2.30a)$$

$$\nabla \bar{\phi} = \begin{pmatrix} \phi_{,x} \\ \phi_{,y} \end{pmatrix} = \begin{bmatrix} N_{I(i),x}^\phi \\ N_{I(i),y}^\phi \end{bmatrix} (a_I) = \mathbf{B}_I^\phi \bar{a}_I \quad I = [1, \text{nós do elemento}] \quad (2.30b)$$

Sendo a notação “ $,x$ ” e “ $,y$ ” usadas para representar as derivadas parciais com relação a x e y respectivamente. Assim, por exemplo, $N_{i,x}^u = \frac{\partial N_I^u}{\partial x}$.

Seguindo as definições anteriores, é possível reescrever as Equações 2.26a e 2.26b em função das variáveis nodais:

Definindo os vetores $\bar{\sigma}$ e $\bar{\epsilon}$ pelas Equações 2.31a e 2.31b transforma-se o produto escalar tensorial em um produto escalar vetorial conforme a Equação 2.31c.

$$\bar{\sigma} = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} \quad (2.31a)$$

$$\bar{\epsilon} = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ 2\epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{pmatrix} \quad (2.31b)$$

$$\underline{\sigma} : \delta \underline{\epsilon} = \sigma_{kl} \cdot \delta \epsilon_{kl} = \delta \bar{\epsilon}^T \cdot \bar{\sigma} \quad (2.31c)$$

Substituindo o produto escalar vetorial na equação dos deslocamento na forma integral (veja Equação 2.26a) e considerando a decomposição da potência externa, $\delta P_{ext} = \delta \bar{d}^T \bar{f}_{ext(i)}$, sendo \bar{d} o vetor de deslocamentos nodais, temos que:

$$\int_{\Omega} \delta \bar{\epsilon}^T \cdot \bar{\sigma} dV = \delta \bar{d}^T \bar{f}_{ext(i)} \quad (2.32)$$

Fazendo a substituição de $\bar{\epsilon}$ como mostrado na Equação 2.30a e usando da propriedade da transposta do produto das matrizes, $(\mathbf{A}\bar{b})^T = \bar{b}^T \mathbf{A}^T$, temos que:

$$\delta \bar{d}^T \int_{\Omega} \mathbf{B}_I^{uT} \bar{\sigma} dV = \delta \bar{d}^T \bar{f}_{ext(i)} \quad (2.33)$$

E, para $\partial \bar{d}$ arbitrário e diferente de 0:

$$\int_{\Omega} \mathbf{B}_I^{uT} \bar{\sigma} dV = \bar{f}_{ext(i)} \quad (2.34)$$

De forma similar, substituindo as Equações 2.29b e 2.30b em 2.26b, tem-se que:

$$\int_B g' \bar{Y} \delta \bar{a}^T \mathbf{N}_I^{\phi T} + \delta \bar{a}^T \frac{G_f}{C_0} \left(\frac{1}{l_0} \alpha' \mathbf{N}_I^{\phi T} + 2l_0 \mathbf{B}_I^{\phi T} \nabla \phi \right) dV \geq \bar{0} \quad (2.35)$$

Para $\partial \bar{a}$ arbitrário e diferente de 0:

$$\int_B g' \bar{Y} \mathbf{N}_I^{\phi T} + \frac{G_f}{C_0} \left(\frac{1}{l_0} \alpha' \mathbf{N}_I^{\phi T} + 2l_0 \mathbf{B}_I^{\phi T} \nabla \phi \right) dV \geq \bar{0} \quad (2.36a)$$

$$\int_B \mathbf{N}_I^{\phi T} \left(g' \bar{Y} + \frac{G_f}{C_0 l_0} \alpha' \right) + \frac{2l_0 G_f}{C_0} \mathbf{B}_I^{\phi T} \nabla \phi dV \geq \bar{0} \quad (2.36b)$$

Na realização do processo iterativo do problema de trajetória de equilíbrio é necessário definir o residual das equações e a matriz tangente. Com esses valores é possível, através de métodos como o de Newton-Raphson, determinar a evolução da variável conforme o modelo de controle. Para o residual, determina-se pelo rearranjo das Equações 2.34 e 2.36, as Equações 2.37a e 2.37b respectivamente.

$$\bar{r}_I^u = \int_{\Omega} \mathbf{B}_I^{uT} \bar{\sigma} dV - \bar{f}_{ext} = \bar{0} \quad (2.37a)$$

$$\bar{r}_I^{\phi} = - \int_B \mathbf{N}_I^{\phi T} \left(g' \bar{Y} + \frac{G_f}{C_0 l_0} \alpha' \right) + \frac{2l_0 G_f}{C_0} \mathbf{B}_I^{\phi T} \nabla \phi dV \leq \bar{0} \quad (2.37b)$$

A matriz tangente dos deslocamentos é encontrada derivando a Equação 2.34 por \bar{d} , criando assim, a matriz \mathbf{K}_{IJ}^{uu} para cada elemento, como mostrado em 2.38a, 2.38b e 2.38c:

$$\mathbf{K}_{IJ}^{uu} = \int_{\Omega} \mathbf{B}_I^{uT} \frac{\partial \bar{\sigma}}{\partial \bar{d}_J} dV \quad (2.38a)$$

$$= \int_{\Omega} \mathbf{B}_I^{uT} \frac{\partial \bar{\sigma}}{\partial \bar{\epsilon}} \frac{\partial \bar{\epsilon}}{\partial \bar{d}_J} dV \quad (2.38b)$$

$$= \int_{\Omega} \mathbf{B}_I^{uT} \frac{\partial \bar{\sigma}}{\partial \bar{\epsilon}} \mathbf{B}_J^u dV \quad (2.38c)$$

De forma similar, a matriz tangente da variável de campo de fase, pode ser encontrado derivando a Equação 2.36b, como mostrado em 2.39a e 2.39b:

$$\mathbf{K}_{IJ}^{\phi\phi} = \int_B \mathbf{N}_I^{\phi T} \left(g'' \bar{Y} + \frac{G_f}{C_0 l_0} \alpha'' \right) \delta \phi_J + \frac{2l_0 G_f}{C_0} \mathbf{B}_I^{\phi T} \delta \nabla \phi_J dV \quad (2.39a)$$

$$= \int_B \mathbf{N}_I^{\phi T} \left(g'' \bar{Y} + \frac{G_f}{C_0 l_0} \alpha'' \right) \mathbf{N}_J^{\phi} + \frac{2l_0 G_f}{C_0} \mathbf{B}_I^{\phi T} \mathbf{B}_J^{\phi} dV \quad (2.39b)$$

2.2.5 Métodos de Solução

Para problemas quase-estáticos encontra-se na literatura dois modelos globais de solução: o método monolítico e o método alternado. Enquanto o primeiro faz uma montagem de uma matriz tangente geral, para convergir simultaneamente as variáveis de deslocamento e campo de fase, o segundo realiza a convergência alternada das variáveis, mantendo uma constante enquanto realiza iterações na outra. Ressalta-se que métodos de solução devem aplicar um modelo de convergência consistente e garantir que a variável de campo de fase respeite os contornos $\phi \in [0, 1]$ e a condição de irreversibilidade da trinca, $\dot{\phi} \geq 0$.

Uma maneira de atender a essas condições, foi a estratégia formulada por Miehe et al. (2010) com a introdução de uma variável histórica \mathcal{H} no lugar de \bar{Y} , sendo definida pela Equação 2.40. Isso representa que os nós estão sempre sob o valor máximo de tensão ativa atingida ao longo dos passos da análise. Miehe et al. (2010) mostra que a utilização dessa abordagem junto a Função de Geometria da Trinca de Bourdin et al. (2000), $\alpha(\phi) = \phi^2$, garante o atendimento de ambas condições.

$$\mathcal{H}^i = \max(\bar{Y}, \mathcal{H}^{i-1}) \quad (2.40)$$

2.2.5.1 Método monolítico

Para realização do método monolítico faz-se necessário definir as componentes $\mathbf{K}_{IJ}^{u\phi}$ e $\mathbf{K}_{IJ}^{\phi u}$, para montar o sistema conforme a Equação 2.41. Esse, posteriormente, pode ser resolvido por um processo iterativo padrão de Newton-Raphson. Entretanto, Wu et al. (2020) atenta que o modelo monolítico só pode ser utilizado quando as condições de contorno da variável de campo de fase e a condição de irreversibilidade da trinca são atendidos de forma intrínseca, como com a adoção da variável \mathcal{H} , mesmo que nesse caso a matriz pode se tornar não-simétrica.

$$\begin{bmatrix} \mathbf{K}_{ij}^{uu} & \mathbf{K}_{ij}^{u\phi} \\ \mathbf{K}_{ij}^{\phi u} & \mathbf{K}_{ij}^{\phi\phi} \end{bmatrix} \begin{pmatrix} d\bar{d}_j \\ d\bar{a}_j \end{pmatrix} = \begin{pmatrix} \bar{r}_i^u \\ \bar{r}_i^\phi \end{pmatrix} \quad (2.41)$$

Esse método de solução, apesar de amplamente utilizado na literatura, apresenta dificuldades de convergência. Nesse sentido, Leão (2021) cita ter enfrentado dificuldades para convergir em situações de decaimento do valor do fator de carga.

2.2.5.2 Método alternado com variável histórica

Como descrito anteriormente, os métodos alternados convergem as variáveis deslocamento e campo de fase individualmente, enquanto mantém a outra constante. O método com uso de variável histórica foi introduzido por Miehe et al. (2010). Para tal, o autor propõe a reformulação da Equação 2.36b incorporando a variável histórica \mathcal{H} , Equação 2.40, e transformando na Equação 2.42.

$$\int_B \mathbf{N}_i^{\phi T} \left(g' \mathcal{H} + \frac{G_f}{C_0 l_0} \alpha' \right) + \frac{2l_0 G_f}{C_0} \mathbf{B}_i^{\phi T} \nabla \phi dV = \bar{0} \quad (2.42)$$

Como citado anteriormente, a utilização desse método é condicionada ao uso conjunto com a Função de Geometria da Trinca de Bourdin et al. (2000) para garantir os limites da variável de campo. Wu et al. (2020) reitera a vantagem da utilização desse modelo com a Função de Degradação Energética de Bourdin et al. (2000), pois torna o variacional do campo de fase um sub-problema linear.

Essas limitações demonstram a necessidade da formulação de novos métodos que possibilitariam a utilização de funções mais complexas, como a citada na Seção 2.2.3.2 que leva em conta propriedades do material. Portanto, posteriormente surgiu o método com constrição de contorno que será abordado a seguir.

2.2.5.3 Método alternado com constrição de contorno

O método alternado com constrição de contorno foi uma alternativa introduzida por Farrel e Maurini (2017) baseado no trabalho de Benson e Munson (2006) que desenvolve métodos de solução para problemas de complementariedade mista. Nesse sentido, a solução pode ser tratada como um problema de otimização que busca encontrar o vetor de campo de fase que melhor atende certas condições de contorno.

De Wu et al. (2020) a solução é otimizada dado o atendimento das seguintes condições:

$$\begin{cases} a_{I,n} < a_{I,n+1} < 1 & r_I^\phi = 0 \\ a_{I,n+1} = a_{I,n} & r_I^\phi \leq 0 \\ a_{I,n+1} = 1 & r_I^\phi \geq 0 \end{cases} \quad (2.43)$$

Para solução do problema, Benson e Munson (2006) descreve um algoritmo com iterações de Newton-Raphson mas que realiza a separação da solução em conjuntos ativos e inativos. Assim, o conjunto ativo é dado pelos nós que atendem as condições da solução e não serão calculados na iteração consecutiva, enquanto o inativo é dado pelos nós que ainda não se enquadram nelas e serão calculados na iteração.

De acordo com Wu et al. (2020) os conjuntos podem ser separados da seguinte maneira:

- Conjunto ativo composto por nós dado que:

- $a_{I,n+1} = a_{I,n}$ e $r_I^\phi < 0$, ou
- $a_{I,n+1} = 1$ e $r_I^\phi > 0$

- Conjunto inativo composto por nós dado que:

- $a_{I,n+1} = a_{I,n}$ e $r_I^\phi > 0$, ou
- $a_{I,n+1} = 1$ e $r_I^\phi < 0$, ou
- $a_{I,n} < a_{I,n+1} < 1$

Ademais, Benson e Munson (2006) sugere a utilização de um procedimento de procura linear seguinte à iteração para acelerar a convergência em modelos altamente não-lineares. Essa definição é endossada por Farrel e Maurini (2017), apesar de não ser trabalhada em Wu et al. (2020). A procura é feita com uma busca retroativa para minimização do vetor de resíduo transformado por um operador restritivo. Ademais, para a constrição do contorno é definido um operador de projeção $\pi(\bar{a})$ dado pela Equação 2.44. Esse serve para garantir de forma explícita que o intervalo de solução $[0, 1]$ será atendido.

$$\pi(\bar{a})_I = \begin{cases} 1, & \text{se } a_I \geq 1 \\ 0, & \text{se } a_I \leq 0 \\ a_I & \text{se } 0 \leq a_I \leq 1 \end{cases} \quad (2.44)$$

Os detalhes de implementação desse algoritmo serão abordados na Seção 3.2.2.1.

2.2.6 Modelos Constitutivos

Os Modelos Constitutivos são formulações que buscam descrever e determinar a relação entre tensão e deformação dos materiais. Mais especificamente aos modelos de Campo de Fase, eles se voltam a determinação da variável ψ_0 que compõe a relação visada pela definição $\underline{\sigma} = \frac{\partial \psi}{\partial \underline{\epsilon}}$. É possível dividir os modelos constitutivos em três grupos: os que não fazem distinção entre regiões sob diferentes esforços, os modelos que fazem essa separação de forma explícita, com a definição de componentes ativas (ψ_0^+) e inativas (ψ_0^-), como mostrado na Equação 2.11c, e modelos híbridos.

2.2.6.1 Modelos sem separação de esforços

O Modelo Constitutivo Isotrópico, descrito em Bourdin et al. (2000) é uma exemplo dessa categoria. Nessa formulação adota-se a aplicação direta da relação entre a tensão e a derivada da densidade da energia de deformação pela deformação, como demonstrado nas Equações 2.45a e 2.45b, assim alcançando uma formulação de \bar{Y} conforme a Equação 2.45c. O problema de modelos nesse formato é a possibilidade de formação de trincas em região sobre compressão, que ocorrem nesse modelo na mesma proporção que em regiões sobre tração.

$$\underline{\sigma} = \frac{\partial \psi}{\partial \underline{\epsilon}} = g(\phi) \tilde{\underline{\sigma}} \quad (2.45a)$$

$$\tilde{\underline{\sigma}} = \frac{\partial \psi_0}{\partial \underline{\epsilon}} = \mathbf{E}_0 : \underline{\epsilon} \quad (2.45b)$$

$$\bar{Y} = \frac{\partial \psi}{\partial g} = \psi_0(\underline{\epsilon}) \quad (2.45c)$$

2.2.6.2 Modelos com separação de esforços

Para modelos com separação de esforços a definição de \bar{Y} muda e passa a incorporar apenas a parte ativa da energia de deformação, Equação 2.46. Várias estratégias são encontradas na literatura para a definição de componentes ativas e inativas. Lancioni e Royer-Carfagni (1989) fazem a separação pelo uso das componentes deviatórias como ativas e volumétricas como inativas. Já Amor et al. (2009) introduz um modelo com a mesma ideia mas que considera a parte volumétrica ativa se o traço do tensor de deformações for positivo.

$$\bar{Y} = \frac{\partial \psi}{\partial g} = \psi_0^+(\underline{\epsilon}) \quad (2.46)$$

Dentre os modelos com separação de variáveis, nesse trabalho se utilizará o modelo de Miehe et al. (2010) que têm destaque na literatura por prevenir a formação de trincas em regiões sob compressão. Primeiramente, para desenvolver o modelo é necessário definir algumas funções:

- Função rampa: retorna o próprio valor, se esse ter o mesmo sinal da função, ou zero nos outros casos.

$$\langle x \rangle_{\pm} = \frac{|x| \pm x}{2} \quad (2.47)$$

- Função sinal: retorna 1, se o valor for positivo, -1, se for negativo, ou 0, se for zero.

$$\text{sgn}(x) = \begin{cases} -1, & \text{para } x < 0 \\ 0, & \text{para } x = 0 \\ 1, & \text{para } x > 0 \end{cases} \quad (2.48)$$

- Função de Heaviside: retorna 1, se o valor for positivo, 0, se for negativo, ou 0,5, se for zero.

$$H(x) = \frac{1 + \text{sgn}(x)}{2} \quad (2.49)$$

- Funções R_n^\pm : Retorna a função de Heaviside do traço de um tensor vezes ± 1 .

$$R_n^\pm = H(\pm \text{tr}(\underline{M})) \quad (2.50)$$

- Essas definições possibilitam encontrar a seguinte relação:

$$\langle \text{tr}(\underline{M}) \rangle_\pm = R_n^\pm \text{tr}(\underline{M}) \quad (2.51)$$

O modelo de Miehe et al. (2010) parte da decomposição espectral do tensor de deformação e a definição de componentes ativas ($\underline{\epsilon}^+$) e inativas ($\underline{\epsilon}^-$). Esse processo é ilustrado nas Equações 2.52a, 2.52b e 2.52c com $n = \{1, 2, 3\}$.

$$\underline{\epsilon} = \epsilon_n \bar{p}_n \otimes \bar{p}_n = \underline{\epsilon}^+ + \underline{\epsilon}^- \quad (2.52a)$$

$$\underline{\epsilon}^+ = \langle \epsilon_n \rangle_+ \bar{p}_n \otimes \bar{p}_n \quad (2.52b)$$

$$\underline{\epsilon}^- = \langle \epsilon_n \rangle_- \bar{p}_n \otimes \bar{p}_n \quad (2.52c)$$

Sendo ϵ_n os autovalores do tensor de deformações e \bar{p}_n os autovetores. Assim, pode-se separar a densidade de energia de deformação a partir das Equações 2.53a e 2.53b.

$$\psi_0^+ = \frac{1}{2} \lambda_0 R_n^+ (\text{tr}(\underline{\epsilon}))^2 + \mu_0 \underline{\epsilon}^+ : \underline{\epsilon}^+ \quad (2.53a)$$

$$\psi_0^- = \frac{1}{2} \lambda_0 R_n^- (\text{tr}(\underline{\epsilon}))^2 + \mu_0 \underline{\epsilon}^- : \underline{\epsilon}^- \quad (2.53b)$$

O desenvolvimento da relação tensão-deformação e do tensor constitutivo para o modelo de Miehe et al. (2010) está fora do escopo desse trabalho. Essas relações e detalhes sobre o desenvolvimento de outros modelos com separação de esforços podem ser encontrados em detalhes no trabalho de Leão (2021).

2.2.6.3 Modelos híbridos

De acordo com Wu et al. (2020), os modelos híbridos foram desenvolvidos como uma alternativa para manter o comportamento assimétrico com respeito a tração e compressão, mas ao

mesmo tempo evitando a não-linearidade excessiva, causada pela definição de \bar{Y} pela parte ativa da energia de deformação (Equação 2.46), o que torna a solução dos deslocamentos pouco eficiente nos modelos que fazem separação de variáveis. Para esse trabalho será feito o uso do modelo introduzido em Wu (2017).

Nesse modelo determina-se ψ_0 pela Equação 2.54:

$$\psi_0 = \frac{1}{2E_0} \tilde{\sigma}_{eq}^2 \quad (2.54)$$

Onde $\tilde{\sigma}_{eq}$ é a tensão equivalente calculada a partir das Equações 2.55a e 2.55b.

$$\tilde{\sigma} = E_0 : \underline{\epsilon} \quad (2.55a)$$

$$\tilde{\sigma}_{eq} = \frac{1}{1 + \beta_c} \left(\beta_c \langle \tilde{\sigma}_1 \rangle_+ + \sqrt{3\tilde{J}_2} \right) \quad (2.55b)$$

Em que \tilde{J}_2 é o segundo invariante do tensor tensão deviatório calculado a partir de $\tilde{\sigma}$, $\tilde{\sigma}_1$ a maior dentre as tensões principais e $\beta_c = f_c/f_t - 1$, para f_c a resistência à compressão uniaxial e f_t a resistência à tração uniaxial. Detalhes sobre implementação desse modelo no INSANE serão abordados na Seção 3.2.2.2.

2.3 CONCEITOS BÁSICOS DE PROGRAMAÇÃO

O Sistema INSANE possui a maior parte de seu código em JAVA e, portanto, é fundado na Programação Orientada a Objetos (POO). Essa característica é essencial para sua construção e entendimento, e permite uma melhor organização, modularização e um funcionamento mais eficiente do programa. Nos tópicos a seguir serão abordados alguns dos padrões importantes da estrutura do programa.

2.3.1 Programação Orientada a Objetos

A POO é um paradigma de programação em que busca-se o encapsulamento das funções e das variáveis dentro de classes que posteriormente poderão ser instanciadas como objetos. Dessa forma, faz-se uma classificação do que representa cada conceito dentro da aplicação e quais as funções que cada um deve exercer.

Para cada classe podem ser definidos variáveis, denominadas como atributos, que podem, portanto, caracterizar cada objeto durante a execução. Ademais, pode-se definir funções específicas de cada classe, denominadas de métodos. É possível hierarquizar os atributos e funções num modelo chamado de hereditariedade. Portanto, as classes podem herdar atributos e métodos de outras classes superiores.

2.3.2 Hereditariedade

De modo a facilitar a integração entre diversas classes e possibilitar o uso de forma unificada, as classes são definidas em enquadramentos superiores e inferiores. Dessa forma, as classes inferiores recebem ou “herdam” os métodos e atributos das respectivas classes superiores. Assim,

é possível evitar a repetição de código pois uma classe superior pode possuir diversas classes inferiores.

Em algumas ocasiões, após herdar os métodos da classe referência, é necessário sobrescrevê-los. Dessa forma, em JAVA, métodos idênticos numa situação de hereditariedade são marcados com o rótulo “*@Override*” e mudam a rotina daquela função específica.

2.3.3 Interfaces

As interfaces servem como facilitadores para a generalização do código. Apesar de não implementarem nenhuma rotina, as interfaces apontam métodos obrigatórios que cada uma das classes que a implementam devem possuir. Assim, torna-se possível criar categorias de classes e ter certeza que todas terão os mesmos métodos, evitando erros causados pelo uso de nomenclaturas equivocadas para as funções.

Outro ponto importante é que, apesar de cada classe poder ter apenas uma classe superior, cada uma pode possuir uma quantidade indeterminada de interfaces. Assim, por exemplo, uma classe específica de modelo constitutivo pode herdar os métodos da classe geral de modelos constitutivos e implementar interfaces relativas a clonagem de objetos, serialização, entre outras.

2.4 PONTOS RELEVANTES A RESPEITO DO TEMA

É importante destacar que apesar de incorporarem abordagens diferentes para a formulação constitutiva, os modelos de degradação elástica e os de campo de fases buscam descrever fenômenos físicos similares, como a nucleação de trincas e o processo de falha estrutural. Na literatura encontram-se tentativas de unificação das teorias: Bleyer e Alessi (2018) apresentam um modelo utilizando duas variáveis de dano, enquanto em Wu (2017) é desenvolvida uma teoria unificada com o objetivo de melhorar os resultados de modelos de campos de fase para falhas quase-frágeis.

Além disso, Leão (2021) faz em seu trabalho uma tentativa de comparação da resposta estrutural entre o modelo de Campos de Fase de Miehe et al. (2010), utilizando de 3 materiais semelhantes, e o modelo de fissuração distribuída de Carreira e Chu (1985). Nesses experimentos, apesar dos materiais apresentarem comportamentos similares num ensaio de tração, o comportamento estrutural varia consideravelmente. Assim, Leão (2021) aponta a necessidade de implementação de novas funções de para modelos de Campo de Fase que levem em consideração limites de resistência dos materiais, tal qual serão implementadas nesse trabalho.

Portanto, no intuito de enriquecer o estudo entre as teorias, este trabalho busca comparar os resultados dos modelos e identificar o comportamento da solução de cada um a diferenças nos parâmetros de entrada.

3 METODOLOGIA

3.1 IMPORTAÇÃO DE MALHAS E PARÂMETROS DO MODELO

O sistema INSANE é uma aplicação completa para análise em MEF. O programa incorpora módulos em CAD, geração de malhas, pré-processamento, processamento e pós-processamento, sendo possível realizar simulações inteiramente por sua interface gráfica. Seu módulo de processamento inclui variados modelos de análise para simulações em 2D, dezenas de modelos constitutivos, suporte para XFEM, GFEM e vários tipos de controle de ciclos e iterações. Entretanto, o núcleo de CAD e geração de malhas não é amplamente desenvolvido e sua interface gráfica é pouco eficiente quando trabalhando com malhas muito refinadas.

3.1.1 Processo de implementação

3.1.1.1 *O programa GMSH*

De Geuzaine e Remacle (2009), o GMSH é uma aplicação, também de código livre, já bem estabelecida para os contextos de CAD, geração de malhas e pós-processamento. Seu núcleo numérico consegue gerar malhas estruturadas e desestruturadas com dezenas de milhares de nós em poucos segundos. Além disso, sua interface é de fácil interação e possui suporte para APIs em Python, Julia, C++, etc. Por esses motivos foi optado pela formulação da aplicação baseado na importação de malhas desse programa.

3.1.1.2 *O formato “.msh”*

As malhas geradas pelo GMSH são armazenadas com a extensão “.msh”, o formato do arquivo, descrito por completo na documentação online da aplicação, Gmsh 4.8.4 (2021), pode ser visualizado como uma série de seções que incorporam as informações separadamente. O arquivo tem seções correspondentes a entidades físicas descritas no processamento, esquemas de interpolação, valores para pós-processamento, entre outros, a Figura 3.1 demonstra um exemplo resumido e comentado do arquivo. Como o interesse nesse trabalho está na importação de malhas a implementação desenvolvida se restringe à leitura dos nós, elementos e formas dos elementos descritos no arquivo.

A leitura implementada inicia o armazenamento de dados com a marcação “\$Nodes”, a linha em sequência informa a quantidade total de nós, de blocos e o valor mínimo e máximo dos rótulos nodais. Como será realizada a leitura de todos, essa linha é ignorada. Em seguida inicia-se o padrão de escrita para os nós. Primeiramente, encontra-se uma linha com quatro inteiros, informando em sequência: a dimensão do elemento da geometria original em que os nós do bloco estão contidos (maiores detalhes na Seção 3.1.3.4), em seguida o rótulo do bloco (corresponde ao mesmo da entidade geométrica original), depois, se é um elemento paramétrico

Figura 3.1 – O formato “.msh”

```

$MeshFormat
4.1 0 8      MSH4.1, ASCII
$EndMeshFormat
$Nodes
1 6 1 6      1 entity bloc, 6 nodes total, min/max node tags: 1 and 6
2 1 0 6      2D entity (surface) 1, no parametric coordinates, 6 nodes
1           node tag #1
2           node tag #2
3           etc.
4
5
6
0. 0. 0.     node #1 coordinates (0., 0., 0.)
1. 0. 0.     node #2 coordinates (1., 0., 0.)
1. 1. 0.     etc.
0. 1. 0.
2. 0. 0.
2. 1. 0.
$EndNodes
$Elements
1 2 1 2      1 entity bloc, 2 elements total, min/max element tags: 1 and 2
2 1 3 2      2D entity (surface) 1, element type 3 (4-node quad), 2 elements
1 1 2 3 4    quad tag #1, nodes 1 2 3 4
2 2 5 6 3    quad tag #2, nodes 2 5 6 3
$EndElements

```

Fonte: Documentação Gmsh 4.8.4 (2021) adaptado.

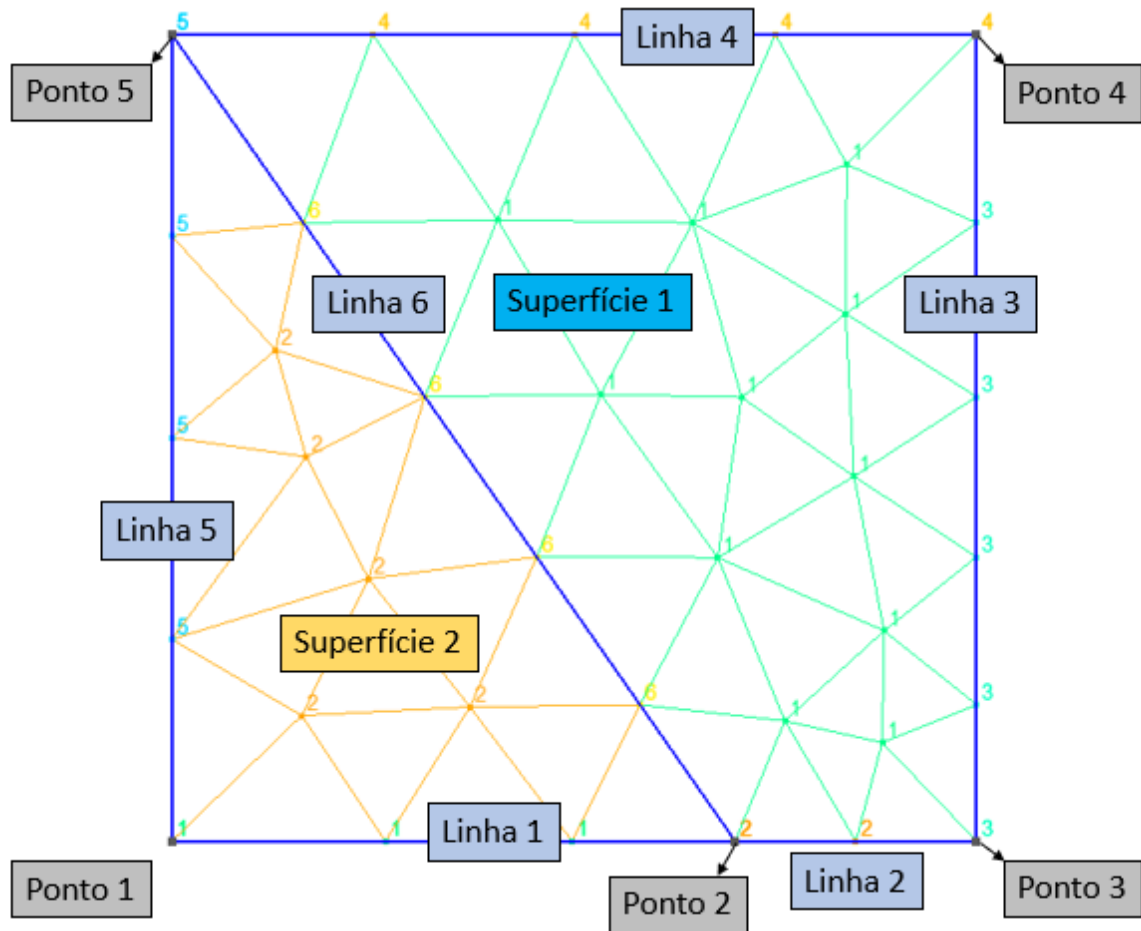
(0, falso e 1, verdadeiro) e, por fim, o número de nós contidos no bloco. Nas linhas seguintes são listados o rótulos dos nós do bloco e, finalmente, as coordenadas de cada nó, respectivamente. Esse padrão se repete até o marcador “\$EndNodes”. A Figura 3.2 mostra a enumeração dos rótulos correspondentes aos blocos de cada nó.

Em seguida, marcado o início pelo rótulo “\$Elements” ocorre a listagem das arestas, faces e volumes. Novamente o processo se inicia com uma linha informando a composição da seção, seguida por um novo padrão de registro. Para os elementos são informados 4 inteiros, os dois primeiros com a mesma lógica que o padrão para nós, o terceiro com um valor correspondente ao tipo de elemento do bloco (1 para elementos unidimensionais, 2 para elementos triangulares e 3 quadrilaterais, entre outros) e o quarto é número de elementos contidos no bloco. Em seguida são listados os elementos do bloco, informando primeiro o rótulo do elemento, seguido pela sequência de nós que compõe sua forma, sendo os nós referenciados pelos respectivos rótulos. A quantidade de nós informada corresponderá ao tipo de elemento. Para o programa foi implementada a leitura para elementos lineares de dois nós, triangulares de três nós e quadrilaterais de quatro nós. A leitura se encerra ao encontrar a marcação “\$EndElements”.

3.1.1.3 Inserção de atributos

Vinculado diretamente com a estrutura da malha importada, para a realização de uma análise em MEF é necessário determinar os atributos do modelo. Esses são, por exemplo, restrições nodais, material de cada elemento, modelo constitutivo, cargas aplicadas, deslocamentos prescritos,

Figura 3.2 – Exemplificação do rotulamento de blocos para nós



Fonte: O Autor.

entre outros. Portanto, para minimizar a dependência da interface gráfica do INSANE, foi necessário desenvolver um processo para facilitar o usuário a inserir os atributos da análise.

3.1.2 Pontos específicos sobre código no INSANE

3.1.2.1 Padrão “Command”

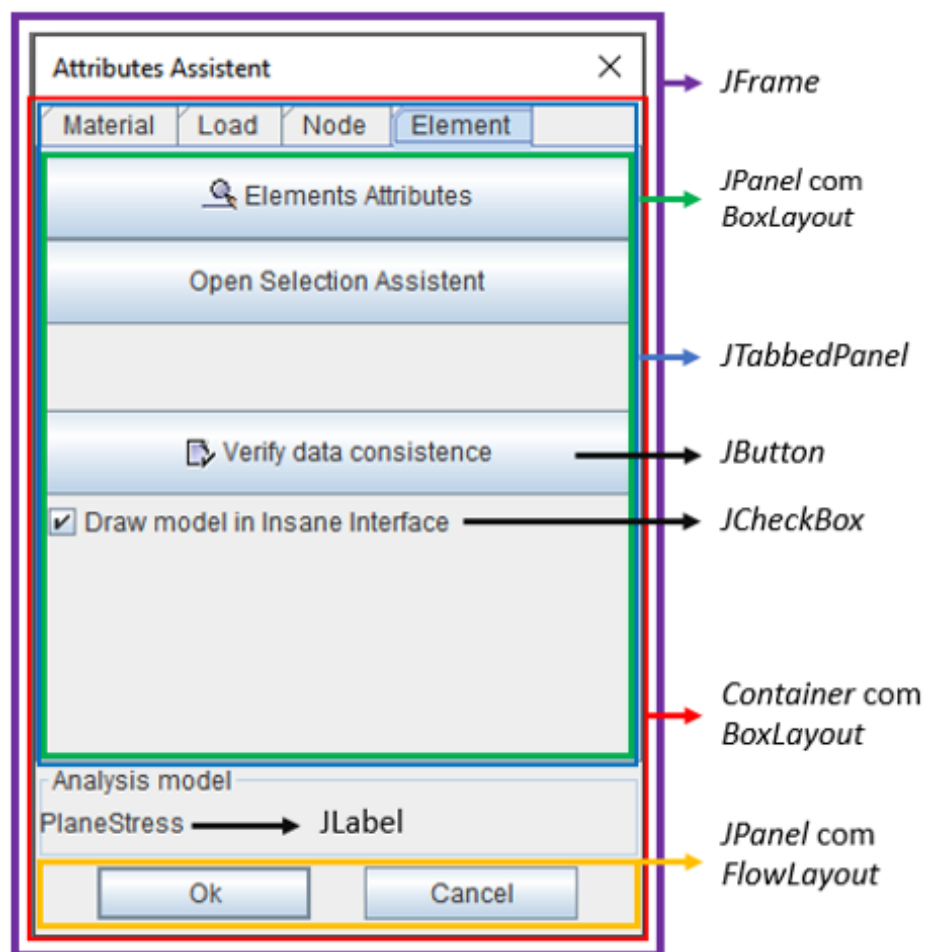
O Padrão *Command* é uma metodologia de organização de tarefas de execução comumente utilizada em programas em JAVA com interface gráfica. Esse padrão estabelece que os objetos interativos presentes na interface gráfica devem ter uma rotina encapsulada numa classe de comando, a qual deve possuir um método “*execute()*” que será acionado durante a interação. Dessa forma, o programa possui muito mais flexibilidade na implementação, pois os objetos de comandos são desvinculados das interfaces de chamada e possuem um tempo de execução próprio. Assim, é possível realocar e substituir esses comandos em outros sistemas do código sem ter problemas de compatibilidade.

3.1.2.2 A biblioteca SWING

A interface gráfica do INSANE é implementada a partir das classes e funções da biblioteca SWING. Essa biblioteca, considerada uma extensão da antiga AWT, incorpora uma série de componentes de fácil implementação para interfaces gráficas. Dessa forma, com o uso de objetos pré-implementados como o *JButton*, *JTextField*, *JPanel*, etc, é possível criar facilmente botões, caixas de texto, entre outros componentes. Além disso sua implementação é multi-plataforma e permite que o mesmo código funcione em Windows, Linux ou macOS sem a necessidade de adaptações.

Para utilizar a biblioteca normalmente cria-se uma classe herdeira da classe *JFrame* pertencente ao SWING. Com a definição do *frame* principal, adiciona-se painéis aglomerados, cada um com um padrão de preenchimento para posicionamento dos elementos internos. Dessa forma, define-se tamanhos mínimos, máximos e preferenciais para cada elemento compondo o *Front-end* da janela, a Figura 3.3 mostra um pouco do agrupamento dos painéis e componentes. Para definir a forma que os componentes irão interagir com o usuário, cria-se *Listeners* que são ativados conforme ações específicas, formando assim o *Back-end* da aplicação.

Figura 3.3 – Composição de uma interface implementada em SWING

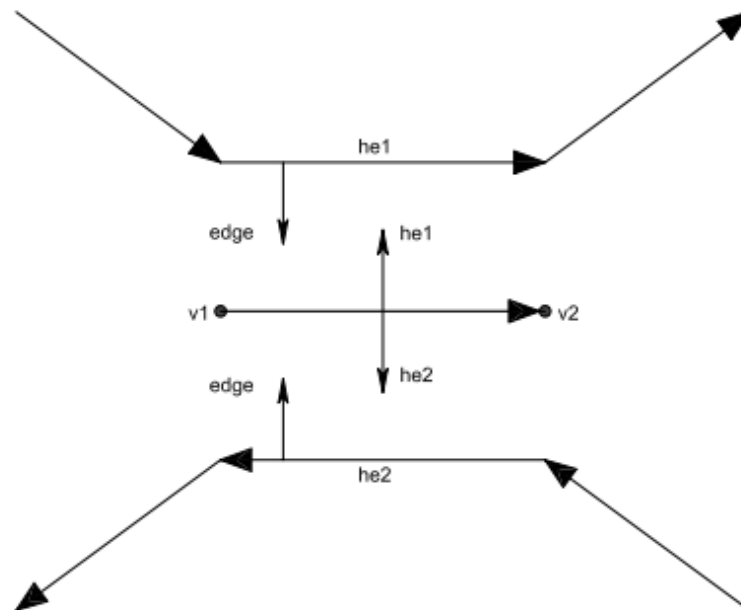


Fonte: O Autor.

3.1.2.3 A Estrutura de Dados de Geometria

O Sistema INSANE implementa do modelo de Semi-arestas para definição geométrica de suas entidades, conhecido como *Half Edge Data Structure* (HEDS). Essa estrutura recebe essa denominação por conta da transformação das arestas da geometria em duas semi-arestas que guiam os *loops* correspondentes às faces, como mostrado na Figura 3.4.

Figura 3.4 – A estrutura de semi-arestas



Fonte: Penna (2007) adaptado.

Penna (2007) descreve em seu trabalho a estrutura e a teoria que compõe esse modelo organizacional com mais detalhes. Como o processo implementado não demanda transformações geométricas, apenas leitura e referência, foi determinado que a malha seria lida num formato intermediário e só seria transformada para o modelo HEDS ao final do processo.

3.1.2.4 Classes essenciais à aplicação

O INSANE já possui um código robusto e estruturado, portanto, é importante entender como cada uma das classes interagem entre si e o que elas representam na aplicação. Dentre elas, algumas essenciais para a aplicação desenvolvida são:

- *InsaneInterface*: Representa a janela que o sistema operacional abre e incorpora os menus e comandos gerais do programa.
- *InternalInsaneInterface*: Representa as guias de trabalho que podem ser utilizadas dentro da aplicação. Para cada parte do processo utiliza-se uma guia que incorpora as ferramentas corretas para a etapa de modelagem e pré-processamento.

- *Combinable*: A classe que incorpora as informações gerais do modelo criado ao longo da modelagem e pré-processamento, como o modelo de análise, as formas de malhas utilizadas, a malha, o modelo constitutivo, os materiais, etc.
- *AnalysisModel*: A classe que contém as informações e métodos de construção de matrizes correspondentes ao modelo de análise aplicado, irá caracterizar as condições do modelo. Pode ser, por exemplo, um modelo pode ser em estado plano de tensão, ou simétrico ao eixo, etc.

3.1.3 Código implementado

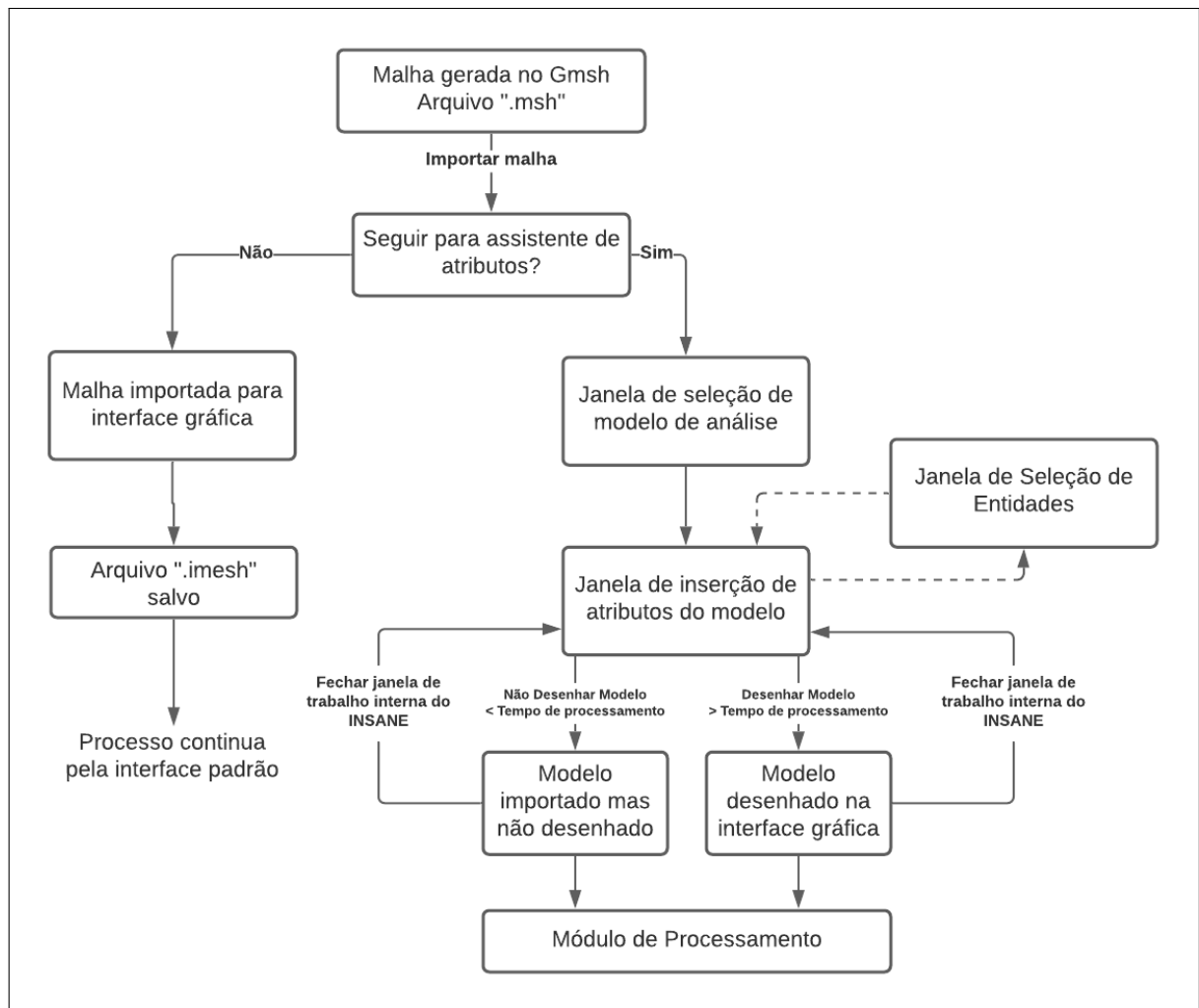
Seguindo o padrão de código das Seções 2.3 e 3.1.2 e os objetivos traçados na Seção 3.1.1 foi realizado a implementação da rotina de importação de malhas a partir da definição de diversas classes. A implementação foi guiada pelo fluxo de utilização pensado de acordo com o fluxograma presente na Figura 3.5.

Nas subseções a seguir, serão expostas e explicadas as classes implementadas para importação da malha e atribuição dos parâmetros do modelo.

3.1.3.1 Classes implementadas para importação de malhas

- Pacote: *br.ufmg.dees.insane.ui.rich.attributes.view.command*
 - *ImportGmshMeshFileCommand*: classe tipo *Command* que gerencia o processo de leitura do arquivo “.msh” e construção das classes respectivas para funcionamento INSANE. Fica localizado no menu *File* da classe *InsaneInterface*. Ao fim da leitura permite o usuário escolher se deseja continuar com um assistente de inserção de atributos ou gerar a malha na interface gráfica do programa.
- Pacote: *br.ufmg.dees.insane.parser.transcriptor*
 - *TranscriptionGmshToMesh*: classe responsável por coordenar as etapas de leitura que ocorrem sequencialmente de acordo com a seção que o ponteiro encontra-se no arquivo de entrada.
 - *FillingMeshFromGmsh*: classe que implementa rotinas de leitura e constrói as classes correspondentes aos vértices, arestas, faces e blocos de entidade e as armazena na classe *GmshMeshTranscription*.
 - *XmlWriterGmshToImesh*: classe ativada quando o usuário opta por gerar a malha na interface gráfica, responsável por salvá-la no formato “.imesh”.
- Pacote: *br.ufmg.dees.insane.parser.transcriptor.mesh*
 - *GmshMeshTranscription*: classe responsável por armazenar os objetos construídos e, ao fim da execução, gerenciar o processo de construção do modelo HEDS.

Figura 3.5 – Fluxograma de Projeto



Fonte: O Autor.

- *GmshElementTranscription*: interface que padroniza a nomenclatura dos métodos implementados nas classes que representam entidades geométricas.
- *GmshNodeTranscription*: classe que representa cada vértice da malha.
- *GmshEdgeTranscription*: classe que representa cada aresta da malha.
- *GmshFaceTranscription*: classe que representa cada face da malha.
- *GmshEntityBlock*: classe que representa cada um dos blocos lidos no arquivo de entrada. É armazenado para facilitar seleção de entidades no assistente de inserção de atributos

As classes que representam objetos geométricos do modelo, *GmshNodeTranscription*, *GmshEdgeTranscription* e *GmshFaceTranscription*, herdam os métodos e atributos das classes correspondentes no modelo HEDS (*Vertex*, *Edge* e *Face*, respectivamente), entretanto não possuem algumas das referências que tornam o modelo HEDS conciso. Isso foi realizado para permitir que a rotina implementada funciona-se com outros métodos e classes pré-existent no IN-

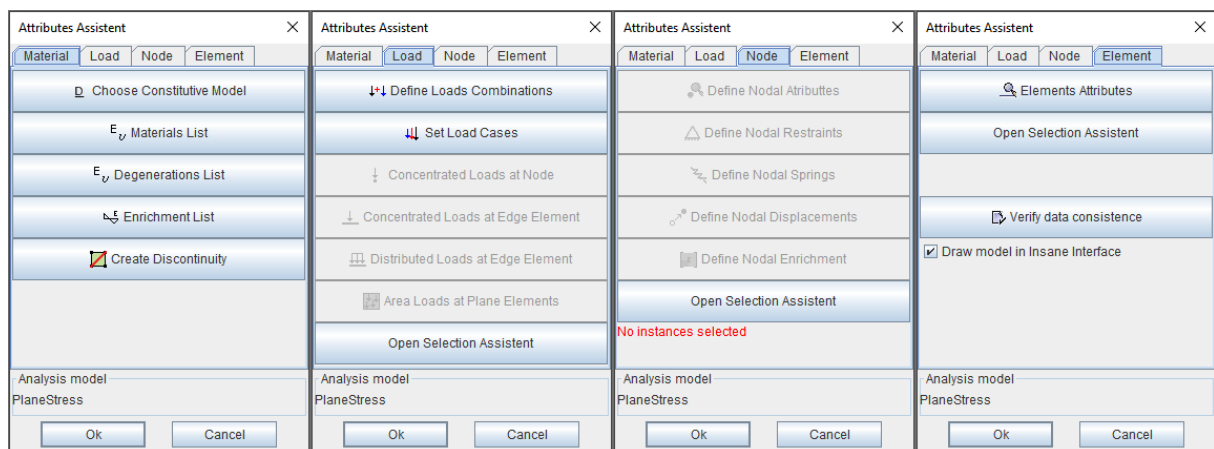
SANE. Dessa forma, não seria necessário reescrever as classes *Command* e tornaria mais fácil, posteriormente, a construção da estrutura em HEDS. Ao mesmo tempo, com as novas classes desenvolvidas, seria possível armazenar certos dados de forma mais eficiente à demanda da nova aplicação e preservar os rótulos e os blocos correspondentes aos elementos e nós no arquivo gerado pelo Gmsh.

3.1.3.2 Classes implementadas para inserir atributos do modelo

- Pacote: *br.ufmg.dees.insane.ui.rich.attributes.view.command*
 - *OpenGmshAttributesAssistantCommand*: classe tipo *Command* que constrói a janela que permite ao usuário inserir atributos no modelo.
- Pacote: *br.ufmg.dees.insane.ui.rich.attributes.view.dialog*
 - *GmshAttributesAssistantDialog*: classe que implementa uma janela interativa, em que o usuário pode incluir cada um dos atributos disponibilizados pelo INSANE para a análise correspondente, Figura 3.6, os atributos do modelo são armazenados na classe *Combinable* e os atributos dos nós e elementos nos respectivos objetos.
 - *SelectionAssistant*: classe que implementa uma janela interativa, em que o usuário pode utilizar de ferramentas de seleção para incluir e remover nós, arestas e faces, Figura 3.7.

3.1.3.3 A janela “Attributes Assistant”

Figura 3.6 – A janela *GmshAttributesAssistantDialog*. Alguns botões permanecem inativos enquanto não possuem entidades selecionadas.



Fonte: Sistema INSANE.

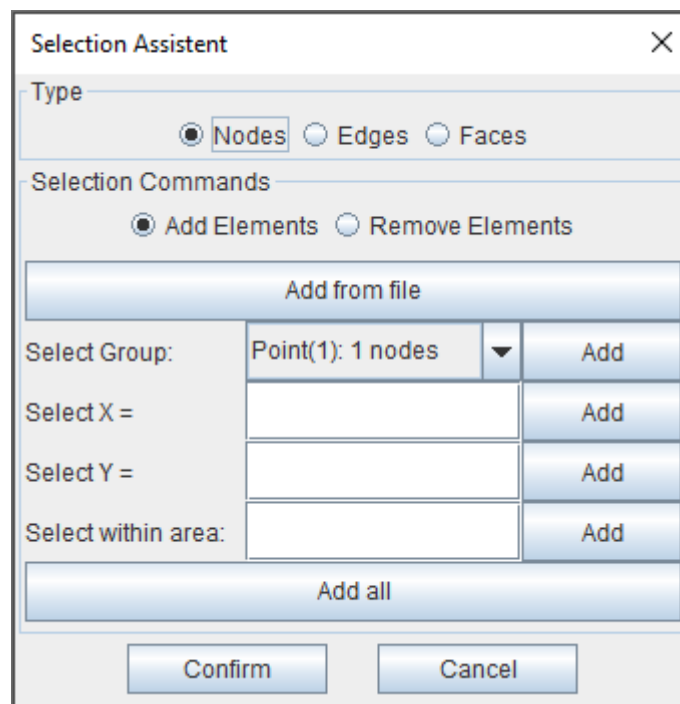
Na janela *Attributes Assistant*, Figura 3.6 o usuário encontra botões para todas as ferramentas que o INSANE disponibiliza para modelos planos e que seriam encontrados na respectiva *InternalInsaneInterface*. A janela foi separada em 4 abas: a primeira, *Material*, contém ferramentas

para definição do modelo constitutivo, materiais e propriedades do tipo de análise, a aba *Load* possui comandos para gerenciamento e aplicação de carregamentos na estrutura, em *Node*, o usuário pode definir os atributos nodais, como restrições, rigidez pontual, entre outros, por fim, na aba *Element*, é possível atribuir materiais a cada elemento, verificar se o modelo possui todas propriedades básicas para análise e alternar uma opção que permite desativar a geração dos elementos gráficos do modelo. Essa opção foi implementada pois a construção do modelo visual pode ser demorada e dispensável, uma vez que a janela de atributos fornece todas as ferramentas para inserir os atributos.

A qualquer momento o usuário pode gerar o modelo clicando no botão “Ok”. Nesse momento a estrutura HEDS será montada, com os atributos inseridos até o momento e o modelo será vinculado à uma janela de trabalho interna no INSANE que se abrirá, permitindo continuar a formulação do modelo por lá ou seguir para a análise. A janela de inserção de atributos não será fechada e possibilita que o usuário feche a janela interna do INSANE, realize alterações no modelo e o construa novamente.

3.1.3.4 A janela “Selection Assistant”

Figura 3.7 – A janela *SelectionAssistant*



Fonte: Sistema INSANE.

Para realizar a seleção de entidades, no intuito de inserir propriedades nodais, carregamentos, etc, a janela *Selection Assistant*, Figura 3.7, disponibiliza ferramentas para seleção de acordo com os rótulos das entidades, posição e bloco que pertence. As entidades de cada bloco são importadas da malha do Gmsh que as organiza de acordo com as geometrias que levaram a geração da malha. Dessa forma, os nós que se encontram na mesma posição que os pontos da

geometria original são inseridos em blocos marcados com dimensão “0” e com o rótulo igual ao do ponto. Para nós que se encontram numa linha da geometria original, ocorre sua inserção em blocos marcados com dimensão “1”, com o mesmo rótulo que linha possuía. Ocorre um processo similar para nós dentro de superfícies da geometria e para as arestas e faces da malha. Portanto, ao armazenar a estrutura de blocos do Gmsh, torna-se fácil a seleção de entidades para o usuário. A Figura 3.2 ilustra a enumeração dos blocos.

3.2 IMPLEMENTAÇÃO DE MODELOS DE CAMPO DE FASE

Como comentado anteriormente, o Sistema INSANE já possuía variados métodos implementados para modelos de Campo de Fase, porém vários outros tiveram que ser desenvolvidos para a realização desse trabalho. Nessa seção será dissertado sobre a metodologia de implementação deles e as alterações no código necessárias para isso.

3.2.1 Funções com valores definidos pelo usuário

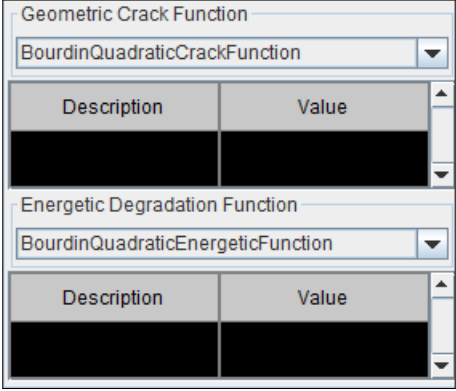
Foi citado na Seção 2.2.2.1 sobre o interesse na implementação da Função de Geometria da Trinca de Wu (2017) e na Seção 2.2.3.2 da Função de Degradação Energética do mesmo autor. Ambas, diferentemente das outras implementações para os respectivos propósitos, possuem valores que devem ser definidos antes da análise. Portanto, foi realizado a generalização do código para trabalhar com funções dos dois tipos.

No Sistema INSANE as cada uma das funções de geometria da trinca e degradação energética são classes inferiores que “herdam” propriedades das classes superiores *GeoCrackFunction* e *EnergeticDegradationFunction*, respectivamente. As classes superiores implementam métodos de criação de objetos e listagem das classes inferiores disponíveis. Para possibilitar que o usuário forneça os dados da função e que os valores necessários fossem exibidos na interface gráfica, foram implementados métodos superiores padrões para todas as classes que seriam posteriormente sobrescritos nas novas classes que demandavam de valores específicos.

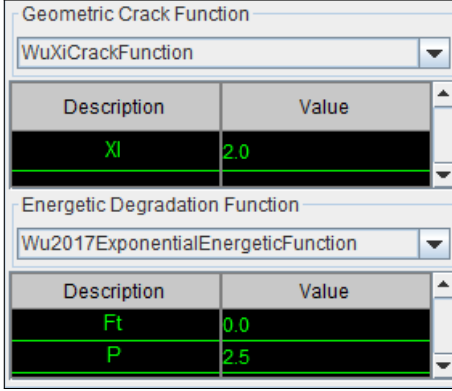
Nesse intuito, foi alterado o método de criação de objetos, introduzindo a necessidade de informar um dicionário, especificamente um *HashMap*, com os valores padrões das funções. Ao mesmo tempo, para não interferir nas funções que não demandam valores, foi criado um método nas classes superiores que retornavam os valores padrões como um dicionário vazio. Desse modo, apenas classes que demandam valores e que sobrescrevem esse método recebem valores definidos pelo usuário. O resultado impresso na interface pode ser visualizado na Figura 3.8.

Para a Função de Geometria da Trinca foi implementada a classe *WuXiCrackFunction()* com métodos para retornar os valores da primeira e segunda derivada em relação a ϕ , como requisitado pelos algoritmos de solução do programa. Nesse sentido, também foi implementado a classe *Wu2017EnergeticFunction()*, com métodos de mesmo objetivo, para a Função de Degradação Energética citada.

Figura 3.8 – Interface gráfica para funções de campo de fase: (a) funções sem valores a serem informados; (b) funções com valores a ser informados



(a)



(b)

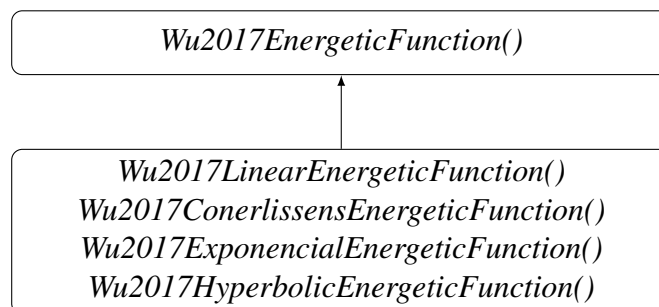
Fonte: Sistema INSANE.

3.2.2 Implementação das leis de amolecimento

Conforme citado na Seção 2.2.3.3, os valores da Função de Degradação Energética de Wu (2017) podem ser derivados de leis de amolecimento específicas. Entretanto, determinar os parâmetros necessários pode ser uma tarefa inconveniente, logo, foi realizada a implementação individual de cada dessas leis.

Para a criação das respectivas classes, foram implementadas as relações de a_1 , a_2 , a_3 e p em relação aos parâmetros específicos, conforme explicitado nas Equações 2.13 a 2.25. Assim, para cada classe inferior é chamado no programa um método de inicialização de variáveis, denominado *initVariables()*, com o objetivo de calcular os valores de interesse e preparação da classe superior para a análise. Desse modo, pode se escrever a hierarquia das classes implementadas conforme a Figura 3.9.

Figura 3.9 – Hierarquia das funções de degradação energéticas de Wu (2017)



Fonte: O autor.

3.2.2.1 Implementação dos métodos de solução

O método de solução alternado com variável histórica foi implementado no Sistema INSANE no trabalho de Leão (2021), assim como as classes responsáveis por montar o modelo em MEF.

Observa-se que o método de solução com constrição de contorno difere-se daquele que usa variável histórica apenas na sub-rotina de convergência da variável de campo de fase, enquanto as outras etapas são idênticas ao método alvo. Para melhor visualização o Algoritmo 1 apresenta a rotina de solução implementada em Leão (2021).

Algoritmo 1: Método alternado com variável histórica

Dados: $(\bar{d}_n, \bar{a}_n, \mathcal{H}_n)$
Resultado: $(\bar{d}_{n+1}, \bar{a}_{n+1}, \mathcal{H}_{n+1})$
Enquanto $n \leq n^\circ \text{ máximo de passos}$ **faça**

 Inicialização: $(\bar{d}_{n+1}^0, \bar{a}_{n+1}^0, \mathcal{H}_{n+1}^0) = (\bar{d}_n, \bar{a}_n, \mathcal{H}_n)$;

 Calcula $[\mathbf{K}^{\text{uu}}]^0$;

 Calcula $\Delta \bar{d}_{n+1}^1$ conforme estratégia iterativa;

 Define $\bar{d}_{n+1}^1 = \Delta \bar{d}_{n+1}^1 + \bar{d}_{n+1}^0$;

 Define $i = 1, j = 1$ e $k = 0$;

Enquanto $\frac{\|\Delta \bar{d}_{n+1}^i\|}{\|\bar{d}_{n+1}^i\|} < \text{tolerância global ou } i \leq n^\circ \text{ máximo de iterações}$ **faça**

 Define $\Delta \bar{d}_{n+1}^{i,1} = \Delta \bar{d}_{n+1}^i$ e $\bar{d}_{n+1}^{i,1} = \bar{d}_{n+1}^i$;

Enquanto $\frac{\|\Delta \bar{d}_{n+1}^{i,j}\|}{\|\bar{d}_{n+1}^{i,j}\|} < \text{tol. local ou } j \leq n^\circ \text{ máx. de iter.}$ **faça** // Conv. desloc.

 Calcula $[\mathbf{K}^{\text{uu}}]^{i,j}$;

 Calcula $\Delta \bar{d}_{n+1}^{i,j+1}$ conforme estratégia iterativa;

 Define $\bar{d}_{n+1}^{i,j+1} = \Delta \bar{d}_{n+1}^{i,j+1} + \bar{d}_{n+1}^{i,j}$;

 $j = j + 1$;

Fim
Para $k \leq n^\circ \text{ máximo de iterações}$ **faça** // Conv. variável de campo

 Calcula $[\mathbf{K}^{\phi\phi}]^k$ e $r^{\phi,k}$, usando \mathcal{H}_{n+1}^k ;

 $\Delta \bar{a}_{n+1}^k = [\mathbf{K}^{\phi\phi}]^{-1,k} \cdot \bar{r}^{\phi,k}$;

 $\bar{a}_{n+1}^{k+1} = \bar{a}_{n+1}^k + \Delta \bar{a}_{n+1}^k$;

 $\mathcal{H}_{n+1}^{k+1} = \max(\mathcal{H}_{n+1}^k, \bar{Y}_{n+1}^{k+1})$;

 $k = k + 1$;

Se $\frac{\|\Delta \bar{a}_{n+1}^k\|}{\|\bar{a}_{n+1}^k\|} < \text{tolerância local}$ **então**

Sai do bloco iterativo;

Fim
Fim

 Calcula $[\mathbf{K}^{\text{uu}}]^i$;

 Calcula $\Delta \bar{d}_{n+1}^{i+1}$ conforme estratégia iterativa;

 $\bar{d}_{n+1}^{i+1} = \Delta \bar{d}_{n+1}^{i+1} + \bar{d}_{n+1}^i$;

 $i = i + 1, j = 1$ e $k = 0$
Fim
Fim

Pode-se observar alguns pontos importantes nesse algoritmo. Primeiramente, antes de começar o bloco de repetição global, é realizada uma iteração de deslocamento para desbalanceamento do sistema e para possibilitar o uso do critério de convergência adotado. Em segundo lugar,

como durante a convergência da variável de campo de fase ocorre o desbalanceamento da variável de deslocamentos, pois permanece constante, assim, pode-se avaliar a convergência global a partir da magnitude desse efeito. Portanto, realiza-se novamente uma iteração nos deslocamentos, ao fim da rotina global, para avaliar a convergência.

Para a implementação do algoritmo de solução baseado na constrição de contorno foi criada uma nova classe inferior à classe que usa variável histórica, assim, foi sobrescrito apenas a sub-rotina de convergência da variável de campo de fase. Desse modo, os outros métodos da classe foram preservados sem a necessidade de repetição de código.

Atentando-se as especificidades desse novo método, além das definições já trabalhadas na Seção 2.2.5.3, foi feito o desenvolvimento do operador de restrição. Esse foi feito baseando-se no trabalho de Yang et al. (2016), porém com a adoção de adaptações para os limites do modelo desenvolvido nesse trabalho. A Equação 3.1, apresenta essa função.

$$F_{\Theta}(\bar{a}_{n+1})_I = \begin{cases} r_I^{\phi}, & \text{para } a_{I,n} < a_{I,n+1} < 1 \\ \min(r_I^{\phi}, 0), & \text{para } a_{I,n+1} = 1 \\ \max(r_I^{\phi}, 0), & \text{para } a_{I,n+1} = a_{I,n} \end{cases} \quad (3.1)$$

O operador de restrição é importante pois ao contrário do modelo tratado anteriormente, o funcional de energia é uma desigualdade, permitindo valores diferente de 0 para o resíduo. Além disso, ele é usado para o algoritmo utilizado na procura linear. Nesse contexto, foi implementada a procura linear retroativa, em Yang et al. (2016) essa é dada pela Equação 3.2.

$$\|F_{\Theta}(\pi [a_n^k + \lambda^l p^{k+1}])\| \leq (1 - \tau \lambda^l) \|F_{\Theta}(a_n^k)\| \quad (3.2)$$

Sendo p^{k+1} a direção de busca, definida como a junção do vetor tangente das componentes inativas com um vetor $\bar{0}$ para os índices ativos, τ um coeficiente de redução e λ^l , com $l = [0, 1, 2, 3, \dots]$, o multiplicador definido para valores acima de um limite Λ previamente estabelecido. Nesse trabalho foi utilizado os valores de $\lambda = 0,5$, $\tau = 10^{-4}$ e $\Lambda = 10^{-12}$, retirados do trabalho de Benson e Munson (2006). A procura linear encontra um valor ótimo quando a condição acima é atingida, caso nenhum valor de λ^l satisfaça a relação, é utilizado o λ^l que promoveu a maior minimização durante a rotina. Desse modo, como mostrado no Algoritmo 2, a nova sub-rotina é montada a partir definições descritas.

Para esse método foi feita a utilização de um novo critério de convergência. Isso se deve ao fato que a utilização da procura linear tende a diminuir o valor da variação do vetor para a iteração trabalhada. Assim, o uso do critério anterior poderia causar situações de falsa convergência. O novo critério faz uso da minimização do módulo do vetor gerado pelo operador de restrição e atende perfeitamente os propósitos de minimização da análise.

Esse novo método de solução é apontado por Wu et al. (2020) como menos eficiente em comparação aos outros citados, porém é extremamente robusto, possibilita uma utilização mais abrangente e abre a possibilidade de aprofundar ainda mais nas formulações de campo de fase. Nesse intuito, existem projetos para incorporar ao INSANE o uso de bibliotecas externas mais

Algoritmo 2: Método com constrição de contorno: sub-rotina de conv. de campo de fases

Dados: $(\bar{d}_{n+1}^{i,j}, \bar{a}_{n+1}^0)$

Resultado: (\bar{a}_{n+1}^k)

Enquanto $\|\bar{F}_\Theta(\bar{a}_{n+1}^k)\| > \text{tolerância local faça}$

 Separa de conjuntos: ativos (\mathcal{A}) e inativos (\mathcal{I});

 Calcula $[\mathbf{N}^{\phi\phi}]^k$;

 Remove índices ativos e define $[\mathbf{N}^{\phi\phi}]_{\mathcal{I}}^k$;

 Calcula $\bar{p}_{\mathcal{I}}^{k+1} = [\mathbf{N}^{\phi\phi}]_{\mathcal{I}}^{-1,k} \cdot \bar{r}_{\mathcal{I}}^{\phi,k}$ e define $\bar{p}_{\mathcal{A}}^{k+1} = \bar{0}$;

 Define a direção de busca como $\bar{q}^{k+1}(\lambda^l) = \pi(\bar{a}_n^k + \lambda^l \bar{p}^{k+1})$;

 Realiza a procura linear para $\|\bar{F}_\Theta(\bar{q}^{k+1}(\lambda^l))\|$ e define um valor ótimo de $\lambda^l = c$;

 Define $\bar{a}_{n+1}^{k+1} = \bar{q}^{k+1}(c)$;

$k = k+1$;

Fim

otimizadas para diminuir o tempo de processamento, mas como esses programas são muitas vezes fechados, ter um algoritmo desenvolvido internamente trás vantagens na facilidade de customização dos processos.

3.2.2.2 Implementação de modelos constitutivos

Os modelos constitutivos de Campo de Fase no sistema INSANE são definidos de forma diferente para os métodos de solução monolítico e alternado. Portanto, visto o enfoque desse trabalho em métodos alternados, a implementação será restrita ao mesmo. Dessa forma, o modelo constitutivo usado em Wu (2017), implementado nesse trabalho, possui como classe superior a *PhaseFieldStaggeredConstitutiveModel*, que define métodos padrões para todas as classes desenvolvidas para métodos de solução alternados.

Como foi descrito na Seção 2.2.6.3, o modelo em questão é definido como híbrido, por não ter em sua formulação a realização da separação de componentes, mas, mesmo assim, definir um comportamento assimétrico para tensões de tração e compressão. Dessa forma, as formulações se assemelham bastante aos modelos sem separação de componentes.

A estratégia utilizada para a implementação, foi a definição desse novo modelo como uma classe inferior, *StgPfWu2013ConstModel*, à classe do modelo isotrópico de Bourdin et al. (2000), *StgPfIsotropicConstModel*. Dessa forma, foi necessário apenas sobrescrever o método “*mountDualInternalVariableVector()*” que retorna as tensões de cada nó e atualiza o valor de \bar{Y} . Assim, a tensão retornada pelo método deixou de ser o ψ_0 correspondente a elasticidade inicial, Equação 2.45b, para retornar o valor baseado na tensão equivalente, Equação 2.54.

4 ANÁLISES E RESULTADOS

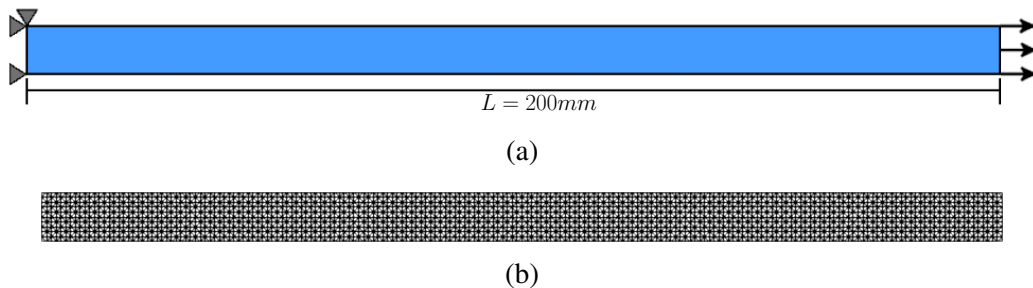
4.1 VALIDAÇÃO DAS IMPLEMENTAÇÕES

Ao longo do Capítulo 3 foram expostas várias novas implementações: a nova interface para importação e preparação de malhas refinadas do Gmsh, a função de geometria da trinca e função de degradação energética de Wu (2017), o modelo constitutivo do mesmo autor e o solver baseado na constrição de contorno. Nas análises de campo de fase apresentadas a seguir, todas foram realizadas a partir da interface desenvolvida e com o método de solução implementado. As outras funções que foram incluídas no desenvolvimento desse trabalho foram utilizadas em variadas combinações.

4.1.1 Validação com tração unidimensional

Para os testes, primeiramente, foram reproduzidas algumas das análises de tração uniaxial apresentadas em Wu (2017). A composição do problema pode ser vista na Figura 4.1a, basicamente a análise consiste em uma simulação da resposta material à condição de tração uniaxial. Portanto, foi modelado uma barra de comprimento $L = 200 \text{ mm}$, altura $b = 10 \text{ mm}$ e espessura $w = 1 \text{ mm}$ em estado plano de tensões. Assim, foi gerada uma malha triangular estruturada com elementos de tamanho médio $h = 1 \text{ mm}$, Figura 4.1b.

Figura 4.1 – Teste uniaxial de tração: (a) modelo de análise; (b) malha;



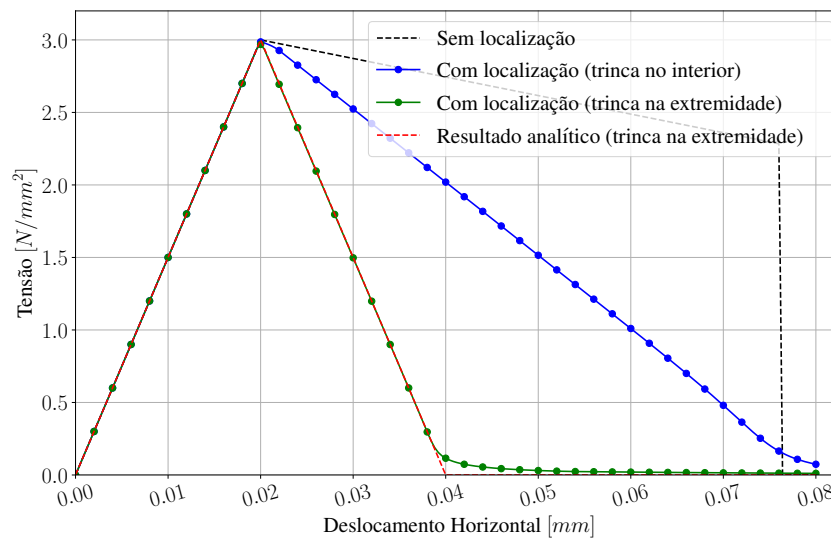
Fonte: O Autor.

O modelo foi montado com os nós à esquerda com restrição de deslocamento em x e um à borda com restrição em x e y. Foi aplicada uma carga pontual em um dos nós a direita e estabelecido uma relação *master-slave* no eixo x com os outros nós na mesma borda, ou seja, eles se deslocariam na mesma proporção. Os parâmetros de análise foram os mesmos utilizados por Wu (2017): $E_0 = 3 \cdot 10^4 \text{ MPa}$, $\nu = 0,2$, $f_t = 3,0 \text{ MPa}$ e $G_f = 0,15 \text{ N/mm}$. Foi optado pela utilização de $l_0 = 20 \text{ mm}$ para seguir a proporção $l_0/h = 20$ usada no trabalho do autor. Para os métodos de controle utilizou-se controle direto de deslocamentos para o nó no qual aplicou-se a carga com incrementos de $4 \cdot 10^{-4} \text{ mm}$, uma tolerância local de $1 \cdot 10^{-6}$ e global de $1 \cdot 10^{-4}$.

Os primeiros resultados apresentaram problemas de localização de trinca. Ou seja, como era

uma malha completamente homogênea e com carga uniformemente distribuída, a análise não iniciava nenhuma degradação localizada. Assim, foi utilizado a abordagem apresentada em Leão (2021), definindo 2 elementos de cada lado da barra, simétricos ao eixo x , para terem um valor de $G_f = 0,015$, logo, com a energia de fratura menor a trinca se iniciaria naquela posição. Como os resultados são baseados numa região de degradação, essa alteração não causaria variações significativas no resultado, pois a energia se dissipa em toda região de trinca difusa. A Figura 4.2 ilustra o problema e os resultados encontrados para a lei de amolecimento linear.

Figura 4.2 – Teste de tração com lei de amolecimento linear



Fonte: O Autor.

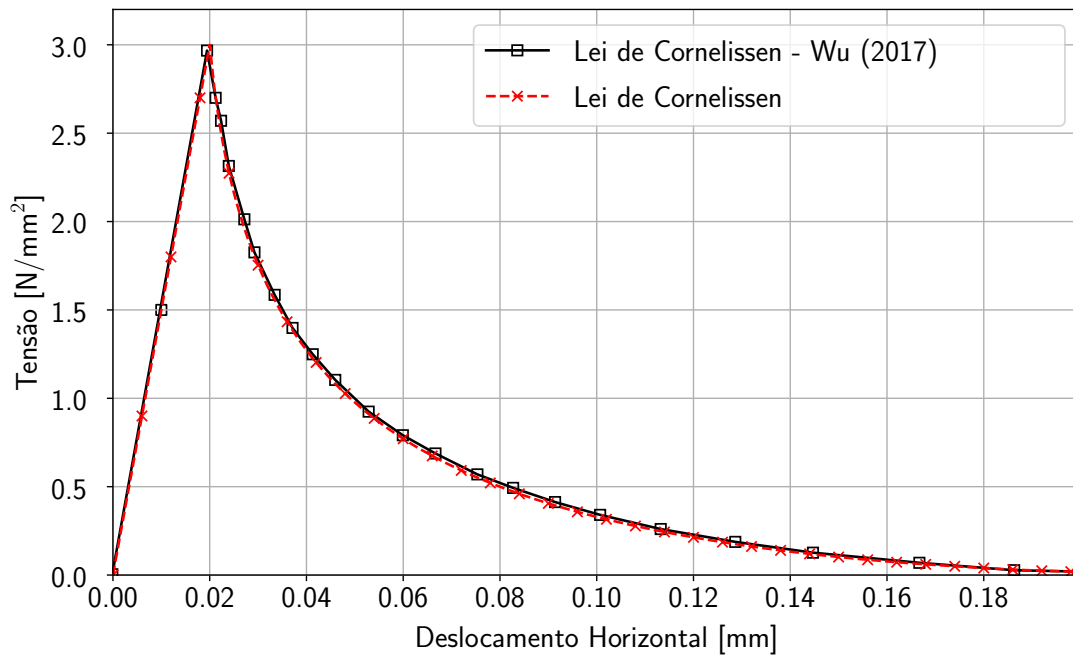
Foi realizado, também, o experimento para a lei de amolecimento de Cornelissen com $\eta_1 = 3,0$ e $\eta_2 = 6,930$, valores típicos para concreto, sendo os outros parâmetros idênticos à análise anterior. Para essa simulação utilizou-se a concentração da trinca no interior da barra. Ademais, foi necessário definir, como em Wu (2017), uma condição de Dirichlet de $\phi = 0$ para ambas as bordas, para garantir que a trinca não se formaria em tais posições. A curva Tensão-deslocamento para esse experimento pode ser vista na Figura 4.3. Além disso, a Figura 4.4 ilustra o o padrão que a variável de campo de fase gerou na barra durante as análises.

Os resultados encontrados foram idênticos aos da referência, Wu (2017), portanto, indicam que a implementação foi similar. Ressalta-se que o autor da referência não cita a utilização de nenhum mecanismo de localização da trinca, mas usa uma malha desestruturada e informa que a não-uniformidade da mesma levou à concentração da trinca.

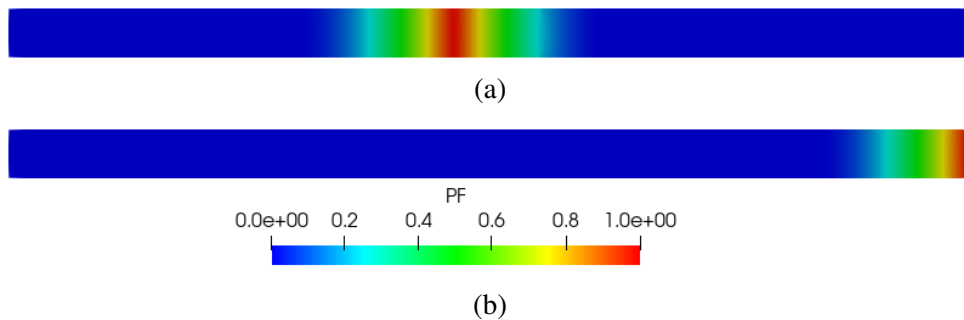
4.1.2 Validação com Painel em L

Os modelos anteriores apresentavam apenas o comportamento do material dentre uma modelagem simples. Para verificar se a implementação está de acordo com o esperado foi realizado um dos exemplos apresentados em Wu (2017), a resposta estrutural de um painel em L.

Figura 4.3 – Teste de tração com lei de amolecimento de Cornelissen



Fonte: O Autor.

Figura 4.4 – ϕ em teste uniaxial de tração: (a) lei de Cornelissen; (b) Lei linear;

Fonte: O Autor.

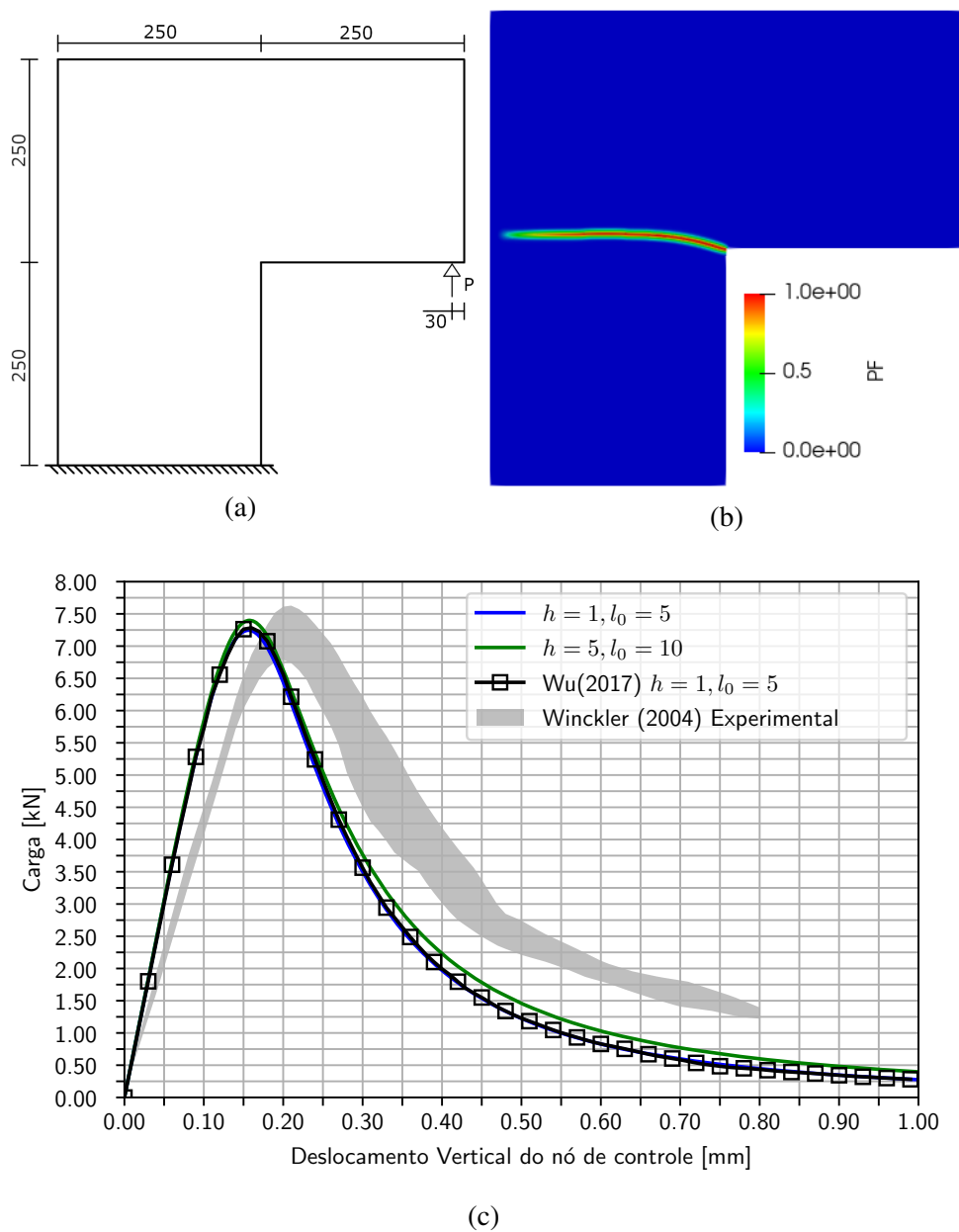
Essa análise, baseada no ensaio experimental de Winkler et al. (2004), foi feita pela montagem utilizada em Wu (2017), Figura 4.5a. O autor realiza a modelagem com a aplicação de uma carga vertical a 30 mm da borda do painel, utiliza os mesmos valores dentre as propriedades mecânicas encontradas nos ensaios experimentais, exceto o valor de G_f , que foi alterado para melhorar adequação da curva de resposta. Portanto, os valores usados foram: $E_0 = 25850$ MPa, $\nu = 0,18$, $f_t = 2,7$ N/mm², $G_f = 0,09$ N/mm². Novamente, foi utilizado a lei de amolecimento de Cornelissen com $\eta_1 = 3,0$ e $\eta_2 = 6,930$.

Para esse trabalho foram feitas 2 simulações, uma com a região refinada com $h = 5$ mm e um segundo com o refinamento de $h = 1$ mm, para os quais usados $l_0 = 10$ mm e $l_0 = 5$ mm, respectivamente. O método de controle direto de deslocamentos foi utilizado com passos de 0,002 mm no nó de aplicação de carga, tolerância local de $1 \cdot 10^{-5}$ e global de $1 \cdot 10^{-4}$. A Figura 4.5b, mostra o caminho da trinca para a simulação mais refinada e a Figura 4.5c mostra os resultados

encontrados.

As curvas conseguiram obter um formato similar aos ensaios experimentais, apesar de não terem uma interseção de valores significativa. Ressalta-se que o resultado obtido na simulação refinada, $h = 1$ mm teve resultado idêntico ao encontrado por Wu (2017) com os mesmos parâmetros, mostrando que a implementação das funções do autor foram feitas com sucesso e que o algoritmo de solução desenvolvido consegue encontrar resultados compatíveis com os da literatura. Os resultados com refinamento de $h = 5$ mm não pôde ser comparado, pois o autor em referência não utiliza desse valor.

Figura 4.5 – Painel em L de Wu (2017): (a) Esquema de montagem; (b) Caminho da trinca; (c) Curvas de resposta para nó de controle



Fonte: O Autor.

4.2 EXEMPLOS NUMÉRICOS

Para comparar os resultados de modelos de campo de fase e degradação elástica, foram adaptados, como mostrado a seguir, alguns exemplos discutidos no trabalho de Penna (2011). Foram comparados os resultados entre os modelos e valores experimentais. Para as análises optou-se por utilizar, quando possível, as mesmas propriedades dos materiais e montagens similares aos modelos apresentados no trabalho do autor. Apesar disso, para as malhas essa não seria uma opção viável, pois, como proposto por Wu (2017), uma proporção de $h \approx (l_0/5, l_0/10)$ é importante para descrever corretamente o campo de fase. Ademais, para calibrar o material de análise, foram realizados testes de tração uniaxial em um elemento quadrilateral de lado 1 mm. Isso permitiu visualizar o comportamento do material sob tração e escolher para os modelos de campo de fase um valor de l_0 que lhes garantissem um comportamento similar aos de degradação elástica. Todas as análises foram calibradas para ter uma resposta material similar ao material de Carreira-Ingraffea para tração.

4.2.1 Flexão em 3 Pontos

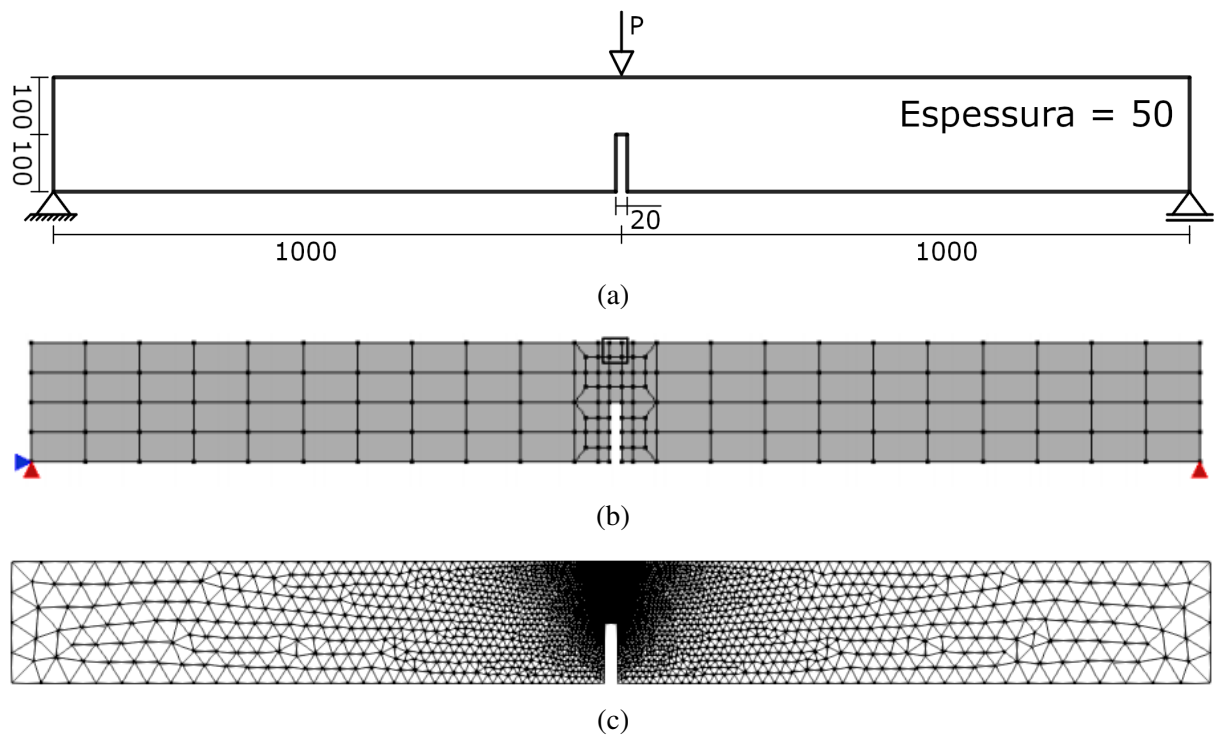
O modelo de flexão em 3 pontos analisa a resposta de uma viga bi-apoiada com um rasgo central a uma carga pontual oposta ao corte que gera flexão no sentido de abertura da trinca, Figura 4.6a. Respostas experimentais para esse modelo foram apresentadas no trabalho de Petersson (1981) para um material com propriedades f_t e G_f determinadas experimentalmente. Para essa análise a Figura 4.6b refere-se a malha usada em Penna (2011) e a Figura 4.6c a malha usada nesse trabalho.

Para a análise, foram usados os mesmos valores apresentados por Penna (2011): $E_0 = 30000$ N/mm², $f_t = 3,3$ N/mm², $f_c = 33,3$ N/mm² e $G_f = 0,124$ N/mm². A calibração ocorreu de forma que os modelos constitutivos de Wu (2017) e de Miehe et al. (2010) tivessem o mesmo comportamento material. A Figura 4.7 apresenta os resultados para o valor de $l_0 = 12,5$ mm, utilizado na análise. Para a lei de amolecimento de Cornelissen optou-se por utilizar os valores para concreto normal, $\eta_1 = 3,0$ e $\eta_2 = 6,930$ e para a lei exponencial, usou-se $p = 2,5$.

A estratégia de controle escolhida foi o controle direto dos deslocamentos, em que acompanhou-se a variação da posição do ponto de aplicação de carga para passos de 0,004 mm no sentido negativo de y. Além disso, usou-se uma tolerância para a convergência local das variáveis de $1 \cdot 10^{-5}$ e para a convergência global de $1 \cdot 10^{-4}$. A Figura 4.8 mostra os resultados encontrados num gráfico que exhibe a faixa de resultados experimentais, resultados de fissuração distribuída, encontrados por Penna (2011), e os de campo de fase realizados para esse trabalho.

Pode-se observar que as curvas de resposta obtiveram resultados similares e que as que usaram a lei de amolecimento de Cornelissen percorreram todo intervalo apresentado dentro da faixa de dados experimentais. Outro ponto a se considerar é que a curva resultante do modelo de campo de fase foi bem mais suave que os resultados dos modelos de fissuração distribuída o que pode ser um indício de maior estabilidade da solução.

Figura 4.6 – Flexão de 3 pontos: (a) modelo; (b) malha de Penna (2011); (c) malha utilizada.



Fonte: (a) e (c) o autor. (b) Penna (2011) adaptado.

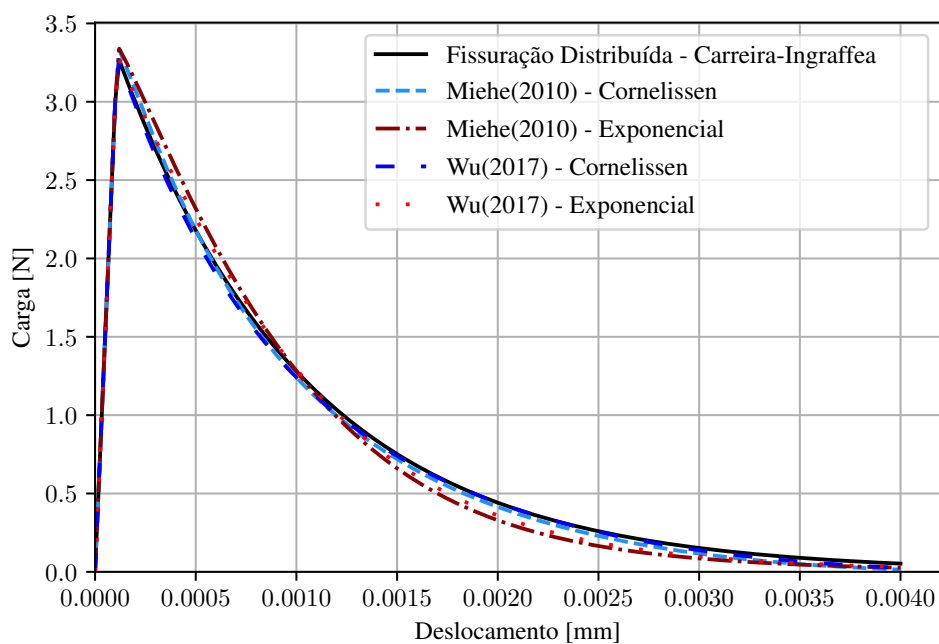
Além disso, é interessante apontar que o modelo constitutivo híbrido de Wu (2017) obteve resultados quase que idênticos que o de separação de variáveis de Miehe et al. (2010). Isso é importante visto que, por não realizar, separação de variáveis, os modelos híbridos costumam convergir significativamente mais rápidos.

A visualização da distribuição da variável ϕ no modelo encontra-se na Figura 4.9. Percebe-se que o amolecimento exponencial gerou uma degradação ligeiramente maior para o mesmo deslocamento. Apesar disso, o caminho do campo de fase foi conforme o esperado para ambos os modelos.

Ademais, por fins de comparação, foi montada outras análises com $l_0 = 10$ mm e uma com $l_0 = 2.5$ mm para avaliar o comportamento dos resultados em vista a mudança do comprimento característico. Para as novas análises foram usadas as mesmas propriedades mecânicas e métodos de convergência, entretanto a malha teve que ser refinada para garantir a proporção $h = l_0/5$ para a análise mais refinada. A Figura 4.10 expõe os resultados encontrados.

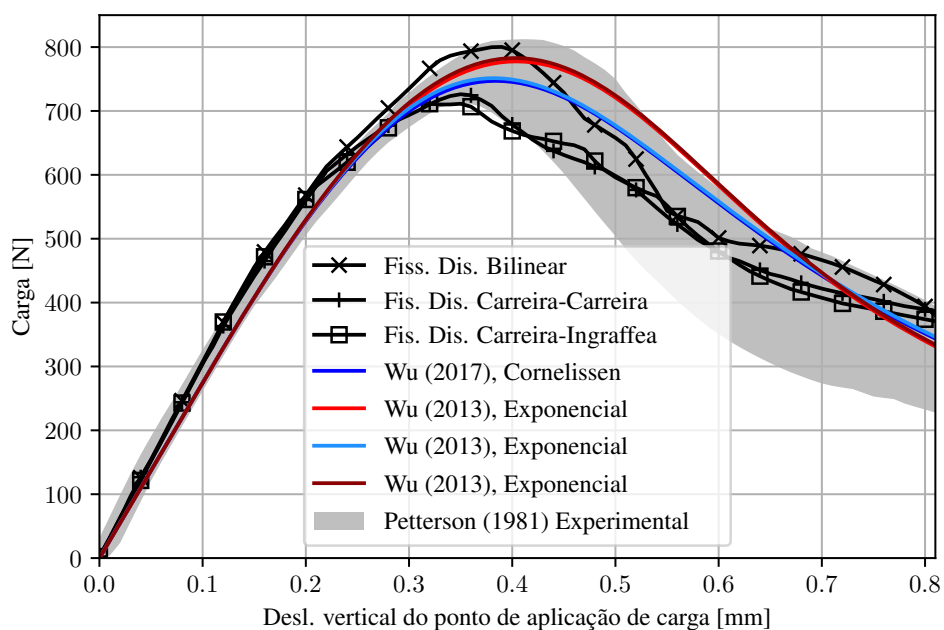
Como esperado, menores valores do comprimento característico resultaram em valores sobre-estimados de carga. Mas, é interessante visualizar que a mudança causou maior impacto para o amolecimento exponencial em comparação aos resultados do amolecimento de Cornelissen que mostrou-se menos sensível a tais variações.

Figura 4.7 – Parametrização flexão de 3 pontos



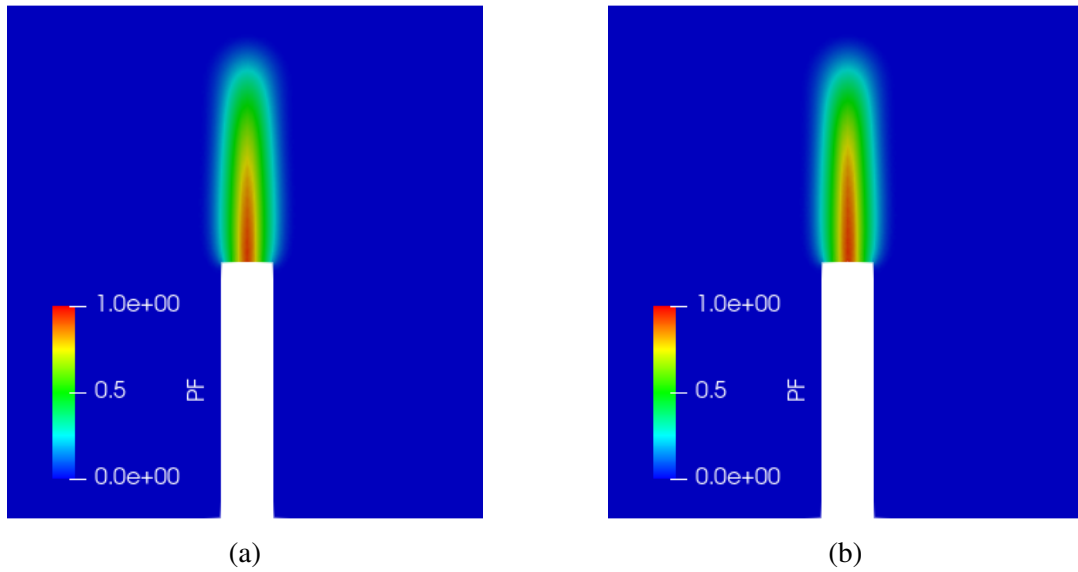
Fonte: O autor.

Figura 4.8 – Resultados para análise de flexão de 3 pontos

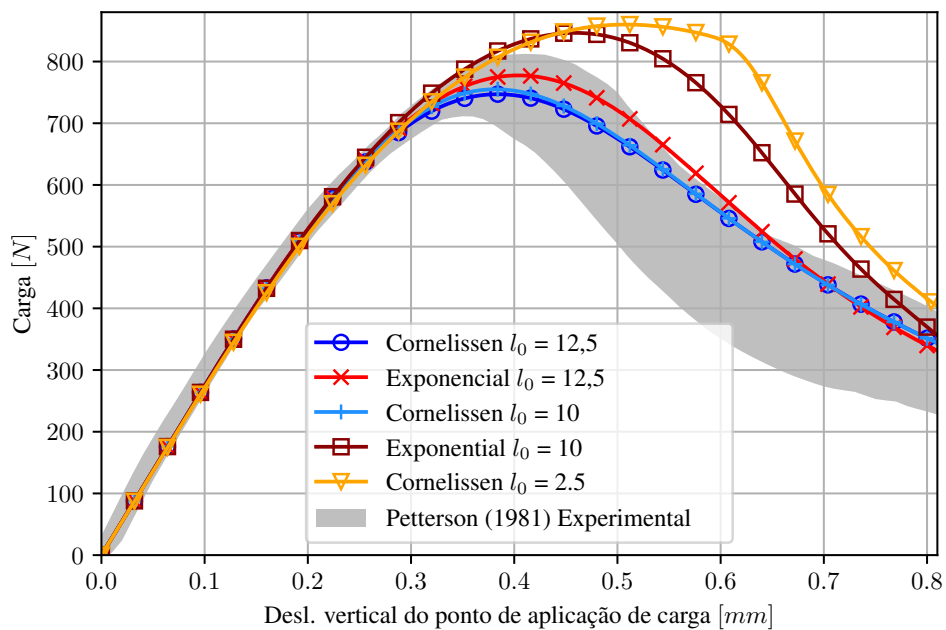


Fonte: Penna (2011) adaptado.

Figura 4.9 – Campo de fase para lei (a) de Cornelissen; (b) exponencial.



Fonte: O autor.

Figura 4.10 – Comparação de resultados para diferentes valores de l_0 

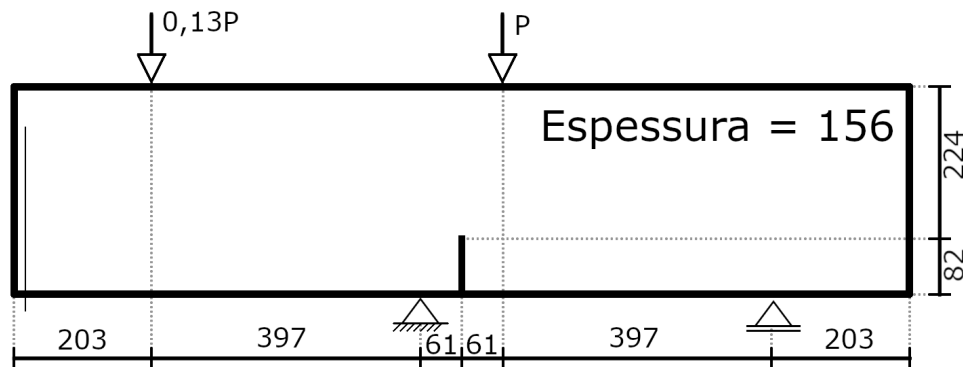
Fonte: O autor.

4.2.2 Cisalhamento em 4 Pontos

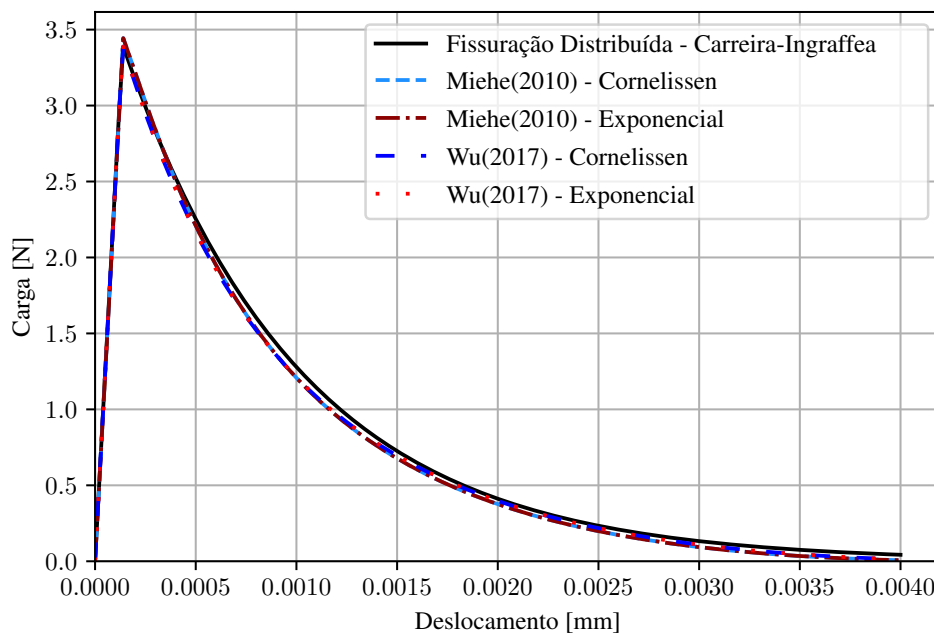
O modelo a seguir segue a montagem experimental realizada por Arrea e Ingraffea (1982), conforme a Figura 4.11. O experimento visa analisar a propagação da trinca numa viga sob esforço de cisalhamento e realiza o acompanhamento da degradação da força para o valor crescente de CMSD (*Crack Mouth Sliding Displacement*), o qual corresponde a distância vertical entre as duas bordas da trinca.

Apesar do trabalho experimental apresentar uma faixa de energia de fratura e resistência a

Figura 4.11 – Cisalhamento 4 pontos: montagem da análise.



Fonte: Penna (2011) adaptado.

Figura 4.12 – Cisalhamento 4 pontos: calibração do valor de l_0 .

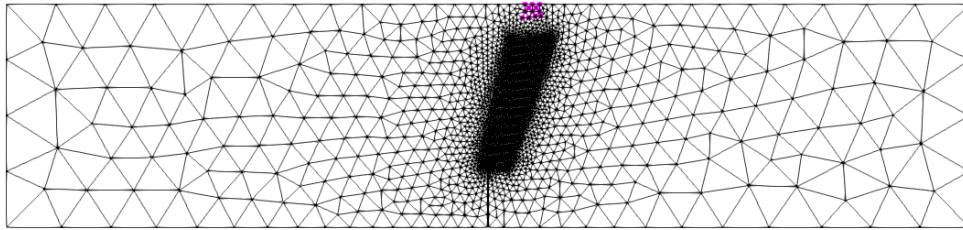
Fonte: O autor.

tração, o processo de calibração foi realizado utilizando das mesmas propriedades físicas utilizadas no trabalho de Penna (2011), especificamente para o modelo de campo de fase: $E_0 = 24800 \text{ N/mm}^2$, $\nu = 0,18$, $f_t = 3,4 \text{ N/mm}^2$, $G_f = 0,12 \text{ N/mm}^2$. Foi calibrado a resposta para os modelos constitutivos de Miehe et al. (2010) e de Wu (2017), com leis de amolecimento exponencial, $p = 2,5$ e de Cornelissen, $\eta_1 = 3,0$ e $\eta_2 = 6,93$. Determinou-se para a análise o valor de $l_0 = 12,5 \text{ mm}$ que gerou uma resposta material similar para os modelos em comparação à referência, como mostrado na Figura 4.12.

Para essa análise, foi utilizada uma malha, Figura 4.13, com região refinada de $h = 2,5 \text{ mm}$, a qual teve os valores de ϕ restringidos próximo aos pontos de aplicação de carga. Essa restrição se deve a concentração de carga, que em tentativas iniciais levaram a instabilidades numéricas. Apesar disso, essa região não interagiu com o campo de fase dentro da faixa apresentada

nas curvas de resposta. Para a convergência, foi usado controle direto de deslocamentos do nó

Figura 4.13 – Cisalhamento 4 pontos: malha utilizada, nós em rosa com restrição $\phi = 0$.



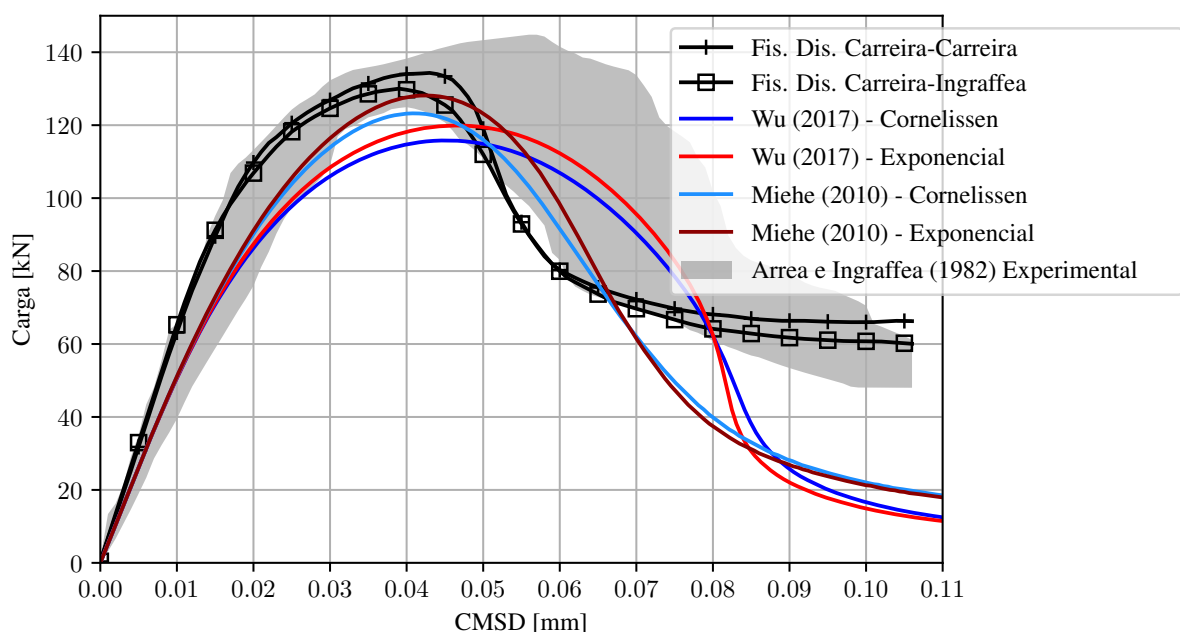
Fonte: O autor.

à direita da trinca, com passos verticais de 0,001 mm, tolerância global de $1 \cdot 10^{-4}$ e local de $1 \cdot 10^{-5}$. A Figura 4.14 apresenta os resultados encontrados.

Ambos os modelos e leis de amolecimento não conseguiram descrever com precisão a mudança de inclinação no ramo descendente da curva de resposta, essa mudança ocorreu, mas foi tardia. Apesar disso, o modelo constitutivo de Miehe et al. (2010) ainda conseguiu descrever o limite de carga aplicado dentro dos valores experimentais, o mesmo não ocorreu para as análises com o modelo híbrido.

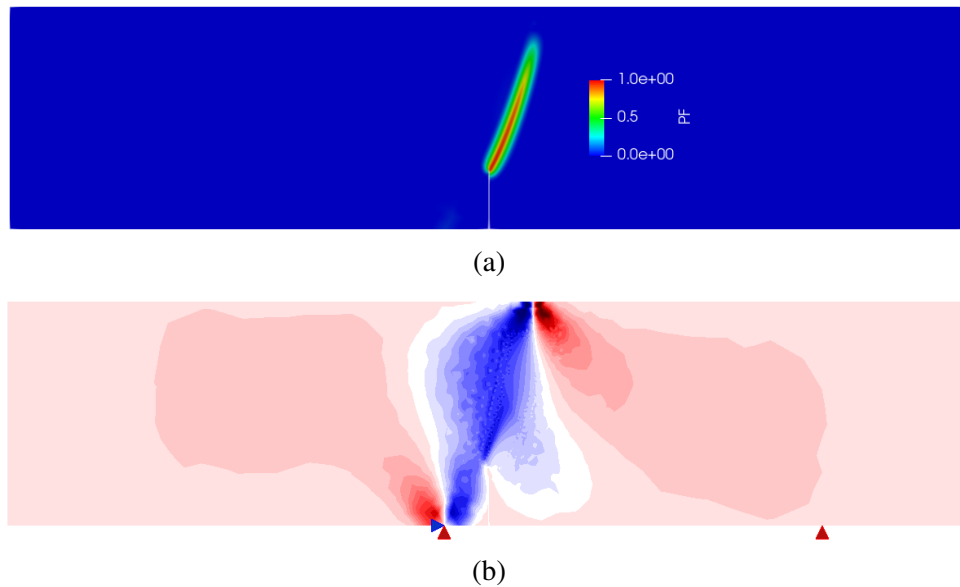
Todos os resultados estudados apresentaram um caminho para a trinca coerente com o observado experimentalmente, uma amostra distribuição do campo de fase na estrutura pode ser observado na Figura 4.15a. Ademais, a distribuição das tensões de cisalhamento no corpo formaram padrões semelhantes com os apresentados por Penna (2011), com uma banda de cisalhamento característica entre os pontos de aplicação de carga e o apoio conjugado. A Figura 4.15b ilustra a distribuição para a situação maior carga atingida na análise com modelo de Wu (2017)

Figura 4.14 – Resultados Cisalhamento 4 pontos



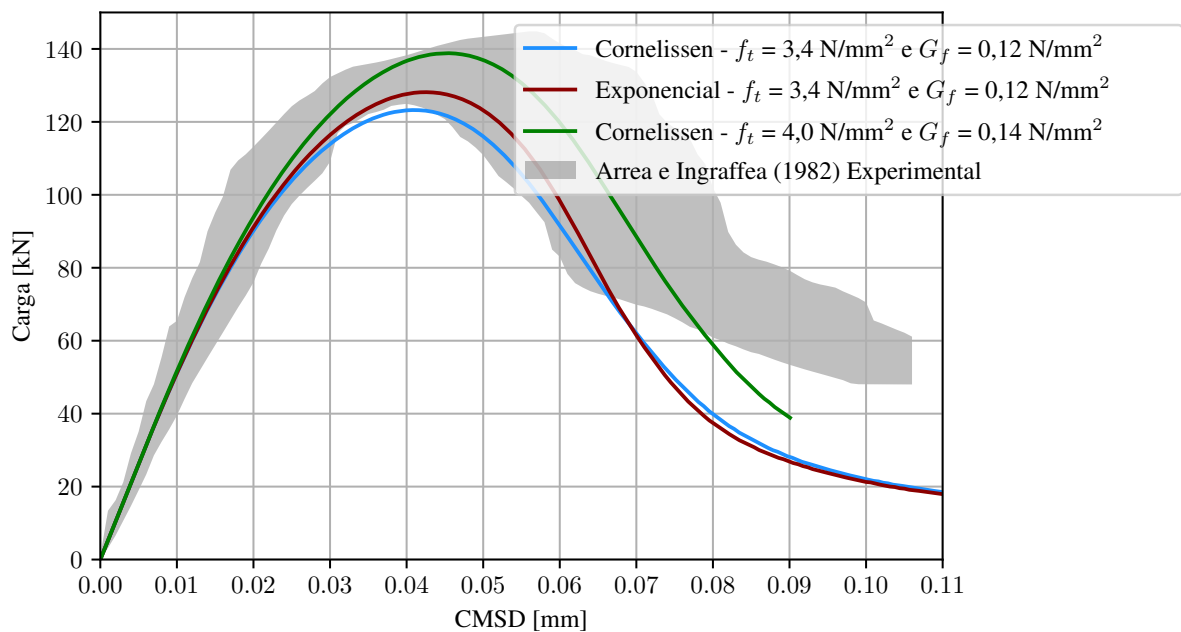
Fonte: Penna (2011) adaptado.

Figura 4.15 – Cisalhamento 4 Pontos: (a) distribuição da variável de campo; (b) tensões de cisalhamento na estrutura.



Fonte: O autor.

Figura 4.16 – Cisalhamento 4 pontos com outras propriedades



Fonte: O autor.

e lei de amolecimento de Cornelissen.

No intuito de explorar melhor as capacidades do modelo de campo de fase, realizou-se uma nova análise com a utilização da faixa superior de propriedades mecânicas determinadas experimentalmente com o modelo de Miehe et al. (2010). Dessa forma a Figura 4.16 apresenta a curva de resposta da nova análise em comparação com as anteriores.

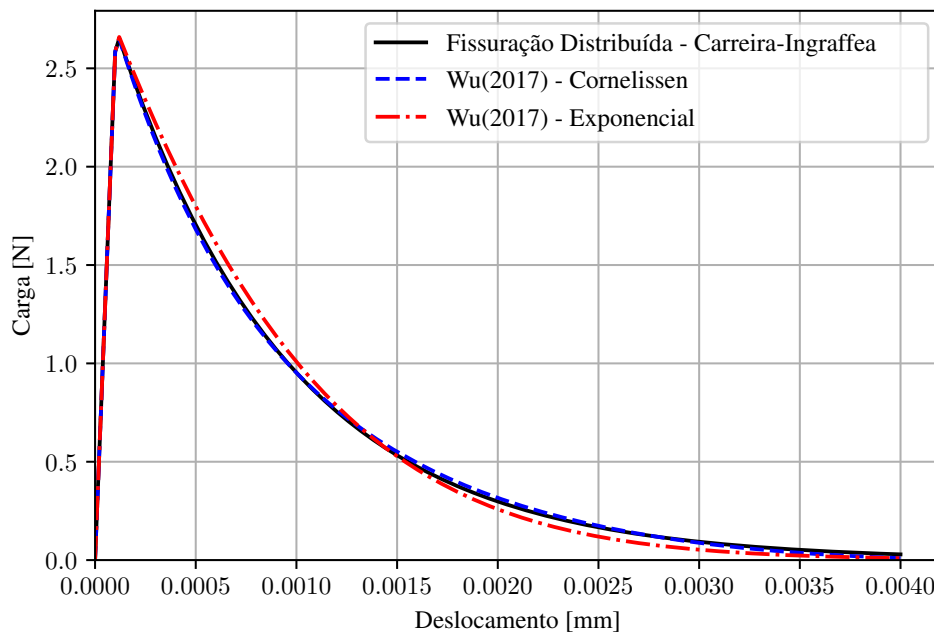
Percebe-se que com os novos parâmetros o modelo também manteve-se dentro da faixa de des-

locamentos experimentais em grande parte da análise, entretanto ainda não conseguiu descrever a mudança de direção no ramo descendente da curva com precisão. Essa queda brusca do fator de carga pode ser visualizado como um crescimento acelerado do campo de fase na simulação. Apesar de não explorado nesse trabalho, a utilização de outros modelos constitutivos ou outras leis de amolecimento poderiam influenciar na qualidade da resposta.

4.2.3 PaineL em L

O PaineL em L foi abordado na Seção 4.1.2 com o propósito de validação da implementação. Nessa seção novas análises serão apresentadas no intuito de comparar resultados com modelos de fissuração distribuída. Nesse sentido, serão utilizados os mesmos valores para as propriedades mecânicas utilizados no trabalho de Penna (2011): $E_0 = 25850$ MPa, $f_t = 2,7$ MPa, $f_c = 31$ N/m², $G_f = 0,065$ N/mm² e $\nu = 0,18$. Para essa análise optou-se pela utilização do modelo constitutivo de Wu (2017) com lei de amolecimento exponencial, com $p = 2.5$, e de Cornelissen, com $\eta_1 = 3,0$ e $\eta_2 = 6,93$. Como nos exemplos anteriores, o modelo foi calibrado comparativamente ao material de Carreira-Ingraffea, para apresentarem uma mesma resposta material. Dessa forma, o valor de $l_0 = 8,5$ mm foi escolhido para as análises, veja a Figura 4.17. Apesar da montagem experimental ter sido realizada com a aplicação de uma carga pontual a

Figura 4.17 – Cisalhamento 4 pontos com outras propriedades

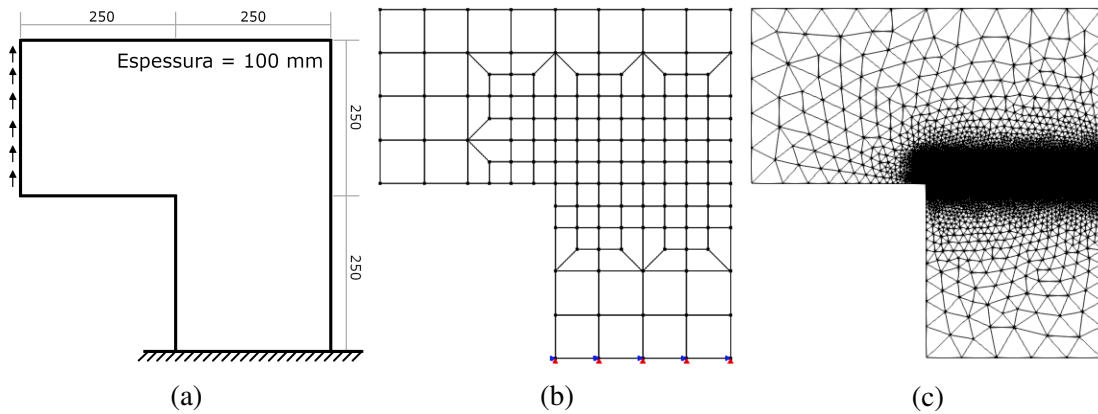


Fonte: Penna (2011) adaptado.

uma distância de 30 mm da borda do paineL, no intuito de comparar os modelos em situações similares, foi realizada a montagem de acordo com a apresentada no trabalho de referência, uma carga linear aplicada em toda extremidade esquerda. A montagem esquemática usada na análise pode ser vista na Figura 4.18a, junto a malha do trabalho de Penna (2011), Figura 4.18b,

e a malha usada aqui para o modelo de campo de fase, Figura 4.18c, com tamanho médio dos elementos de $h = 1$ mm na região refinada.

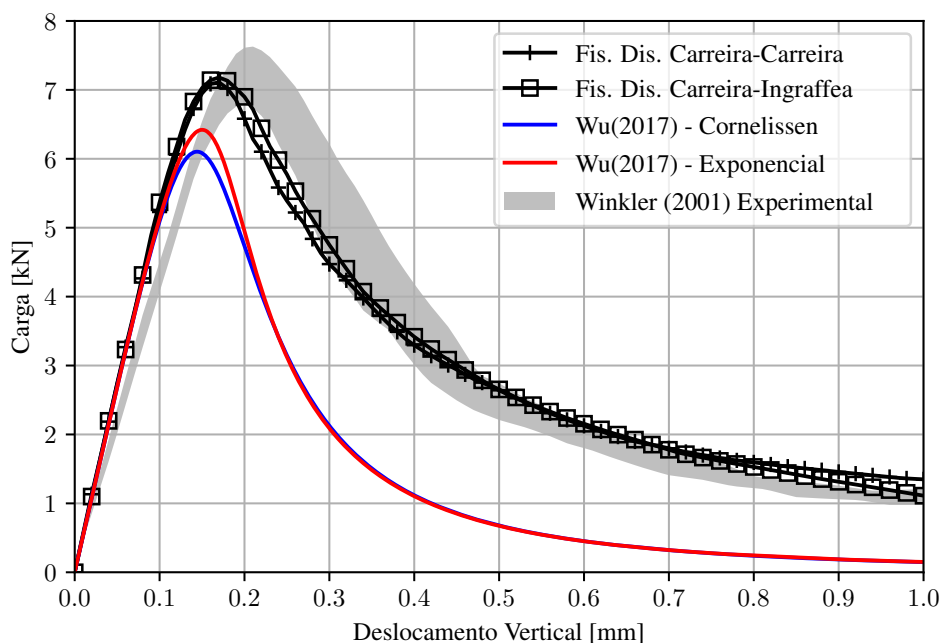
Figura 4.18 – Painel em L de Penna (2011): (a) esquema de montagem; (b) malha para fissuração distribuída; (c) malha para campo de fase.



Fonte: (a) e (c) o autor e (b) Penna (2011) adaptado.

Para convergência, foi usado o controle direto de deslocamentos com passos de 0,002 mm no eixo y para o nó superior esquerdo, tolerância local de $1 \cdot 10^{-5}$ e global de $1 \cdot 10^{-4}$. Os resultados podem ser encontrados na Figura 4.19.

Figura 4.19 – Resultados para análise do Painel em L



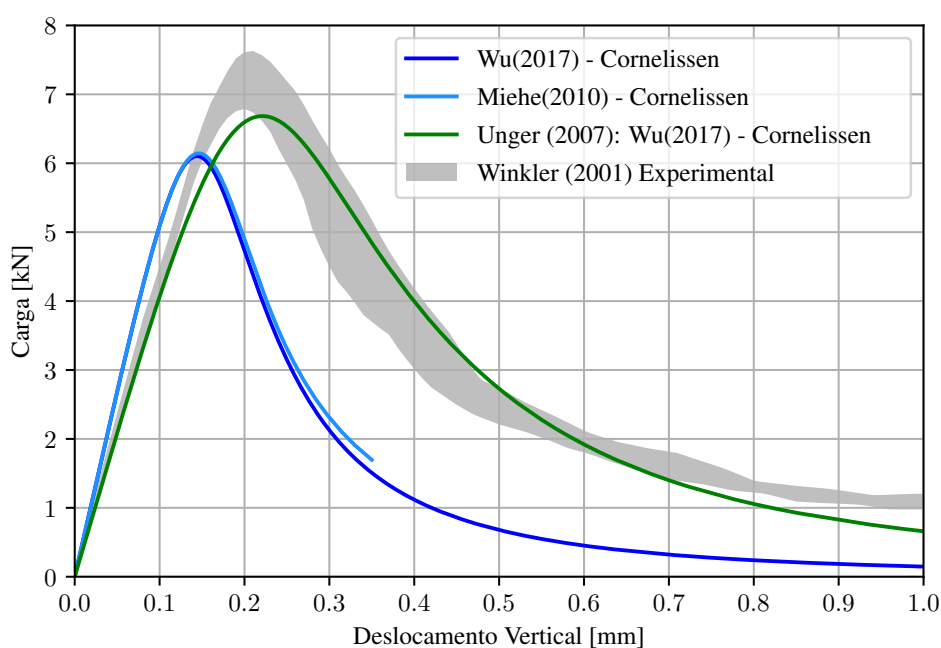
Fonte: O autor.

Observa-se que ambas as análises não obtiveram bons resultados. Todas falharam em alcançar o pico de carga aplicada e sobrestimaram a degradação. Além disso, devido a necessidade de alto refinamento da malha e a necessidade de realização de várias iterações para alcançar a convergência, a solução de cada uma das análises com o modelo constitutivo de Wu (2017)

demorou por volta de 35 horas com os recursos computacionais usados para esse trabalho. Isso mostrou-se uma barreira a uma possível tentativa de buscar outras otimizações, como com a variação parâmetro de ξ ou explorar mais funções de degradação energética. Esse problema é ainda mais acentuado para o modelo de Miehe et al. (2010) que, devido à severa não-linearidade, a convergência pode demorar um tempo várias vezes maior para ser atingida.

Foram realizadas tentativas de utilização do modelo constitutivo de Miehe et al. (2010), entretanto, em análises com carga distribuída o processo ultrapassava o limite de 500 iterações imposto às iterações locais. A Figura 4.20 mostra os resultados encontrados numa tentativa que percorreu 175 passos em cerca de 40 horas de processamento antes de ultrapassar o limite de iterações e interromper a análise. É possível observar que mesmo sendo um modelo constitutivo mais complexo o resultado apresenta pouca variação, com o pico aproximadamente igual e apenas uma pequena melhora na estimativa da degradação.

Figura 4.20 – Outros resultados para análise do Painel em L



Fonte: O autor.

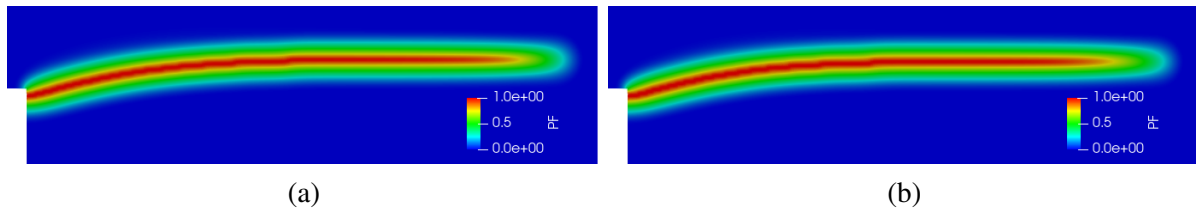
Isso ressalta a importância no desenvolvimento de modelos híbridos, pois, em muitos casos conseguem resultados similares e reduzem o custo computacional de forma significativa. No caso apresentado, o uso do modelo híbrido pôde, com os mesmos recursos computacionais, realizar os 500 passos propostos em menos tempo que o modelo com separação percorreu apenas 175.

Ademais, numa tentativa de alcançar resultados mais próximos do experimental, encontra-se na literatura o trabalho de Unger et al. (2007) que aponta ter alcançado melhores resultados para o ramo decrescente em suas análises em XFEM ao mudar a elasticidade e a energia da fratura do modelo para alcançar mesmo pico de carga. Wu et al. (2020) apresenta resultados para essas mesmas propriedades para FEM. Seguindo esse raciocínio, foi feita uma análise usando uma

montagem similar a esses autores, com $E_0 = 20000$ MPa, $G_f = 0,13$ N/mm², $l_0 = 5$ mm e carga pontual no nó posicionado na coordenada (0,250). Os resultados podem ser encontrados também na Figura 4.20. A nova curva é compatível com os dados experimentais, mas a necessidade de adaptação das propriedades físicas é uma questão debatível.

Por fim, todos os modelos apresentaram um caminho da trinca similar e correspondente à encontrada nos resultados experimentais de Winkler et al. (2004). Destaca-se que por permitir o uso de malhas refinadas, apesar de demandar um alto custo computacional, o modelo de campo de fase permite uma melhor visualização do caminho da trinca, quando comparado aos modelos de fissuração distribuída, em que os elementos são significativamente maiores.

Figura 4.21 – Destaque da distribuição da variável ϕ no Painel em L: (a) lei de amolecimento de Cornelissen; (b) Lei de amolecimento Exponencial.



Fonte: O autor.

5 CONCLUSÃO

O objetivo desse trabalho era implementar novas aplicações no sistema INSANE que viabilizassem o estudo mais aprofundado de modelos de campo de fase. Como mostrado ao longo do desenvolvimento, foram implementados recursos para importação de malhas e montagem de modelo, novos modelos constitutivos, funções com base nas propriedades mecânicas dos materiais e um novo algoritmo de solução mais robusto que permitisse a utilização de todas essas novas ferramentas. Com essas funcionalidades foi possível visualizar comparativamente como os modelos de campo de fase se comportavam em simulações similares com modelos de degradação elástica.

Em relação as implementações de pré-processamento, foi mostrado, através das várias análises realizadas, que a aplicação é bem estruturada e fornece ferramentas para a montagem de modelos suficientemente complexos para análises bidimensionais com modelo de campo de fase. Além disso, por atingir resultados similares aos da literatura, percebe-se que tanto o algoritmos de solução, quanto as funções novas se portam de maneira esperada para suas aplicações. Assim, o sistema INSANE incorpora agora novas ferramentas que possibilitam estudos mais sofisticados nas áreas de campo de fase.

Visando as comparações de resultados, foi possível perceber que os métodos com campo de fase com materiais parametrizados aos de degradação elástica não atingiram curvas com mesma qualidade de resultados para os exemplos de painel em L e cisalhamento em 4 pontos. Apesar disso, ressalta-se que a possibilidade de trabalhar com malhas refinadas torna possível compreender melhor o desenvolvimento de trincas no material, por permitir acompanhar seu desenvolvimento ao longo da malha. A qualidade da resposta ainda tem potencial para ser melhorada com o uso de parâmetros mais otimizados, como a variação das funções utilizadas e os parâmetros de cada. Entretanto, visto a quantidade de variáveis necessárias para definir completamente o modelo, existe a necessidade de estabelecer-se critérios para a determinação valores ideais para cada condição e material de análise.

Os exemplos de flexão de 3 pontos e painel em L mostraram que os modelos híbridos tem potencial de melhorar o desempenho das análises com pouca influência na qualidade da resposta. Nos exemplos trabalhados nesse texto isso parece ter ocorrido em vista a predominância de tração em algumas análises. Novos modelos híbridos podem desenvolver estratégias mais sofisticadas que melhorem a resposta para condições mais complexas de solicitação.

Ademais, percebe-se uma necessidade de relacionar as propriedades físicas do material com os parâmetros essencialmente do modelo de campo de fase. O estudo dessas relações é essencial para tornar o modelo de campo de fase menos um método simplesmente numérico e mais uma ferramenta de estudo de engenharia, para permitir simulações mais confiáveis com menor dependência de resultados experimentais para validar o modelo. A utilização da análise de Wu (2017) para modelos unidimensionais permitiu relacionar a resistência a tração com variáveis

do modelo, mas ainda necessita-se determinar uma forma de parametrizar melhor o comprimento característico para as análises.

Por fim, sugere-se que se incorpore ferramentas que tornem os algoritmos de solução mais eficientes no INSANE. No momento, as simulações apresentadas nesse trabalho demandavam mais de um dia, para modelos como o painel em L, o que se torna uma barreira para a realização de estudos no programa. Algumas soluções já existem na bibliografia, como o uso de métodos de sobre-relaxamento, bibliotecas otimizadas, entre outros.

Dito isso, seguem sugestões para trabalhos futuros:

- Implementação de novas ferramentas para aumento de eficiência na solução de modelos de campo de fase.
- Estudo aprofundado da escolha de funções de degradação energética e suas relações com outros parâmetros do modelo.
- Estudo sobre o relacionamento do comprimento característico da trinca com propriedades dos materiais.
- Implementar e estudar novos modelos constitutivos híbridos.

REFERÊNCIAS

- AMOR, H.; MARIGO, J.-J.; MAURINI, C. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. *Journal of the Mechanics and Physics of Solids*, v. 57, n. 8, p. 1209–1229, ago. 2009. 29
- ARREA, M.; INGRAFFEA, A. *Mixed mode crack propagation in mortar and concrete*. [S.l.], 1982. 54
- BĂZANT, Z. P.; OH, B. H. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. *Matériaux et Constructions*, v. 16, n. 3, p. 155–177, mai. 1983. 16
- BENSON, S. J.; MUNSON, T. S. Flexible complementarity solvers for large-scale applications. *Optimization Methods and Software*, v. 21, n. 1, p. 155–168, jul. 2006. 28, 45
- BLEYER, J.; ALESSI, R. Phase-field modeling of anisotropic brittle fracture including several damage mechanisms. *Computer Methods in Applied Mechanics and Engineering*, v. 336, p. 213–236, fev. 2018. 32
- BOONE, T.; WAWRZYNEK, P.; INGRAFFEA, A. Simulation of the fracture process in rock with application to hydrofracturing. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, v. 23, n. 3, p. 255–265, 1986. 16
- BOURDIN, B.; FRANCFORT, G.; MARIGO, J. J. Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids*, v. 48, n. 4, p. 797–826, mar. 2000. 18, 21, 27, 29, 46
- CARREIRA, D. J.; CHU, K.-H. Stress-strain relationship for plain concrete in compression. *American Concrete Institute Journal*, v. 82, n. 6, p. 797–804, jan. 1985. 16, 32
- de BORST, R.; GUTIÉRREZ, M. A. A unified framework for concrete damage and fracture models including size effects. *International Journal of Fracture*, v. 95, p. 261–227, 1999. 15
- de VREE, J. H.; BREKELMANS, W. A. M.; GILS, M. A. J. v. Comparison of nonlocal approaches in continuum damage mechanics. *Computers & Structures*, v. 55, n. 4, p. 581–588, mai. 1995. 15
- DUGDALE, D. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids*, v. 8, n. 2, p. 100–104, mai. 1960. 12
- FARREL, P.; MAURINI, C. Linear and nonlinear solvers for variational phase-field models of brittle fracture. *International Journal for Numerical Methods in Engineering*, v. 109, p. 648–667, jul. 2017. 28
- GEUZAIN, C.; REMACLE, J.-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, v. 79, n. 11, p. 1309–1331, 2009. 33
- GMSH 4.8.4. *Gmsh 4.8.4 Documentation*. [S.l.], 2021. Disponível em: <https://gmsh.info/doc/texinfo/gmsh.html#MSH-file-format>. 33, 34

- GRIFFITH, A. A. The phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London*, v. 221, n. 1, p. 163–198, fev. 1921. 12
- JIRÁSEK, M. Damage and smeared crack models. In: _____. *Numerical Modeling of Concrete Cracking*. Vienna: Springer Vienna, 2011. p. 1–49. 16
- JU, J. W. On energy-based coupled elastoplasticity damage theories: constitutive modeling and computational aspects. *International Journal of Solids and Structures*, v. 25, n. 7, p. 803–833, dez. 1989. 15
- KACHANOV, L. M. Time of the rupture process under creep conditions. *Izvestiia Akademii Nauk SSSR, Otdelenie Tekhnicheskikh Nauk*, v. 8, n. 1, p. 26–31, 1958. 12
- KARMA, A.; KESSLER, D.; LEVINE, H. Phase-field model of mode iii dynamic fracture. *Physical Review Letters*, v. 87, mai. 2001. 21
- KUHN, C.; A., S.; MÜLLER, R. On degradation functions in phase field fracture models. *Computational Materials Science*, v. 108, p. 374–384, 2015. 21
- LANCIONI, G.; ROYER-CARFAGNI, G. The variational approach to fracture mechanics. *International Journal of Solids and Structures*, v. 25, n. 7, p. 803–833, dez. 1989. 29
- LEMAITRE, J.; CHABOCHE, J. L. *Mechanics of Solid Materials*. Cambridge: Cambridge University Press, 1990. 15
- LEÃO, H. M. *Implementation in INSANE of the Phase-field Modelling with FEM*. Dissertação (Mestrado) — Pós-graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, MG (Brasil), 2021. 21, 24, 27, 30, 32, 43, 44, 48
- LOGAN, D. L. *A First Course in the Finite Element Method*. [S.l.]: Cengage Learning, 2015. 12
- MAZARS, J. *Application de la mécanique de l'endommagement au comportement non lineaire et à la rupture du béton de structure*. Tese (Doutorado) — Université Paris, Paris (França), 1984. 15
- MAZARS, J.; LEMAITRE, J. Application of continuous damage mechanics to strain and fracture behavior of concrete. In: _____. *Application of Fracture Mechanics to Cementitious Composites*. Dordrecht: Springer Netherlands, 1985. p. 507–520. 15
- MIEHE, C.; WELSCHINGER, F.; HOFACKER, M. Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field fe implementations. *International Journal for Numerical Methods in Engineering*, v. 83, n. 10, p. 1273–1311, set. 2010. 13, 27, 29, 30, 32, 51, 52, 55, 56, 57, 60
- PENNA, S. S. *Pós-Processador para Modelos Bidimensionais Não-Lineares do Método dos Elementos Finitos*. Dissertação (Mestrado) — Pós-graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, MG (Brasil), 2007. 37
- PENNA, S. S. *Formulação Multipotencial para Modelos de Degradação Elástica: Unificação Teórica, Proposta de Novo Modelo, Implementação Computacional e Modelagem de Estruturas de Concreto*. Tese (Doutorado) — Pós-graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, MG (Brasil), 2011. 6, 7, 15, 16, 17, 51, 52, 53, 55, 56, 58, 59

- PETERSSON, P. Crack growth and development of fracture zones in plain concrete and similar materials. *Report TVBM*, v. 1006, dez. 1981. 51
- PHAM, K. et al. Gradient damage models and their use to approximate brittle fracture. *International Journal of Damage Mechanics*, v. 20, n. 4, p. 618–652, 2011. 18
- RASHID, Y. R. Ultimate strength analysis of prestressed concrete pressure vessels. *Nuclear Engineering and Design*, v. 7, n. 4, p. 334–344, abr. 1968. 16
- ROTS, J. G. et al. Smeared crack approach and fracture localization in concrete. *HERON*, v. 30, n. 1, p. 1–48, jan. 1985. 16
- SIMO, J. C.; JU, J. W. Strain- and stress-based continuum damage models i. – formulation. *International Journal of Solids and Structures*, v. 23, n. 7, p. 821–840, dez. 1987. 15
- SUIDAN, M.; SCHNOBRICH, W. C. Finite element analysis of reinforced concrete. *Journal of the structural division*, v. 99, n. 10, p. 2109–2121, 1973. 16
- UNGER, J.; ECKARDT, S.; KÖNKE, C. Modelling of cohesive crack growth in concrete structures with the extended finite element method. *Computer Methods in Applied Mechanics and Engineering*, v. 196, p. 4087–4100, set. 2007. 60
- WINKLER, B.; HOFSTETTER, G.; LEHAR, H. Application of a constitutive model for concrete to the analysis of a precast segmental tunnel lining. *International Journal for Numerical and Analytical Methods in Geomechanics*, v. 28, p. 797–819, 2004. 49, 61
- WU, J. Y. A unified phase-field theory for the mechanics of damage and quasi-brittle failure. *Journal of the Mechanics and Physics of Solids*, v. 103, p. 72–99, mar. 2017. 6, 18, 19, 21, 22, 23, 31, 32, 42, 43, 46, 47, 48, 49, 50, 51, 52, 55, 56, 58, 59, 62
- WU, J. Y. et al. Phase-field modelling of fracture. *Advances in applied mechanics: multi-scale theory and computation*, v. 52, n. 1, p. 1–183, jan. 2020. 12, 19, 20, 21, 27, 28, 30, 45, 60
- YANG, H.; YANG, C.; SUN, S. Active-set reduced-space methods with nonlinear elimination for two-phase flow problems in porous media. *SIAM Journal on Scientific Computing*, v. 38, n. 4, p. B593–B618, jul 2016. 45