

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia de Estruturas

Túlio Roberto Eládio Marques

**IMPLEMENTAÇÃO NÃO INTRUSIVA DO MÉTODO DOS
ELEMENTOS FINITOS GENERALIZADOS COM
ENRIQUECIMENTO GLOBAL-LOCAL PARA ANÁLISES
MULTI-DOMÍNIO**

Belo Horizonte
2025

Túlio Roberto Eládio Marques

**IMPLEMENTAÇÃO NÃO INTRUSIVA DO MÉTODO DOS ELEMENTOS
FINITOS GENERALIZADOS COM ENRIQUECIMENTO
GLOBAL-LOCAL PARA ANÁLISES MULTI-DOMÍNIO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Estruturas da Escola de Engenharia da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Estruturas.

Orientador: Prof. Dr. Felício Bruzzi Barros

Coorientador: Prof. Dr. Ramon Pereira da Silva

Belo Horizonte
2025

M357i

Marques, Túlio Roberto Eládio.

Implementação não intrusiva do método dos elementos finitos generalizados com enriquecimento global-local para análises multi-domínio [recurso eletrônico] / Túlio Roberto Eládio Marques. - 2025.

1 recurso online (154 f. : il., color.) : pdf.

Orientador: Felício Bruzzi Barros.

Coorientador: Ramon Pereira da Silva.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Inclui bibliografia.

1. Engenharia de estruturas - Teses. 2. Método dos elementos finitos - Teses. 3. Acoplamentos - Teses. 4. Mecânica da fratura - Teses. 5. Computação paralela - Teses. I. Barros, Felício Bruzzi. II. Silva, Ramon Pereira da. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 624(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS



PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ESTRUTURAS



ATA DA DEFESA DE DISSERTAÇÃO DE MESTRADO EM ENGENHARIA DE ESTRUTURAS Nº: 411 DO ALUNO TÚLIO ROBERTO ELÁDIO MARQUES

Às **08:30** horas do dia **19** do mês de **fevereiro** de **2025**, reuniu-se em ambiente virtual, na Escola de Engenharia da Universidade Federal de Minas Gerais - UFMG, a Comissão Examinadora indicada pelo Colegiado do Programa em **27 de janeiro de 2025**, para julgar a defesa da Dissertação de Mestrado intitulada "**Implementação não Intrusiva do Método dos Elementos Finitos Generalizados com Enriquecimento Global-Local para Análises Multi-Domínio**", cuja aprovação é um dos requisitos para a obtenção do Grau de MESTRE EM ENGENHARIA DE ESTRUTURAS na área de ESTRUTURAS.

Abrindo a Sessão, o Presidente da Comissão, Prof. Dr. Felício Bruzzi Barros, após dar conhecimento aos presentes do teor das Normas Regulamentares, passou a palavra ao aluno para apresentação de seu trabalho. Finalizada a apresentação, seguiu-se para a fase de arguição pelos examinadores, com as respectivas respostas do aluno. Logo após a fase de arguição, a Comissão se reuniu, sem a presença do aluno e do público, para julgamento e expedição do resultado final, a saber:

- Aprovado
 Reprovado

O resultado final foi comunicado publicamente ao aluno pelo Presidente da Comissão.

Nada mais havendo a tratar, o Presidente encerrou a reunião e lavrou a presente ata, que será assinada por todos os membros participantes da Comissão Examinadora e pelo aluno.

Comissão Examinadora:

Prof. Dr. Felício Bruzzi Barros - DEES - UFMG (Orientador)
Prof. Dr. Ramon Pereira da Silva - DEES - UFMG (Coorientador)
Prof. Dr. Jeferson Wilian Dossa Fernandes - DEES-UFMG
Prof. Dr. Marcos Arndt - UFPR
Profa. Dra. Gabriela Marinho Fonseca - UFES

A aprovação do aluno na Defesa da Dissertação de Mestrado não significa que o mesmo tenha cumprido todos os requisitos necessários para obtenção do DIPLOMA de Mestre em Engenharia de Estruturas.

Para ciência do aluno:

1. Atesto que as alterações solicitadas pela Comissão Examinadora serão cumpridas no prazo determinado pela banca de **ATÉ 60 dias** corridos.

2. Atesto estar ciente que a versão corrigida da dissertação deverá ser entregue ao Repositório Institucional da UFMG, conforme orientações disponíveis em: <https://repositorio.ufmg.br/custom/instructions.jsp>

Ciente: Túlio Roberto Eládio Marques
Aluno

Belo Horizonte, 19 de fevereiro de 2024.

Este documento não terá validade sem a assinatura do Coordenador do Programa de Pós-Graduação. Após a Homologação da Defesa pelo Colegiado do Programa, este documento será assinado pela Coordenação e disponibilizado ao aluno.



Documento assinado eletronicamente por **Felício Bruzzi Barros, Professor do Magistério Superior**, em 19/02/2025, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Gabriela Marinho Fonseca, Usuário Externo**, em 19/02/2025, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Jeferson Wilian Dossa Fernandes, Professor do Magistério Superior**, em 19/02/2025, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Ramon Pereira da Silva, Professor do Magistério Superior**, em 19/02/2025, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcos Arndt, Usuário Externo**, em 19/02/2025, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Túlio Roberto Eládio Marques, Usuário Externo**, em 19/02/2025, às 11:57, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3945824** e o código CRC **AAA60856**.

Aos meus pais, Matildes e Sérgio.

AGRADECIMENTOS

Agradeço a Deus pela oportunidade de concretizar este sonho e pela concessão da força e sabedoria necessárias para superar os desafios ao longo deste percurso. À Senhora de Nazareth, cuja presença materna constante foi um apoio nos momentos de incerteza e cansaço, e por sua intercessão, encontrei a coragem para perseverar.

Aos meus pais, Matildes e Sérgio, pelos ensinamentos e valores transmitidos e o constante apoio e incentivo no decorrer da minha trajetória acadêmica. Agradeço a presença de vocês ao meu lado nas alegrias e nos momentos difíceis. Vocês são minhas maiores inspirações.

Ao professor Felício Bruzzi Barros, que ao longo de todo o período de orientação foi um exemplo de ética, profissionalismo e generosidade, agradeço a dedicação em ensinar, solucionar dúvidas e assistência ofertada ao longo desta pesquisa. Estendo meus agradecimentos aos demais professores do Departamento de Engenharia de Estruturas da UFMG, em especial o professor Ramon Pereira da Silva, cujo auxílio em momentos decisivos foi de fundamental importância para o desenvolvimento deste mestrado.

Aos meus familiares e amigos, pelo apoio incondicional ao longo desta jornada. Em especial, ao amigo Felipe Hosken, cuja presença constante, partilha das angústias e conquistas, e disposição para ajudar foram fundamentais. Expresso minha gratidão pela amizade, companhia e significativa contribuição para esta pesquisa, sem as quais o êxito desta dissertação não teria sido possível.

Aos colegas de pesquisa do grupo *INSANE*, com destaque para Ana Clara Bueno, cuja disponibilidade e assistência na transmissão dos fundamentos inerentes à plataforma computacional foram de grande importância para o desenvolvimento deste trabalho.

A todos que contribuíram de alguma forma em minha trajetória acadêmica, meus agradecimentos.

RESUMO

Esta dissertação de mestrado expande a implementação não intrusiva do Método dos Elementos Finitos Generalizados com enriquecimento Global-Local (MEFG^{gl}-GLI) para análises multi-domínios. No MEFG^{gl}-GLI, inicialmente, o problema global é discretizado utilizando uma malha grosseira, sem a consideração dos fenômenos localizados, minorando a interferência do usuário na construção do modelo de análise. A solução deste domínio é obtida por meio da formulação convencional do Método dos Elementos Finitos, a qual é aplicada neste trabalho utilizando o *software* comercial *Abaqus*. Em seguida, são definidas mesoescalas, tantas quantas necessárias, que atuam como escalas intermediárias entre os domínios global e local. Os fenômenos de interesse, como trincas, são efetivamente representados na escala local. A associação entre cada mesoescala e seu respectivo problema local é realizada por meio do enriquecimento global-local do MEFG^{gl}. O acoplamento das mesoescalas ao problema global é estabelecido através da transferência de deslocamentos e forças generalizadas, caracterizando a estratégia não intrusiva denominada Global-Local Iterativa (GLI). As simulações numéricas utilizando o MEFG^{gl} são realizadas no sistema computacional *INSANE* (*IN*teractive *S*tructural *A*Nalysis *E*nvironment - www.insane.dees.ufmg.br), um projeto de *software* livre desenvolvido no Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais. A capacidade de realizar análises com múltiplos domínios locais e a combinação de *solvers*, visa fomentar o emprego de algoritmos desenvolvidos no ambiente acadêmico como ferramentas que aprimoram a resolução de modelos complexos por meio de *softwares* comerciais. Exemplos numéricos demonstram o desempenho da implementação e investigam a influência dos principais parâmetros relacionados à estratégia de solução proposta. A qualidade dos resultados é avaliada com base no estado de tensões e deformações em pontos críticos dos modelos, além de parâmetros da Mecânica da Fratura Linear Elástica. As investigações indicam que o processo de convergência não é afetado pela consideração de múltiplos domínios locais, desde que estes não impliquem em um aumento significativo na diferença de rigidez entre as escalas acopladas. Adicionalmente, foi avaliado o desempenho computacional do algoritmo, comparando simulações nas quais o processamento das mesoescalas é realizado de forma sequencial e em paralelo.

Palavras-chave: Método dos Elementos Finitos Generalizados; análise global-local; acoplamento não intrusivo; Mecânica da Fratura Linear Elástica; computação paralela.

ABSTRACT

This master's thesis expands the non-intrusive implementation of the Generalized Finite Element Method with Global-Local enrichment (IGL-GFEM^{gl}) for multi-domain analyses. In IGL-GFEM^{gl}, the global problem is initially discretized using a coarse mesh, without considering localized phenomena, thereby minimizing user interference in the analysis model construction. The solution for this domain is obtained through the standard formulation of the Finite Element Method, which is applied in this work using *Abaqus* commercial software. Subsequently, as many mesoscales as necessary are defined, acting as intermediate scales between the global and local domains. The phenomena of interest, such as cracks, are effectively represented in the local scale. The association between each mesoscale and its respective local problem is carried out through the global-local enrichment of GFEM^{gl}. The mesoscales coupling to the global problem is established through the transfer of displacements and generalized forces, characterizing the non-intrusive strategy known as Iterative Global-Local (IGL). Numerical simulations using GFEM^{gl} are performed in the computational system *INSANE* (*Interactive Structural ANalysis Environment* - www.insane.dees.ufmg.br), a free software project developed at the Department of Structural Engineering at the Federal University of Minas Gerais. The ability to perform analyses with multiple local domains and the combination of solvers, aims to use algorithms developed in the academic environment to improve complex models solving through commercial software. Numerical examples are presented to demonstrate simulation's performance and to investigate the influence of the main parameters related to the proposed strategy. The results quality are evaluated based on the state of stresses and strains at critical points of the models, in addition to parameters of Linear Elastic Fracture Mechanics. The investigations indicate that the convergence process is not affected by the consideration of multiple local domains, as long as these do not result in a significant increase in the stiffness difference between the coupled scales. Additionally, the computational performance of the algorithm was evaluated, comparing simulations which the mesoscales processing were executed sequentially and in parallel.

Key-words: Generalized Finite Element Method; global-local analysis; non-intrusive coupling; Linear Elastic Fracture Mechanics; parallel computing.

LISTA DE FIGURAS

2.1	Representação esquemática do processo de solução via técnica <i>zooming</i> . . .	31
2.2	Domínios do pórtico em “L” analisado via MEF Global-Local Iterativo (MEF-GLI).	32
2.3	Representação esquemática do processo de solução via Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI).	34
2.4	Estratégia de Enriquecimento da nuvem ω_j - (Barros, 2002).	38
2.5	Representação esquemática do processo de enriquecimento Global-Local. .	43
2.6	Ilustração do ciclo iterativo Global-Local.	45
2.7	Representação esquemática da zona <i>buffer</i>	46
2.8	Sistema de coordenadas polares relativas à ponta da trinca	48
2.9	Representação esquemática da modelagem de trincas através da estratégia de enriquecimento do MEF ^{gl} . A representação da trinca (linha vermelha) é apenas ilustrativa, tal como descrito na estratégia de enriquecimento do MEF ^{gl} indicada na presente seção.	48
2.10	Representação esquemática da estratégia MEF ^{gl} -GLI.	51
2.11	Fluxograma dos principais passos da estratégia MEF ^{gl} -GLI segundo abordagem monolítica e <i>staggered</i> . Adaptado de (Bueno, 2024)	53
2.12	Fluxograma dos Estados do Ciclo de vida de <i>threads</i> em JAVA	57
3.1	Modificações da classe <code>PersistenceAsXml</code>	75
3.2	Variáveis e métodos da Classe <code>ConsoleManager</code>	76
3.3	Fluxograma de solução da abordagem monolítica	78
3.4	Fluxograma de solução da abordagem <i>Staggered</i>	79
3.5	Variáveis e métodos da Classe <code>IGLGlobalLocalCrackPropagationSimulationManager</code>	80
3.6	Variáveis e métodos da Classe <code>IGLGlobalLocal</code>	83

4.1	Pórtico com trincas em modo misto de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos da ligação viga-pilar à direita podem ser obtidos por simetria.	88
4.2	Evolução da força residual relativa ao longo dos ciclos para a abordagem monolítica.	89
4.3	Evolução da força residual relativa ao longo dos ciclos para a abordagem <i>staggered</i>	89
4.4	Evolução da erro dos fatores de intensidade de tensão para a abordagem monolítica.	90
4.5	Evolução da erro dos fatores de intensidade de tensão para a abordagem <i>staggered</i>	90
4.6	Componente de deslocamento vertical, u_y , do modelo global.	93
4.7	Componente de deslocamento vertical, u_y , do modelo local.	93
4.8	Componente de deslocamento vertical, u_y , na aresta superior do pórtico ($y = 300$).	94
4.9	Chapa tracionada com trincas de borda em modo I de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos das escalas meso e local associados à trinca da direita, podem ser obtidos por simetria.	95
4.10	Evolução da força residual relativa ao longo dos ciclos para a abordagem monolítica.	96
4.11	Evolução da força residual relativa ao longo dos ciclos para a abordagem <i>staggered</i>	97
4.12	Evolução do erro de K_I para a abordagem monolítica.	98
4.13	Evolução do erro de K_I para a abordagem para a abordagem <i>staggered</i>	98
4.14	Modelos locais gerados ao longo dos passos de propagação da trinca.	101
4.15	Deslocamento horizontal, u_x , no domínio global ao final do último passo de propagação.	101
4.16	Componente de deslocamento horizontal, u_x , observado para os 5 passos de propagação da trinca no domínio local.	102
4.17	Componente de tensão σ_{YY} inerente aos 5 passos de propagação da trinca.	103

4.18	Pórtico com oito trincas em modo misto de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade.	105
4.19	Pórtico com 8 mesoescalas: Representação gráfica do dispêndio de tempo total em função do número de mesoescalas e processadores. Trincas estáticas.	117
4.20	Pórtico com 8 mesoescalas: Parcelas de tempo referentes aos trechos sequenciais e paralelizáveis dos algoritmos do MEF G^l -GLI. Trincas estáticas.	121
4.21	Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise para cálculo do <i>speedUp</i> . Trincas estáticas	122
4.22	Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.	125
4.23	Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$). Trincas em propagação.	126
4.24	Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando 5 passos de propagação de trinca.	127
4.25	Chapa com dez trincas em Modo I de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos das escalas meso e local associados às trinca da direita, podem ser obtidos por simetria.	129
4.26	Chapa com dez mesoescalas: Representação gráfica do dispêndio de tempo em função do número de mesoescalas e processadores. Trincas estáticas. . .	131
4.27	Chapa com dez mesoescalas: Parcelas de tempo referentes aos trechos sequenciais e paralelizáveis dos algoritmos do MEF G^l -GLI. Trincas estáticas.	133
4.28	Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise. Trincas estáticas.	135
4.29	Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.	136
4.30	Chapa com dez mesoescalas: Representação gráfica do dispêndio de tempo total em função do número de mesoescalas e processadores considerando cinco passos de propagação.	139

4.31	Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise. Trincas em propagação. . . .	140
4.32	Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas em propagação.	141
A.1	Geometria em planta da estrutura analisada.	153

LISTA DE TABELAS

4.1	Resultados em termos dos fatores de intensidade de tensão obtidos pela abordagem monolítica.	91
4.2	Resultados em termos dos fatores de intensidade de tensão obtidos pela abordagem <i>staggered</i>	91
4.3	Solução de referência para K_I referente aos 5 passos de propagação da trinca.	96
4.4	Resultados de K_I obtidos pela abordagem monolítica.	99
4.5	Resultados de K_I obtidos pela abordagem <i>staggered</i>	99
4.6	Forças concentradas equivalentes a solicitação de vento para o pórtico de 8 pavimentos.	106
4.7	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 1 e 2.	107
4.8	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 3 e 4.	108
4.9	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 5 e 6.	109
4.10	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 7 e 8.	110
4.11	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo <i>staggered</i> - Valores correspondentes às mesoescalas 1 e 2.	111
4.12	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo <i>staggered</i> - Valores correspondentes às mesoescalas 3 e 4.	112
4.13	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo <i>staggered</i> - Valores correspondentes às mesoescalas 5 e 6.	113

4.14	Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo <i>staggered</i> - Valores correspondentes às mesoescalas 7 e 8.	114
4.15	Pórtico com 8 mesoescalas: Número de iterações e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas estáticas.	116
4.16	Pórtico com 8 mesoescalas: Número de iterações e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem <i>staggered</i> . Trincas estáticas.	116
4.17	Pórtico com 8 mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas estáticas.	118
4.18	Pórtico com 8 mesoescalas: Medidas de desempenho inerentes a abordagem <i>staggered</i> considerando o tempo total da análise. Trincas estáticas.	119
4.19	Pórtico com 8 mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.	124
4.20	Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$) para a abordagem monolítica. Trincas em propagação	126
4.21	Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$) para a abordagem <i>staggered</i> . Trincas em propagação.	126
4.22	Pórtico com 8 mesoescalas: Medidas de desempenho considerando 5 passos de propagação de trinca.	127
4.23	Chapa com dez mesoescalas: Número de iterações (N. Iter.) e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas estáticas.	130
4.24	Chapa com dez mesoescalas: Número de iterações (N. Iter.) e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem <i>staggered</i> . Trincas estáticas.	130
4.25	Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas estáticas.	134
4.26	Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem <i>staggered</i> considerando o tempo total da análise. Trincas estáticas.	134
4.27	Chapa com dez mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.	137
4.28	Fator de Intensidade de Tensão extraído ao longo dos cinco passos de propagação.	138

4.29	Chapa com dez mesoescalas: Dispêndio de tempo inerente ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas em propagação.	138
4.30	Chapa com dez mesoescalas: Dispêndio de tempo inerente ao processo de solução sequencial ($n_p = 1$) via abordagem <i>staggered</i> . Trincas em propagação.	138
4.31	Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas em propagação.	140
4.32	Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem <i>staggered</i> considerando o tempo total da análise. Trincas em propagação.	140
4.33	Chapa com dez mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas em propagação.	141

LISTA DE ABREVIATURAS

EPD	Estado Plano de Deformação
EPT	Estado Plano de Tensão
GLI	Global-Local Iterativo
HNA	<i>Hierarchical Non-Intrusive Algorithm</i>
IGL-FEM	<i>Iterative Global-Local Finite Element Method</i>
INSANE	<i>INteractive Structural ANalysis Environment</i>
MEF	Método dos Elementos Finitos
MEF-GL	Método dos Elementos Finitos Global-Local
MEF-GLI	Método dos Elementos Finitos Global-Local Iterativo
MEFG	Método dos Elementos Finitos Generalizados
MEFG ^{gl}	Método dos Elementos Finitos Generalizados com enriquecimento Global-Local
MEFG ^{gl} -GLI	Abordagem não intrusiva do Método dos Elementos Finitos Generalizados com enriquecimento Global-Local
MFLE	Mecânica da Fratura Linear Elástica
POO	Programação Orientada à Objetos
PU	Partição da Unidade
XFEM	<i>eXtended Finite Element Method</i>
ZZ - BD	Campo de tensões recuperadas baseado no estimador de erro ZZ (<i>Zienkiewicz e Zhu</i>)

LISTA DE SÍMBOLOS

Sobrescritos

\cdot^0	Índice que se refere aos graus de liberdade do problema global inicial
$\cdot^{(1)}$	Índice relativo ao modo I de abertura
$\cdot^{(2)}$	Índice relativo ao modo II de abertura
\cdot^E	Índice que se refere ao problema global enriquecido
\cdot^D	Índice relativo às condições de contorno de Dirichlet
\cdot^N	Índice relativo às condições de contorno de Neumann
\cdot^i	Índice relativo à iteração do GLI
\cdot^j	Índice relativo à iteração do MEF G^{gl}
\cdot^{gl}	Índice que se refere aos graus de liberdade introduzidos pelo enriquecimento
\cdot^{Γ_I}	Índice relativo à interface entre domínios

Subscritos

\cdot_C	Índice relativo ao domínio complementar
\cdot_G	Índice relativo ao domínio global
\cdot_{GC}	Índice relativo ao domínio complementar no problema global
\cdot_{GL}	Índice relativo ao domínio local no problema global
\cdot_L	Índice relativo ao domínio local
\cdot_I	Índice relativo à interface
\cdot_M	Índice relativo ao domínio da mesoescala
\cdot_{MEnr}	Índice relativo ao domínio da mesoescala enriquecido
\cdot_R	Índice relativo às forças residuais
\cdot_{gl}	Índice relativo à iteração global-local
\cdot_{gli}	Índice relativo à iteração GLI
\cdot_s	Índice relativo à iteração <i>staggered</i>
\cdot_{tot}	Índice relativo às iterações totais

Letras maiúsculas

E	Módulo de elasticidade longitudinal
G	Módulo de elasticidade transversal
H	Funções de Heaviside
\mathcal{H}^1	Espaço de Hilbert de ordem 1
I_j	Conjunto de funções de enriquecimento do MEFG
J	Jacobiano do elemento global
\mathbf{K}	Matriz de rigidez
L_{ji}	Funções de enriquecimento do MEFG
\mathcal{N}_j	Funções Lagrangianas Lineares
Q	Constante da função de singularidade
V	Volume do elemento utilizado no sistema paramétrico
\mathcal{X}	Espaço de discretização

Letras minúsculas

a	comprimento inicial da trinca
\mathbf{b}_{ji}	Parâmetros nodais associados ao MEFG
\mathbf{f}	Vetor de forças
l	dimensão do elemento finito
n	Número de dimensões espaciais
n_{GL}	Número de ciclos global-local
r	Coordenada Polar com origem na ponta da trinca
\mathbf{t}	Vetor de forças de superfície
$\bar{\mathbf{t}}$	Vetor de forças de superfície prescritas
u_y	Componente de deslocamento na direção y
\mathbf{u}	Solução numérica aproximada
$\tilde{\mathbf{u}}$	Solução genérica do MEFG
$\bar{\mathbf{u}}$	Solução global prevista
\mathbf{u}_G^0	Solução global inicial
\mathbf{u}_j	Parâmetros nodais associados ao MEF
\mathbf{u}_f	Solução numérica de referência do GLI
u_x	Função de singularidade na direção “x”
u_y	Função de singularidade na direção “y”
\mathbf{v}	Funções tentativa
x	Posição em um domínio
\mathbf{x}_j	Vetor de coordenadas de um nó j
SU	<i>SpeedUp</i>
E	Eficiência da paralelização

Letras gregas maiúsculas

β	Inclinação inicial da trinca
Γ_I	Interface entre os domínios
Δ^i	Vetor de incremento de deslocamentos de Aitken
Δa	Incremento de propagação da trinca
Δu	Incremento da solução numérica
Ω	Domínio aberto e suave
$\partial\Omega$	Contorno do domínio Ω
ϕ_{ji}	Função de forma do MEFG

Letras gregas minúsculas

ϵ	Tolerância para o resíduo relativo
ε	Tensor de deformações
θ	Coordenada polar com origem na ponta da trinca
κ	Parâmetro de penalidade
$\bar{\kappa}$	Constante determinada a partir do coeficiente de Poisson
λ	Constante da função de singularidade
ν	Coefficiente de Poisson
ξ	Posição da descontinuidade
σ_{xx}	Componente de tensão na direção x
$\boldsymbol{\sigma}$	Tensor de tensões
ω	Fator de relaxação
ω_j	Nuvem de elementos que concorrem no nó j

Operadores e símbolos especiais

\forall	Para qualquer que seja
\cup	União de conjuntos
\cap	Interseção de conjuntos
\subset	Está contido
\emptyset	Conjunto vazio
$\ a\ $	Norma euclidiana do vetor \mathbf{a}
$\underline{\mathbf{a}}$	Prolongamento do vetor ou da matriz \mathbf{a}

Conjuntos numéricos

\mathbb{R}	Conjunto dos números reais
--------------	----------------------------

SUMÁRIO

1	Introdução	22
1.1	Contextualização	22
1.2	Justificativa	24
1.3	Objetivos	26
1.3.1	Objetivo geral	26
1.3.2	Objetivos específicos	26
1.4	Metodologia	26
1.4.1	Aspectos de formulação	26
1.4.2	Aspectos computacionais	27
1.4.3	Validação dos resultados	27
1.5	Organização do texto	28
2	Revisão da Literatura	29
2.1	Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI)	29
2.1.1	Convergência do algoritmo GLI	35
2.2	Método dos Elementos Finitos Generalizados	37
2.3	Método dos Elementos Finitos Generalizados com Enriquecimento Global-Local - MEFG ^{gl}	40
2.3.1	Principais aspectos inerentes a estratégia Global-Local	44
2.3.2	Modelagem de trincas através do MEFG ^{gl}	47
2.4	MEFG ^{gl} -GLI	49
2.4.1	Abordagem monolítica	52
2.4.2	Abordagem <i>staggered</i>	54
2.5	Computação Paralela	55
2.5.1	Computação de Alto Desempenho	56
2.5.2	Paralelismo Local	57
2.5.3	Medidas de desempenho	58
2.5.4	Paralelização no âmbito do Método dos Elementos Finitos	59
3	Aspectos Computacionais	62
3.1	Algoritmos de solução do MEFG ^{gl} -GLI aplicado a problemas com múltiplas mesoescalas	62

3.1.1	Algoritmo de solução monolítica	63
3.1.2	Algoritmo de solução <i>staggered</i>	66
3.2	Aprimoramento dos algoritmos de solução do MEFG ^{gl} -GLI	69
3.2.1	Algoritmo de solução monolítica	70
3.2.2	Algoritmo de solução <i>staggered</i>	72
3.3	Implementação dos métodos de leitura, gerenciamento e exportação de dados	74
3.4	Implementação paralelizada da estratégia de acoplamento MEFG ^{gl} -GLI em análises com múltiplas mesoescalas	76
3.4.1	Classe IGLGlobalLocalCrackPropagationSimulationManager	77
3.4.2	Classe IGLGlobalLocal	81
4	Simulações Numéricas	86
4.1	Validação da implementação	87
4.1.1	Pórtico com Trincas em Modo Misto de abertura	87
4.1.2	Chapa com Trincas de borda em Modo I de abertura	94
4.2	Desempenho Computacional	104
4.2.1	Pórtico com oito Trincas em Modo Misto de abertura	104
4.2.2	Chapa com dez trincas em Modo I de abertura	128
5	Considerações Finais	142
5.1	Considerações deste Trabalho	142
5.1.1	Validação da implementação	142
5.1.2	Desempenho Computacional	143
5.1.3	Relevância da pesquisa	144
5.2	Sugestões para Trabalhos Futuros	145
	Apêndices	152
A	Cálculo das forças de Vento	153

CAPÍTULO 1

INTRODUÇÃO

1.1 Contextualização

A formulação matemática de problemas de engenharia estrutural, em geral, exige hipóteses simplificadoras para que seja possível descrevê-los de forma viável. Na maioria das situações, as equações diferenciais que regem esses modelos não admitem soluções analíticas simples, o que torna sua resolução pouco prática do ponto de vista computacional. Como alternativa, recorre-se a métodos numéricos aproximados, os quais permitem representar de maneira adequada o comportamento do sistema em análise. Entre esses métodos, o Método dos Elementos Finitos (MEF) ocupa posição de destaque, tanto no meio acadêmico quanto na prática profissional. Nesse contexto, o MEF consiste em dividir o domínio do problema em subdomínios finitos de geometria reduzida, denominados elementos, cujas respostas locais, associadas a parâmetros nodais, permitem a construção dos campos globais de deslocamentos e tensões do modelo.

Nos problemas da Mecânica da Fratura, os fenômenos de nucleação e coalescência de microvazios podem levar à falha de componentes estruturais por meio da formação de trincas. Esses casos envolvem descontinuidades e concentrações de tensões, resultando em soluções não suaves e exigindo um refinamento excessivo da malha de elementos na formulação convencional do MEF, o que torna o processo computacionalmente oneroso. Para superar essas limitações, foram desenvolvidas variações do método capazes de lidar, de forma mais eficiente com certos fenômenos presentes nos modelos analisados. Entre essas variações, destaca-se a estratégia denominada Global-Local Iterativa-GLI (Whitcomb, 1991), que permite aplicar o refinamento da malha em regiões bem definidas do domínio onde se encontram as singularidades.

Outro aprimoramento proveniente dessa abordagem é o acoplamento não intrusivo, que viabiliza a integração de algoritmos desenvolvidos no ambiente acadêmico com programas comerciais. Essa estratégia disponibiliza à indústria formulações avançadas, como o Método dos Elementos Finitos Generalizados - MEFG (Strouboulis *et al.*, 2000), com enriquecimento Global-Local - MEFG^{gl} (Duarte e Kim, 2008), em combinação com programas amplamente reconhecidos por sua robustez e familiaridade entre engenheiros de projeto.

No MEFG utilizam-se, geralmente, as funções de forma convencionais do MEF multiplicadas por funções enriquecedoras, previamente definidas com base no comportamento esperado da solução, aprimorando a representação do modelo. Um aspecto adicional favorável, conforme destacado em Barros (2002), é que a formulação do MEFG proporciona uma simulação numérica com relativa independência da malha de elementos empregada, reduzindo potenciais efeitos adversos na solução decorrentes de escolhas do analista.

Somado a este aprimoramento, no MEFG^{gl}, a solução do problema é conduzida em três etapas. Inicialmente, realiza-se uma discretização grosseira de todo o domínio. Em seguida, soluciona-se um problema local definido em uma região que contém singularidades nos gradientes de deslocamentos, como trincas, utilizando os resultados previamente obtidos como condições de contorno. Finalmente, o problema global é reprocessado enriquecido com funções construídas a partir da solução do problema local, incorporando os efeitos dos fenômenos localizados ao comportamento do modelo global.

A estratégia de enriquecimento do MEFG^{gl} possibilita o emprego de diferentes níveis de discretização para cada escala do problema, sem a necessidade de empregar elementos distorcidos ou técnicas de remalhamento para compatibilizar a interface entre os problemas global e local. Em geral, a discretização do domínio do modelo inicial é mantida e apenas o modelo local é refinado. Esta abordagem minora efeitos deletérios na qualidade da aproximação.

Quanto a implementação não intrusiva (GLI) do MEFG^{gl} em problemas multiescala, destaca-se o trabalho de Li *et al.* (2021), que introduziu o MEFG^{gl}-GLI. Essa abordagem diferencia-se das estratégias anteriores, que frequentemente baseavam o acoplamento em algoritmos de subestruturação, condensação estática da matriz de rigidez global e métodos de Schwarz (Duval *et al.*, 2014), o que restringia a utilização de programas comerciais com código-fonte fechado. Em contraste, o acoplamento não intrusivo elimina a necessidade de modificar os solucionadores dos programas utilizados. A inclusão do MEFG^{gl}, conforme proposto por Li *et al.* (2021), é especialmente relevante para problemas com fenômenos localizados, como a propagação de defeitos em materiais, que exigem remalhamento contínuo na formulação convencional do MEF, aumentando o esforço computacional.

Neste contexto, em continuidade aos trabalhos de Silveira Filho (2023) e Bueno (2024), nesta pesquisa, o MEFG^{gl}-GLI encontra-se acoplado ao programa comercial *Abaqus*, responsável por representar o comportamento global do problema, enquanto o *INSANE* (*INteractive Structural ANalysis Environment*, disponível em www.insane.dees.ufmg.br) é responsável pela descrição dos fenômenos locais por meio do MEFG^{gl}.

A plataforma *INSANE* é um projeto de *software* livre desenvolvido no Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais, implementado em JAVA com base no paradigma da Programação Orientada a Objetos (POO) (Pitangueira *et al.*, 2008). Este ambiente computacional dispõe de recursos que podem ser empregados em análises pelo MEFG e MEFG^{gl}, conforme descrito em Alves (2012), Malekan (2017),

Fonseca (2019) e Fonseca (2023).

A implementação não intrusiva do MEF^{gl}, conforme proposta por Li *et al.* (2021), foi adaptada ao *INSANE* por Silveira Filho (2023) para problemas com comportamento linear elástico, considerando a inclusão de vazios no modelo local ou alterações na rigidez do material. Na referida dissertação, foi desenvolvido um algoritmo em linguagem de programação *Python*, externo ao *INSANE*, para gerenciar a estratégia de acoplamento não intrusivo proposto por Whitcomb (1991). Posteriormente, Bueno (2024) transpôs o algoritmo para *JAVA*, integrando-o ao *INSANE* e expandindo sua aplicação para análises de propagação de trincas em problemas de Mecânica da Fratura Linear Elástica, conforme as contribuições de Fonseca (2019).

O presente trabalho expande a atual versão do MEF^{gl}-GLI implementada no ambiente computacional *INSANE*, possibilitando a análise de modelos com múltiplos domínios locais e, conseqüentemente, a consideração de diversos fenômenos localizados. De modo complementar, foi desenvolvido um gerenciador que particiona o processo de análise entre múltiplos processadores utilizando técnicas de paralelismo, com o objetivo de reduzir o tempo e o esforço computacional. Tal abordagem é justificada pela independência entre os múltiplos modelos locais na estratégia de solução global-local, eliminando a necessidade de transferência de dados ou consideração de influências mútuas durante a análise local. Essas influências são incorporadas posteriormente no modelo global enriquecido. Ressalta-se que esta é a primeira implementação em que o processamento das múltiplas mesoescalas do MEF^{gl}-GLI é realizado de forma paralelizada, representando um avanço inédito desta pesquisa.

1.2 Justificativa

Na maioria das estruturas, ocorrem fenômenos localizados que exigem análises refinadas, como a presença de trincas, furos, regiões de geometria complexa, pontos de apoio e de soldas. A adoção de um modelo único para a análise estrutural que contemple os efeitos de escalas com grandezas significativamente diferentes pode resultar em um custo computacional excessivo ou na incapacidade de representar adequadamente os efeitos localizados. Assim, em prol da eficiência computacional, torna-se atrativo para estruturas de grande porte, que a resposta global seja representada por um modelo que não se preocupe em discretizar os fenômenos locais de interesse, os quais devem ser estudados em modelos particulares e adequadamente refinados, incentivando o uso de métodos que adotem diferentes níveis de discretização.

A análise de grandes estruturas, como um edifício de múltiplos andares, demanda alta capacidade de processamento e, comumente, é realizada em softwares comerciais de elevada robustez, visando representar especialmente o comportamento global do modelo. A descrição dos fenômenos locais e sua influência no comportamento global da estrutura,

no entanto, ficam prejudicadas. Essa penalização ocorre devido a dificuldade em compatibilizar os diferentes níveis exigidos de refinamento da malha e a ausência de estratégias numéricas adequadas para representar tais fenômenos. Uma situação que exemplifica essa limitação é o uso da formulação clássica do MEF para o tratamento de fenômenos de fissuração, conforme descrito por Fonseca (2019).

A estratégia de solução do MEF gl -GLI possibilita a construção de modelos locais considerando a influência de suas regiões de entorno e, posteriormente, a incorporação dos efeitos das singularidades locais no comportamento global. Para tal, vale-se de um processo iterativo que compatibiliza as diferentes escalas de análise. Assim, justifica-se o foco deste trabalho em estudar estratégias de acoplamento que conectam programas de simulação comerciais à *solvers* desenvolvidos nas universidades. A manutenção do uso de programas já empregados na indústria, em uma das esferas de solução, contribui para a receptividade da metodologia proposta no mercado profissional, aumentando a probabilidade de adesão.

Neste contexto, esta pesquisa emprega o programa *ABAQUS* como solucionador do modelo global, por sua robustez e eficiência computacional em análises via MEF, e o ambiente computacional *INSANE* para o estudo dos fenômenos localizados. Considerando que o código-fonte dos programas comerciais é fechado, o acoplamento não intrusivo constitui uma estratégia de solução adequada para integrar formulações e métodos numéricos avançados, implementados no *INSANE*, com programas amplamente utilizados, como o *ABAQUS*.

Embora já existam estudos sobre estratégias de acoplamento, inclusive no âmbito do *INSANE*, como os trabalhos desenvolvidos por Silveira Filho (2023) e Bueno *et al.* (2023), o tema se configura como uma linha de pesquisa recente e em desenvolvimento, com ramificações a serem exploradas e expandidas. Como apresentado, as estratégias de acoplamento têm como objetivo aproximar os *solvers* comerciais e acadêmicos, sendo, portanto, fundamental viabilizar a capacidade de resolução de modelos com múltiplos domínios locais, visto que esta é a realidade dos problemas enfrentados pela indústria. Devido a crescente demanda por estratégias que otimizem o desempenho computacional, a paralelização da solução dos domínios locais no âmbito do MEF gl -GLI configuram uma abordagem promissora, permitindo a distribuição das cargas de processamento em múltiplos núcleos de processamento, reduzindo significativamente o tempo de análise.

Sob essa perspectiva, o direcionamento desta pesquisa em expandir a implementação do MEF gl -GLI no *INSANE* a partir da incorporação de estratégias de paralelismo apresenta-se como uma contribuição relevante para a área, promovendo avanços em soluções eficientes de análise computacional a serem aplicadas no ambiente acadêmico e industrial.

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo geral desta pesquisa consiste em expandir a implementação do MEF G^{gl} -GLI, fundamentada no acoplamento não intrusivo entre o *software ABAQUS* e o sistema *INSANE*, possibilitando a análise de problemas da mecânica da fratura com múltiplos domínios locais.

1.3.2 Objetivos específicos

Como objetivos específicos deste trabalho, podem ser mencionados:

- i. Contribuir para o avanço do conhecimento acerca da aplicação de técnicas de acoplamento não intrusivo como metodologia para a resolução de problemas na engenharia de estruturas, com ênfase no balanceamento entre custo computacional e qualidade da solução obtida;
- ii. Implementar uma aplicação para leitura, gerenciamento e exportação dos parâmetros de análise (*input* e *output*) de problemas com múltiplos domínios locais, ocorrendo a entrada de dados majoritariamente via arquivo XML, dispensando a intervenção do usuário no código computacional;
- iii. Avaliar o impacto da taxa de convergência do GLI de cada domínio local na convergência global do problema;
- iv. Investigar a sensibilidade da solução em função da interferência mútua entre os múltiplos domínios locais e seus respectivos fenômenos localizados;
- v. Avaliar a eficiência computacional para os casos de múltiplos domínios analisados sequencialmente ou por processamento paralelo.

1.4 Metodologia

O presente trabalho foi fundamentado nas seguintes etapas:

1.4.1 Aspectos de formulação

Estudo dos fundamentos teóricos e aplicações das principais teorias e métodos aplicados nesta pesquisa, a saber:

- o MEF G e o MEF G^{gl} , visando a familiarização com os principais conceitos, o histórico e a motivação para o desenvolvimento do método e sua formulação. O foco principal recai sobre os casos que abordam estratégias utilizadas para a solução de problemas envolvendo trincas;

- simulação da propagação de trincas baseada na estratégia de enriquecimento do MEFG;
- algoritmos para acoplamento não intrusivo no contexto da análise global-local, com ênfase no MEFG^{gl}-GLI.

1.4.2 Aspectos computacionais

Estudo acerca das linguagens de programação e programas inerentes a implementação e operação do algoritmo apresentado nesta pesquisa, destacando:

- familiarização com a linguagem de programação JAVA e o paradigma de programação orientada a objetos (POO), devido ao seu uso nas implementações desta pesquisa;
- ambientação no sistema *INSANE*, abrangendo sua organização macro (divisão nos núcleos pré-processador, processador e pós-processador), e aplicação e funcionamento das implementações referentes ao MEFG^{gl}-GLI desenvolvidas por Silveira Filho (2023) e Bueno (2024);
- familiarização com o programa comercial *Abaqus*, compreendendo suas funcionalidades e as especificidades envolvidas em seu acoplamento ao sistema *INSANE* por meio de *scripts* em linguagem *Python*;
- estudo de estratégias para processamento paralelo utilizando a linguagem JAVA;
- implementação da estratégia proposta para expansão do sistema *INSANE* possibilitando a execução de análise via MEFG^{gl}-GLI com múltiplos domínios locais, inicialmente resolvidos de forma sequencial e, posteriormente, paralelamente.

1.4.3 Validação dos resultados

Etapa de validação dos resultados e análise da eficiência computacional do método, desenvolvida a partir de:

- seleção de experimentos numéricos com solução de referência para a validação do algoritmo desenvolvido;
- avaliação da convergência inerente à estratégia GLI e a interferência no fluxo do processamento em função da variação da taxa de transferência de informações entre o modelo resolvido no *ABAQUS* e os modelos de solução global-local resolvidos no *INSANE*;
- expansão da aplicação da implementação proposta nesta pesquisa para modelos mais complexos, com o objetivo de mensurar a eficiência computacional associada ao processamento sequencial ou paralelo.

1.5 Organização do texto

Quanto à organização do texto, este trabalho está dividido em 5 capítulos e 1 apêndice, conforme descrito a seguir.

O Capítulo 1 apresenta uma introdução concisa do trabalho, justificando o tema abordado e a relevância da pesquisa. São definidos também os objetivos gerais e específicos, bem como a metodologia empregada para alcançá-los.

O Capítulo 2 consiste em uma revisão da literatura, destacando as principais referências e os conceitos fundamentais relacionados aos métodos numéricos aplicados neste estudo.

No Capítulo 3, são discutidos os aspectos computacionais das implementações realizadas no ambiente *INSANE*, com uma descrição geral dos procedimentos efetuados no código, incluindo as principais classes e métodos adicionados ou revisados, bem como as funcionalidades da expansão do sistema.

O Capítulo 4 apresenta as simulações numéricas realizadas com o objetivo de validar a implementação desenvolvida nesta pesquisa, além de discutir aspectos relacionados à eficiência computacional e à influência dos principais parâmetros da estratégia de acoplamento GLI.

No Capítulo 5, são apresentadas as considerações finais deste trabalho, suas contribuições e relevância, além de sugestões para o desenvolvimento de pesquisas futuras relacionadas ao tema.

O Apêndice A descreve o processo de cálculo para definição das cargas de vento aplicadas nas simulações numéricas, conforme a NBR-6123 (ABNT (2023)).

CAPÍTULO 2

REVISÃO DA LITERATURA

Neste capítulo são apresentados os conceitos e métodos relevantes para o tema abordado nesta dissertação, com ênfase na abordagem não intrusiva do Método dos Elementos Finitos Generalizados com enriquecimento Global-Local, acoplado ao algoritmo Global-Local Iterativo (MEFG^{gl}-GLI).

2.1 Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI)

Uma ampla gama de problemas na mecânica dos sólidos envolve fenômenos físicos complexos, como trincas, soldas, não linearidades físicas, elevados gradientes de tensão, entre outros. Comumente, esses fenômenos se manifestam em regiões bem definidas, o que motiva uma investigação mais detalhada dessas áreas específicas do problema, visto que, o refinamento de domínios extensos pode gerar uma sobrecarga computacional significativa, dificultando, inclusive, a manipulação do modelo por meio de interfaces gráficas. Com base nessa motivação, diversos métodos computacionais foram desenvolvidos para solucionar problemas multiescala¹ ou de decomposição do domínio, visando à redução do custo computacional e a melhoria da qualidade da solução, permitindo equilibrar a demanda computacional com as limitações do *hardware* disponível. Como destacado por Duval *et al.* (2014), os avanços em computação paralela e técnicas de multiprocessamento têm aprimorado a eficiência de métodos de acoplamento e de estratégias multi-escala, além de impulsionar sua disseminação no meio científico e comercial.

Os métodos voltados para essa abordagem podem ser divididos em dois grupos principais: os que recorrem ao enriquecimento das funções de forma do MEF, como ocorre com a partição da unidade no Método dos Elementos Finitos Generalizados (MEFG - explorado na seção 2.2), e os que adotam o acoplamento entre modelos, como o Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI) (Duval *et al.*, 2014).

¹O termo multiescala mencionado neste trabalho refere-se a diferentes escalas de discretização, e não está relacionado a soluções do tipo multifísico, onde as várias escalas representam o problema físico de maneira distinta.

De acordo com Silveira Filho (2023), os métodos com acoplamento de modelos apresentam as seguintes vantagens:

- a matriz de rigidez de cada domínio não sofre influência dos resultados dos demais, não sendo necessário reconstruí-la ou invertê-la novamente. No MEEG^{gl}-GLI, entretanto, essa premissa é perdida devido à estratégia de enriquecimento;
- o modelo global permanece inalterado durante o processo de solução;
- em situações em que o modelo computacional atinge grande extensão, a manipulação através da interface gráfica torna-se lenta, sendo possível decompor o domínio em problemas menores;
- em alguns casos, uma das escalas do problema pode ser melhor resolvida por um programa, cujo desempenho computacional não justifique sua utilização para simular o problema como um todo. Nesse caso, torna-se vantajoso o uso de diferentes plataformas computacionais para cada esfera de solução.

As estratégias de acoplamento podem ser intrusivas, quando há interferência na matriz de rigidez ou no solucionador do sistema de equações. Em contrapartida, abordagens não intrusivas não exigem alterações no código-fonte. Esta característica torna o acoplamento com programas comerciais particularmente atraente, uma vez que o código não é disponibilizado para modificações.

O uso de estratégias de acoplamento não intrusivo na resolução de problemas com diferentes escalas de discretização por meio do MEF foi inicialmente proposto por Whitcomb (1991), com o Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI). Segundo Li *et al.* (2021), esse método pode ser visto como uma variação da técnica *zooming*, que consiste na solução de um problema global (correspondente ao domínio completo) utilizando uma malha grosseira de elementos, seguida pela análise de domínios locais (definidos nas regiões críticas do modelo) com uma discretização mais refinada, apta a descrever o comportamento nas proximidades do fenômeno localizado. A solução do problema global é utilizada como condição de contorno para o modelo local, comumente, em termos de deslocamentos, visto que, a adoção de forças na interface entre os domínios pode não ser autoequilibrada (Figura 2.1). A principal vantagem da técnica *zooming* é o ganho de eficiência computacional, proporcionado pelo uso de malhas refinadas apenas em regiões restritas e bem definidas do modelo.

Conforme apontado por Noor (1986), as principais limitações da técnica de *zooming* estão associadas:

- a interface entre os problemas local e global, sendo que a compatibilidade entre a malha grosseira global e a refinada local advém do uso de elementos de transição;
- ao nível de qualidade necessário para a solução do problema global que é utilizado como condição de contorno no modelo local;
- à falta de interação direta entre o modelo local e o global.

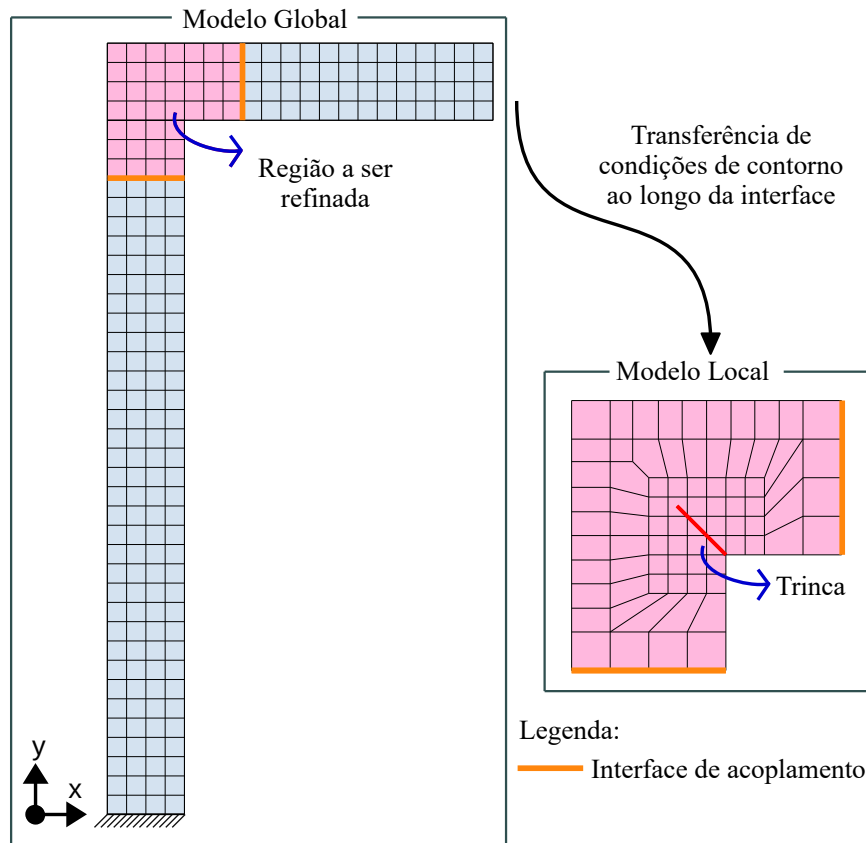


Figura 2.1: Representação esquemática do processo de solução via técnica *zooming*.

Um aspecto de grande relevância do MEF-GLI está relacionado ao último item mencionado na listagem anterior, pois esse método envolve um processo iterativo que impõe condições de equilíbrio e continuidade entre os modelos global e local (Li *et al.*, 2021). Para ilustrar o processo de solução utilizando esse método, considera-se o pórtico em “L” apresentado na Figura 2.2. O modelo global corresponde a uma discretização grosseira do domínio completo, ou seja, $\Omega = \Omega_L \cup \Omega_C$, sendo Ω_L o domínio local e Ω_C o domínio complementar. O modelo local é uma discretização refinada do domínio Ω_L , definida na região de conexão entre a viga e o pilar no pórtico, onde ocorre a presença de uma trinca, que induz um campo de deslocamento descontínuo com elevados gradientes. A região $\Gamma_I = \partial\Omega_L \cap \partial\Omega_C$ é definida como a interface entre os modelos global e local. A rigidez do modelo local é representada por \mathbf{K}_L , enquanto a rigidez do modelo global, \mathbf{K}_G , é composta por duas parcelas: a rigidez \mathbf{K}_{GL} , correspondente a região do modelo global coincidente com o domínio local, e a rigidez \mathbf{K}_{GC} , referente a região complementar.

Nas equações matriciais apresentadas a seguir, o operador de adição não representa a soma direta entre as matrizes, mas sim a soma matricial ou vetorial apropriada após a compatibilização dos graus de liberdade envolvidos.

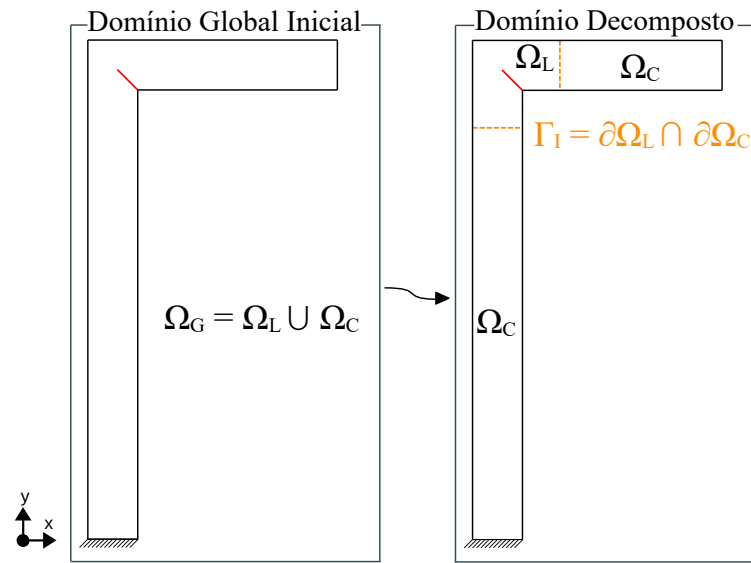


Figura 2.2: Domínios do pórtico em “L” analisado via MEF Global-Local Iterativo (MEF-GLI).

Em análises lineares, a equação que rege o modelo global é dada por:

$$\begin{aligned} \mathbf{K}_G \cdot \mathbf{u}_G &= f_G \\ (\mathbf{K}_{GC} + \mathbf{K}_{GL}) \cdot \mathbf{u}_G &= f_G \end{aligned} \quad (2.1)$$

O campo de deslocamentos \mathbf{u}_G , obtido através da Equação (2.1), corresponde a solução do problema global inicial, denotada por \mathbf{u}_G^0 , e é empregado como condição de contorno de Dirichlet para o problema local.

De modo similar, a aproximação do problema local, \mathbf{u}_L , é obtida por meio da Equação (2.2):

$$\mathbf{K}_L \cdot \mathbf{u}_L = \mathbf{f}_L \quad (2.2)$$

É importante ressaltar que os modelos global e local representam Ω_L de maneira distinta, uma vez que, na maioria dos casos, $\mathbf{K}_L \neq \mathbf{K}_{GL}$. Na interface Γ_I entre os modelos global e local, ocorre um desequilíbrio de forças, mas uma compatibilidade de deslocamentos é assegurada pelo uso de \mathbf{u}_G (calculado a partir da Equação (2.1), onde se emprega \mathbf{K}_{GL}) como condição de contorno para a solução de \mathbf{u}_L (calculado a partir da Equação (2.2), onde se utiliza \mathbf{K}_L). Nesse contexto, Whitcomb (1991) propôs um método iterativo para alcançar o equilíbrio de forças em Γ_I por meio de sucessivas correções no vetor de forças do modelo global — esta estratégia corresponde à Equação (2.3), explicitada no item 3 do procedimento de solução do MEF-GLI.

O procedimento de solução é descrito de forma detalhada abaixo:

1. Análise global: resolver o modelo global obtendo a solução inicial \mathbf{u}_G^0 ;
2. Análise local: definir e resolver os modelos locais obtendo a solução \mathbf{u}_L^i . Nesta etapa

emprega-se a solução inicial, \mathbf{u}_G^0 , como condição de contorno ao longo da interface Γ_I de cada modelo local;

3. Cálculo de resíduos: a partir das soluções \mathbf{u}_G^i e \mathbf{u}_L^i associadas à iteração i , calcular as reações de forças nodais no perímetro Γ_I para o problema global (Ω_C) e para o local (Ω_L), respectivamente. A partir da diferença destes esforços, obtém-se o vetor resíduo empregado para atualizar o vetor de forças do modelo global (multiplicado por -1 em função do cálculo se tratar de reações de apoio, que naturalmente possuem sentido contrário ao carregamento):

$$\mathbf{f}_R = -(\mathbf{f}_L^{\Gamma_I} + \mathbf{f}_C^{\Gamma_I}) = -[(\mathbf{K}_L \cdot \mathbf{u}_L^i - \mathbf{f}_L) |_{\Gamma_I} + (\mathbf{K}_{GC} \cdot \mathbf{u}_{GC}^i - \mathbf{f}_{GC}) |_{\Gamma_I}] \quad (2.3)$$

onde \mathbf{f}_L e \mathbf{f}_{GC} são os vetores de forças correspondentes ao contorno Γ_I , referente ao modelo local e global, respectivamente.

4. Verificar se o critério de convergência foi atingido, ou seja, a norma do vetor resíduo é menor que a tolerância adotada. Em caso contrário, o processo iterativo tem continuidade com a atualização do modelo global (através da soma do resíduo \mathbf{f}_R ao vetor de forças do modelo global por meio da Equação (2.1));
5. Iterações i global-local: retorna-se ao passo 1 para obter uma nova solução \mathbf{u}_G^i para o problema global, repetindo-se as etapas subsequentes descritas, com o objetivo de alcançar a convergência (compatibilidade de deslocamentos e equilíbrio de forças em Γ_I) e a solução final de referência (\mathbf{u}_{ref}) a ser obtida.

A solução final corresponde a combinação das soluções global e local da seguinte forma: \mathbf{u}_{GC} (deslocamentos da solução global, \mathbf{u}_G , correspondente aos nós que compõem o domínio complementar) válida no domínio Ω_C e \mathbf{u}_L (proveniente da solução do problema local) válida no domínio Ω_L . Na região da interface entre os domínios, a solução é coincidente, conforme indicado na etapa 5 do procedimento de solução. Havendo convergência do método, o resultado final é equivalente a solução de um modelo de referência composto pelas malhas do global complementar e do local, \mathbf{u}_{ref} .

Conforme exposto por Li *et al.* (2021), como o acoplamento entre os modelos global e local ocorre pela transferência de deslocamentos (obtidos do problema global) e forças nodais (provenientes da etapa de cálculo 3), que comumente são os dados de saída de programas de elementos finitos, o MEF-GLI é uma estratégia de solução não intrusiva.

Conforme apontado por Silveira Filho (2023), o MEF-GLI apresenta dois pontos críticos:

1. Variação de rigidez: a taxa de convergência da solução depende da diferença de rigidez entre o problema global (\mathbf{K}_{GL}) e o problema local (\mathbf{K}_L) na região do domínio Ω_L ;

2. Conformidade da malha: a eficiência computacional e a facilidade de implementação do método são fortemente dependentes da disposição dos elementos na interface de contato, Γ_I .

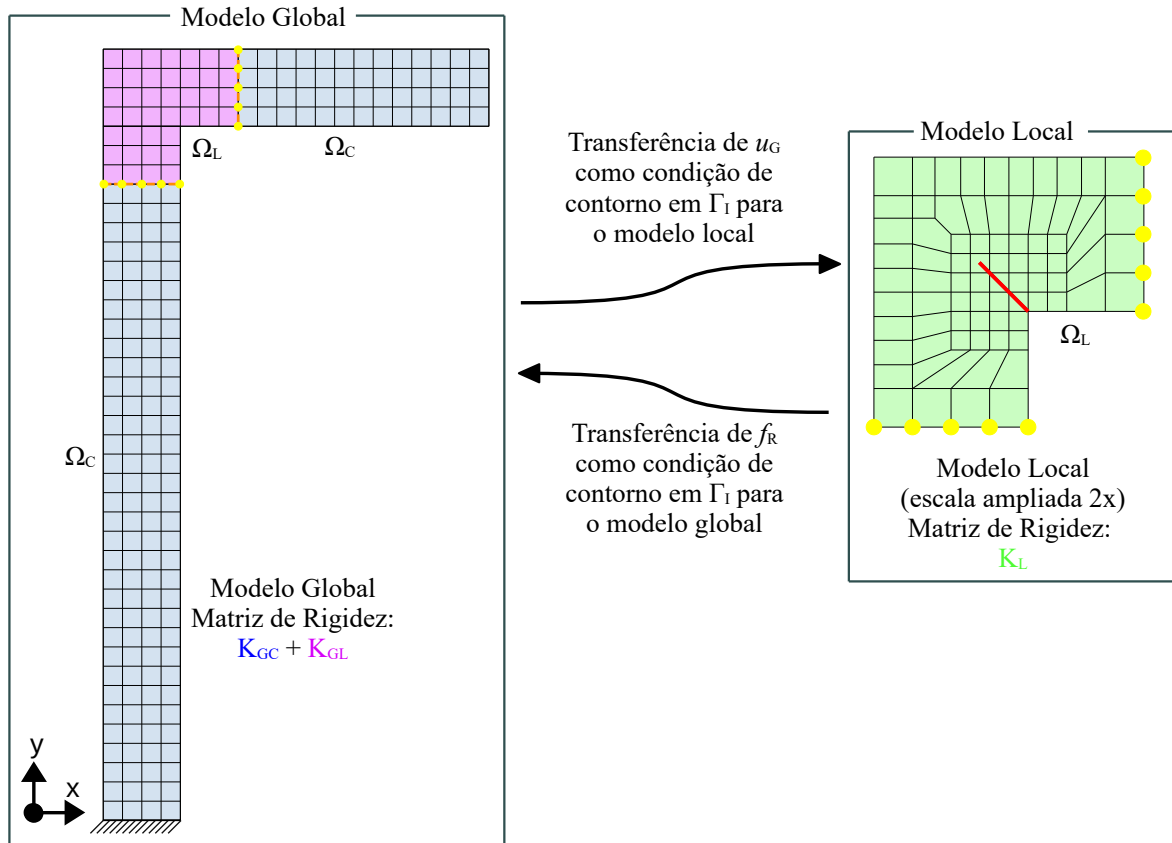


Figura 2.3: Representação esquemática do processo de solução via Método dos Elementos Finitos Global-Local Iterativo (MEF-GLI).

Quanto ao primeiro tópico, existem duas situações possíveis. O primeiro caso ocorre quando o modelo local é menos rígido que o modelo global (*softening* local), o que resulta em uma taxa de convergência mais lenta a medida que $\mathbf{K}_L \ll \mathbf{K}_{GL}$, embora a convergência seja sempre alcançada (Whitcomb, 1991). Este é o cenário mais frequente nas análises, pois, geralmente, a malha local é mais refinada, produzindo uma solução mais flexível.

A segunda situação ocorre quando o modelo local é mais rígido que o modelo global, o que pode acontecer, por exemplo, em presença de enrijecedores e/ou soldas no domínio Ω_L . Neste caso, não se garante o equilíbrio de forças em Γ_I , e o método de cálculo pode divergir.

Diversas técnicas podem ser empregadas para melhorar a convergência do método, em caso de lentidão no processo, ou mesmo garantir a convergência em problemas com modelos locais mais rígidos que o modelo global. Li *et al.* (2021) destacam os relaxamentos estático e dinâmico (Duval *et al.*, 2014), que visam limitar o tamanho do passo de correção

da solução global aplicada como condição de contorno no modelo local a cada iteração.

Quanto a conformidade da malha, segundo Whitcomb (1991), é necessário aplicar estratégias especiais de acoplamento nos casos em que os pontos nodais das malhas do modelo global e local não coincidem na interface Γ_I . Devido a dificuldade em conformar malhas que garantam essa característica, algumas estratégias podem ser empregadas para lidar com essa incompatibilidade, incluindo o uso de elementos de transição, multiplicadores de Lagrange e o método de Nitsche (Li *et al.*, 2021). Contudo, a adoção desses procedimentos leva ao aumento do custo computacional e a dificuldades de implementação, o que motivou a utilização do Método dos Elementos Finitos Generalizados com enriquecimento Global-Local, conforme será abordado na seção 2.4.

2.1.1 Convergência do algoritmo GLI

A definição do critério de convergência do GLI é estabelecida pelo analista, sendo neste trabalho adotada uma tolerância da ordem de 10^{-5} para o resíduo relativo, conforme utilizado nos trabalhos de Li *et al.* (2021); Li *et al.* (2022); Silveira Filho (2023) e Bueno (2024), que obtiveram resultados acurados. O resíduo relativo da iteração i é obtido a partir da razão entre as normas euclidianas do vetor de forças residuais na referida iteração, \mathbf{f}_R^i , e do vetor de forças residuais da primeira iteração, \mathbf{f}_R^0 .

Conforme Allix e Gosselet (2020) é conveniente empregar algoritmos numéricos refinados no tratamento da convergência do algoritmo GLI, possibilitando, inclusive, reverter a divergência da solução em alguns casos. Dentre esses algoritmos, destacam-se as técnicas de relaxação, classificadas por Liu *et al.* (2014) em dois grupos principais:

- Técnicas associadas ao método quasi-Newton: a adoção dessa estratégia implica, normalmente, em alterações na matriz de rigidez do modelo global, o que comprometeria a não intrusividade do GLI;
- Técnicas que incidem apenas nas funções que representam a solução do modelo global na i -ésima iteração, $f(\mathbf{u}_G^i)$: essas técnicas proporcionam um aprimoramento da convergência, preservam a não intrusividade e demandam menor esforço computacional e de implementação, visto que corrigem a solução \mathbf{u}_G^i utilizando apenas as soluções das iterações anteriores. Dentre essas técnicas, destacam-se a relaxação estática (Seção 2.1.1.1) e a relaxação dinâmica (Seção 2.1.1.2).

A aplicação das técnicas de relaxação mencionadas requer que a solução do problema global, dada pela Equação (2.1), seja reescrita em sua forma iterativa (Duval *et al.*, 2014). Para uma dada iteração i , tem-se:

$$\begin{aligned} \mathbf{u}_G^{i+1} &= \mathbf{u}_G^i + \mathbf{K}_G^{-1} f(\mathbf{u}_G^i) \\ \mathbf{u}_G^{i+1} &= \mathbf{u}_G^i + (\mathbf{K}_{GC} + \mathbf{K}_{GL})^{-1} f(\mathbf{u}_G^i) \end{aligned} \tag{2.4}$$

onde $f(\mathbf{u}_G^i) = -\mathbf{f}_R^i$, e, \mathbf{f}_R^i corresponde ao prolongamento ² do vetor resíduo (\mathbf{f}_R) para o espaço do problema global na i -ésima iteração.

2.1.1.1 Relaxação estática

Consiste na abordagem mais simples entre as técnicas de relaxação, corrigindo a aproximação com base em um ponderador escalar constante (fator de relaxação, ω) aplicado a solução obtida na iteração anterior. Em métodos iterativos, a relaxação é realizada a partir da Equação (2.5):

$$\mathbf{u}_G^{i+1} = \mathbf{u}_G^i + \omega \Delta \mathbf{u}_G^{i+1} \quad (2.5)$$

onde $\Delta \mathbf{u}_G^{i+1} = \bar{\mathbf{u}}_G^{i+1} - \mathbf{u}_G^i$, sendo $\bar{\mathbf{u}}_G^{i+1}$ - Equação (2.6) - a solução prevista inicialmente a ser corrigida, ou seja, a aproximação caso a correção não fosse aplicada:

$$\bar{\mathbf{u}}_G^{i+1} = \mathbf{u}_G^i + K_G^{-1} f(\mathbf{u}_G^i) \quad (2.6)$$

na estratégia GLI, a função $f(\mathbf{u}_G^i)$ corresponde ao vetor de resíduos, \mathbf{f}_R .

A Equação (2.5) pode ser reescrita como:

$$\mathbf{u}_G^{i+1} = \mathbf{u}_G^i + \omega(\bar{\mathbf{u}}_G^{i+1} - \mathbf{u}_G^i) \quad (2.7)$$

O fator de relaxação, ω , é a única variável a ser definida na Equação (2.7), sendo um limitador do método a necessidade de determinar um valor apropriado. Procedimentos para estimar ω em problemas aplicados ao GLI podem ser encontrados em Chevreuil *et al.* (2013), mas esses aumentam o custo computacional.

Sugere-se, portanto, que para os problemas em que o domínio local é mais rígido na discretização da escala local do que na global ($\mathbf{K}_L > \mathbf{K}_{GL}$ - Figura 2.3), os valores de ω sejam definidos no intervalo $0 < \omega < 1,0$. Nesses casos, onde é comum a divergência do método, o uso da técnica de relaxação pode garantir a convergência. Para problemas em que o domínio local é mais rígido na escala global do que na local ($\mathbf{K}_{GL} > \mathbf{K}_L$ - Figura 2.3), deve-se adotar valores de $\omega > 1,0$. Não é aconselhável utilizar valores elevados de ω , para evitar que a solução seja excedida ao longo das iterações.

Conforme destacado por Duval *et al.* (2014), a técnica de relaxação dinâmica, cuja formulação é apresentada na seção 2.1.1.2, proporciona maior efetividade ao processo de solução em comparação com a relaxação estática, mesmo quando esta utiliza o fator de relaxação ótimo.

²O prolongamento consiste em mapear cada valor do vetor resíduo local na sua posição correspondente no vetor global, preenchendo zeros nas entradas que não pertencem ao subdomínio considerado.

2.1.1.2 Relaxação dinâmica

A relaxação dinâmica utiliza uma formulação semelhante a da relaxação estática, com a diferença de que o fator de relaxação é atualizado a cada iteração. Nesse caso, a Equação (2.7) é reescrita como:

$$\mathbf{u}_G^{i+1} = \mathbf{u}_G^i + \omega^{i+1}(\bar{\mathbf{u}}_G^{i+1} - \mathbf{u}_G^i) \quad (2.8)$$

Conforme implementado por Silveira Filho (2023) no sistema *INSANE*, emprega-se a formulação do Δ^2 de Aitken proposta por Irons e Tuck (1969) para calcular o valor de ω a cada iteração. A adoção desta técnica foi recomendada por Li *et al.* (2021), que aponta os seguintes fatores positivos no contexto do uso do MEF-GLI:

1. a efetividade no aprimoramento da convergência;
2. o baixo custo computacional para o cálculo do parâmetro de relaxação;
3. a relativa simplicidade na incorporação da técnica ao algoritmo GLI.

Nesta abordagem, os vetores de incremento de deslocamentos previstos a cada iteração, calculados na interface Γ_I , podem ser expressos como:

$$\Delta^{i+1} = (\bar{\mathbf{u}}_G^{i+1} - \mathbf{u}_G^i) |_{\Gamma_I}, \quad \Delta^i = (\bar{\mathbf{u}}_G^i - \mathbf{u}_G^{i-1}) |_{\Gamma_I} \quad (2.9)$$

ω^{i+1} é o fator de relaxação referente a iteração i , definido por:

$$\omega^{i+1} = -\omega^i \frac{\Delta^i \cdot (\Delta^{i+1} - \Delta^i)}{\|\Delta^{i+1} - \Delta^i\|^2} \quad (2.10)$$

O cálculo de ω^{i+1} não implica uma oneração computacional significativa no processo de solução, uma vez que a determinação de Δ^{i+1} e Δ^i na Equação (2.9) envolve apenas os graus de liberdade na interface Γ_I , os quais correspondem a uma porção minoritária dos nós do modelo em análise. Por fim, ressalta-se que o cálculo de ω^{i+1} requer a solução das duas iterações anteriores, i e $i - 1$. Portanto, o cálculo do fator de relaxação dinâmica e, conseqüentemente, a correção da aproximação, se inicia na terceira iteração ($i = 2$), sendo $\omega^0 = \omega^1 = 1, 0$.

2.2 Método dos Elementos Finitos Generalizados - MEFG

O Método dos Elementos Finitos Generalizados pode ser entendido como uma variação do Método dos Elementos Finitos convencional, incorporando, entretanto, atributos similares aos encontrados nos métodos sem malha (Alves, 2012). De acordo com Barros (2002), a denominação atual de Método dos Elementos Finitos Generalizados foi introduzida por Melenk (1995), sendo considerados precursores da metodologia:

- o “Método dos Elementos Finitos Especiais”, posteriormente denominado como “Método da Partição da Unidade”(Babuska *et al.*, 1994), (Babuška e Melenk, 1996), (Melenk e Babuška, 1996);
- o “Método das Nuvens-*hp*”, uma formulação híbrida do MEF (Duarte e Oden, 1995).

As funções de aproximação do MEF_G são construídas por procedimentos semelhantes aos associados ao Método das Nuvens-*hp*, o que reduz a influência da malha de elementos na solução obtida. A principal diferença entre as metodologias de construção das aproximações nos referidos métodos é que, no MEF_G, o conjunto de pontos (nuvem) empregados na discretização do domínio possui uma relação de vínculo, estabelecida por meio da malha de elementos.

Dessa forma, o MEF_G caracteriza-se pelo uso de funções da Partição da Unidade (PU - funções que somam a unidade em cada ponto do domínio) enriquecidas (de maneira análoga aos métodos sem malha) sobre uma malha de elementos, valendo-se de funções especiais definidas de acordo com as especificidades do problema em questão, podendo essas não serem polinomiais. Após o enriquecimento, as funções da PU determinam as funções de forma, cuja estratégia de construção para um modelo bidimensional é ilustrada na Figura 2.4 e descrita a seguir.

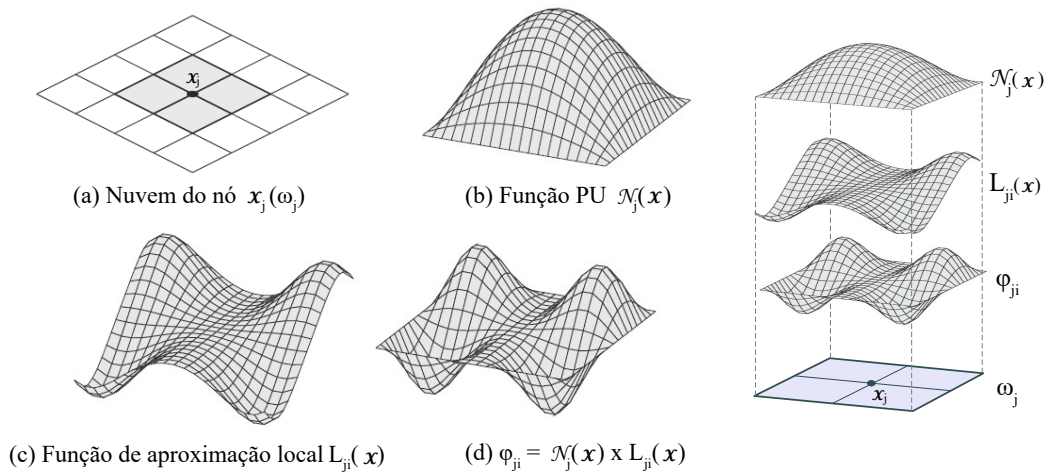


Figura 2.4: Estratégia de Enriquecimento da nuvem ω_j - (Barros, 2002).

A partir do emprego das funções convencionais do MEF, como as funções de Lagrange lineares, representadas por $\mathcal{N}_j(\mathbf{x})$ na Figura 2.4(b), obtém-se a formação da PU, uma vez que, em qualquer ponto do domínio do problema, as seguintes condições são atendidas:

$$\sum_{j=1}^n \mathcal{N}_j(\mathbf{x}) = 1 \quad (2.11)$$

Proveniente do Método das Nuvens-*hp*, o conceito de nuvem (denotado como ω_j na Figura 2.4(a)) refere-se ao conjunto de elementos finitos que compartilham o mesmo ponto nodal \mathbf{x}_j , cujas funções interpoladoras de Lagrange definem a função $\mathcal{N}_j(\mathbf{x})$. O conjunto

de funções enriquecedoras $I_j(\mathbf{x})$, também conhecidas como funções de aproximações locais (que devem representar adequadamente a solução do problema em questão), é composto por q_j funções linearmente independentes definidas para cada nó \mathbf{x}_j :

$$I_j \stackrel{def}{=} \{L_{j1}(\mathbf{x}), L_{j2}(\mathbf{x}), \dots, L_{jq_j}(\mathbf{x})\} = \{L_{ji}(\mathbf{x})\}_{i=1}^{q_j} \quad (2.12)$$

com $L_{j1}(\mathbf{x}) = 1$.

As funções de aproximação local estão representadas na Figura 2.4(c).

Concluindo o processo, as funções de forma $\phi_{ji}(x)$ do MEFG associadas ao nó \mathbf{x}_j , ilustradas na Figura 2.4(d), são definidas a partir do enriquecimento da PU proveniente da formulação convencional do MEF, obtido pelo produto das funções $\mathcal{N}_j(\mathbf{x})$ e as q_j associadas a cada nuvem ω_j :

$$\{\phi_{ji}\}_{i=1}^{q_j} = \mathcal{N}_j(\mathbf{x}) \times \{L_{ji}(x)\}_{i=1}^{q_j} \quad (2.13)$$

sem somatório em j .

Através da combinação linear das funções de forma anteriormente definidas, obtém-se a aproximação genérica $\tilde{u}(x)$ do MEFG:

$$\tilde{u}(x) = \sum_{j=1}^N \mathcal{N}_j(\mathbf{x}) \left\{ u_j + \sum_{i=2}^{q_j} L_{ji}(x) b_{ji} \right\} \quad (2.14)$$

sendo u_j e b_{ji} parâmetros nodais associados aos componentes $\mathcal{N}_j(\mathbf{x})$ do MEF e $L_{ji}(x)$ do MEFG.

A aproximação global obtida no elemento, através da combinação das funções produto referentes a cada nó, é construída sem penalizar a continuidade entre os elementos, atendendo ao critério de conformidade (Barros, 2002). Aliado a esse fator, a variedade de funções que podem ser empregadas no enriquecimento, definidas de acordo com o tipo de aplicação, proporciona diversas vantagens ao método, especialmente, em problemas que envolvem descontinuidade no campo de deslocamentos ou singularidades no campo de tensões, como aqueles causados pela presença de trincas, cuja estratégia de solução é apresentada na seção 2.3.2. Além disso, é possível enriquecer diferentes conjuntos de nós no domínio com funções distintas.

O Método dos Elementos Finitos Estendido (MEFX - (Belytschko e Black, 1999)) foi desenvolvido em paralelo ao MEFG, sendo ambas as abordagens voltadas à solução de problemas que envolvem trincas, descontinuidades e singularidades. No entanto, diferentemente do MEFG, no qual, para os primeiros trabalhos, o enriquecimento da partição da unidade é aplicado ao longo de toda a extensão do domínio, o MEFX é proposto restringindo o enriquecimento a um grupo pré-selecionado de nós. De acordo com Fries e Belytschko (2010), o MEFG e o MEFX podem ser considerados métodos numéricos equivalentes na atualidade, e portanto, referenciados como MEFG/X.

2.3 Método dos Elementos Finitos Generalizados com Enriquecimento Global-Local - MEFG^{gl}

O tratamento de problemas que apresentam solução não suave ou fenômenos concentrados em pequenas regiões do domínio frequentemente exige um refinamento excessivo da malha na formulação convencional do MEF. Para esses casos, o MEFG requer um conhecimento prévio sobre o comportamento do problema para selecionar as funções utilizadas na construção das funções de forma. Nesse contexto, foi desenvolvida uma variante do método que utiliza a solução numérica de um problema local, definido na região com singularidades, como enriquecimento para a aproximação do problema global original. Essa abordagem é denominada Método dos Elementos Finitos Generalizados com estratégia de solução Global-Local (MEFG^{gl}) (Duarte e Kim, 2008). Essa técnica tem sido amplamente empregada para problemas lineares (Kim *et al.*, 2010, Pereira *et al.*, 2012, Gupta, Kim e Duarte, 2012, Evangelista *et al.*, 2013, O'Hara *et al.*, 2016, O'Hara *et al.*, 2016, Malekan *et al.*, 2018, Malekan e Barros, 2018, Fonseca *et al.*, 2020, Santos e Barros, 2020, Li *et al.*, 2021, 2022, Pinheiro e Pitangueira, 2022) e também não lineares (Novelli *et al.*, 2020, Kim *et al.*, 2012, Kim e Duarte, 2015, Evangelista *et al.*, 2020, Monteiro *et al.*, 2020).

A solução pelo MEFG^{gl} pode ser estruturada em três etapas, conforme ilustrado na Figura 2.5 para um problema bidimensional em Mecânica da Fratura, no qual a presença de uma trinca induz elevados gradientes e descontinuidades no campo de deslocamentos. A região correspondente ao domínio local é destacada em azul, enquanto os nós globais enriquecidos com a solução do problema local aparecem em amarelo. A trinca, representada pela linha vermelha, tem caráter ilustrativo, pois sua influência é simulada apenas por meio da estratégia de enriquecimento do MEFG nos problemas local e global enriquecido.

O procedimento de solução é descrito de forma detalhada abaixo:

1. **Solução do Problema Global Inicial** - Figura (2.5a): inicialmente, o problema global é resolvido utilizando o MEF ou o MEFG, com uma discretização grosseira da malha de elementos ao longo de todo o domínio (Ω_G). Nesse estágio, não há esforço computacional significativo direcionado à representação de fenômenos locais, como a trinca. A solução aproximada do problema global inicial, $\tilde{\mathbf{u}}_G^0$, é obtida por meio do seguinte problema:

$$\text{Encontre } \tilde{\mathbf{u}}_G^0 \in \tilde{\mathcal{X}}_G^0(\Omega_G) \subset \mathcal{H}^1(\Omega_G) \forall \mathbf{v}_G^0 \in \tilde{\mathcal{X}}_G^0(\Omega_G)$$

$$\int_{\Omega_G} \boldsymbol{\sigma}(\tilde{\mathbf{u}}_G^0) : \boldsymbol{\epsilon}(\mathbf{v}_G^0) d\mathbf{x} = \int_{\partial\Omega_G^\sigma} \bar{\mathbf{t}} \cdot \mathbf{v}_G^0 ds \quad (2.15)$$

onde

- $\boldsymbol{\sigma}$ é o tensor de tensões;
- $\boldsymbol{\epsilon}$ é o tensor de deformações;

- $\bar{\mathbf{t}}$ é o vetor de tensões prescritas;
- \mathbf{v}_G^0 é a função tentativa;
- Ω_G é o domínio do modelo global;
- $\partial\Omega_G^\sigma$ é o contorno do domínio global no qual são prescritas forças;
- $\tilde{\mathcal{X}}_G^0(\Omega_G)$ é a discretização de $\mathcal{H}^1(\Omega_G)$, um espaço de Hilbert, definido em Ω_G , construído com as funções de forma do método utilizado (MEF ou MEFG) e que atende às condições de contorno homogêneas de Dirichlet definidas em $\partial\Omega_G^u$.

2. **Definição e Solução do Problema Local** - Figura (2.5b): considerando a região que contém os fenômenos locais do problema em análise, define-se um novo domínio (Ω_L), o qual é um subconjunto do domínio global ($\Omega_L \subset \Omega_G$). As condições de contorno deste problema, expressas em termos de tensões e/ou deslocamentos, são derivadas da solução obtida na análise inicial do problema global. Para o problema local, emprega-se uma malha de elementos refinada e enriquecimentos apropriados, de forma a capturar adequadamente os fenômenos responsáveis pelos gradientes elevados nos campos de tensões e deslocamentos. Sua formulação incorpora as condições de contorno provenientes do problema global ($\tilde{\mathbf{u}}_G^0$), considerando o parâmetro de rigidez $\bar{\kappa}$:

$$\text{Encontre } \tilde{\mathbf{u}}_L \in \tilde{\mathcal{X}}_L(\Omega_L) \subset \mathcal{H}^1(\Omega_L) \forall \mathbf{v}_L \in \tilde{\mathcal{X}}_L(\Omega_L)$$

$$\int_{\Omega_L} \boldsymbol{\sigma}(\tilde{\mathbf{u}}_L) : \boldsymbol{\varepsilon}(\mathbf{v}_L) d\mathbf{x} + \bar{\kappa} \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} \tilde{\mathbf{u}}_L \cdot \mathbf{v}_L ds = \int_{\partial\Omega_L \cap \partial\Omega_G^\sigma} \bar{\mathbf{t}} \cdot \mathbf{v}_L ds + \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} (\mathbf{t}(\tilde{\mathbf{u}}_G^0) + \bar{\kappa} \tilde{\mathbf{u}}_G^0) \cdot \mathbf{v}_L ds \quad (2.16)$$

onde

- Ω_L é o domínio local;
- $\partial\Omega_G$ e $\partial\Omega_L$ são os contornos dos domínios global e local, respectivamente;
- $\mathbf{t}(\tilde{\mathbf{u}}_G^0)$ é o vetor de forças prescritas da solução global;
- $\tilde{\mathbf{u}}_G^0$ é o vetor de deslocamentos prescritos da solução global;
- \mathbf{v}_L é a função tentativa;
- $\bar{\mathbf{t}}$ é o vetor de forças generalizadas prescritas;
- $\tilde{\mathcal{X}}_L(\Omega_L)$ é a discretização do $\mathcal{H}^1(\Omega_L)$, espaço de Hilbert construído com as funções de forma do MEFG no problema local, Ω_L .

Quanto às condições de contorno provenientes do problema global inicial, estas podem ser as propostas por Neumann (transferência apenas de tensões - impõe-se $\bar{\kappa} = 0$), Dirichlet (transferências apenas de deslocamentos - adota-se um valor elevado para $\bar{\kappa}$ tal como no Método da Penalidade) ou Cauchy (transferência de deslocamentos e de tensões, sendo que o parâmetro $\bar{\kappa}$ deve assumir um valor intermediário entre os correspondentes às condições de contorno anteriormente apresen-

tadas). A imposição de condições de contorno deve ocorrer apenas na parcela do contorno do domínio local que não possui interseção com o contorno do domínio global ($\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)$ - Figura 2.5). O restante da fronteira do domínio local ($\partial\Omega_L^N = \partial\Omega_L \cap \partial\Omega_G^N$ - Figura 2.5) possui condições de contorno naturais.

No que se refere às condições de contorno de Cauchy, o parâmetro $\bar{\kappa}$ pode ser determinado de forma empírica, conforme proposto por Kim *et al.* (2010), ou numericamente, seguindo a metodologia de Birner e Schweitzer (2019). Esta última é a estratégia adotada neste trabalho, conforme implementação no sistema INSANE, em que o parâmetro $\bar{\kappa}$ é determinado numericamente a cada ciclo de iteração global-local.

3. **Solução do Problema Global Enriquecido** - Figura (2.5c): Nesta etapa, a solução numérica $\tilde{\mathbf{u}}_L$, obtida a partir do problema local, é utilizada para a formulação de funções enriquecedoras associadas a partição da unidade dos nós localizados nas proximidades dos elementos contendo a trinca. Esse procedimento permite que o comportamento dos campos de deslocamentos e de tensões induzidos pela trinca sejam devidamente representados no domínio global. Tem-se, portanto, a i -ésima função de forma associada ao nó \mathbf{x}_j , definida por:

$$\phi_{ji} = \mathcal{N}_j(\mathbf{x})\tilde{\mathbf{u}}_L \quad (2.17)$$

onde $\mathcal{N}_j(\mathbf{x})$ corresponde a PU empregada no modelo global inicial.

O problema global enriquecido emprega a mesma malha do problema global inicial, sendo sua solução dada por:

$$\boxed{\text{Encontre } \tilde{\mathbf{u}}_G^E \in \tilde{\mathcal{X}}_G^E(\Omega_G) \subset \mathcal{H}^1(\Omega_G) \forall \mathbf{v}_G^E \in \tilde{\mathcal{X}}_G^E(\Omega_G)}$$

$$\int_{\Omega_G} \sigma(\tilde{\mathbf{u}}_G^E) : \varepsilon(\mathbf{v}_G^E) d\mathbf{x} = \int_{\partial\Omega_G^s} \bar{\mathbf{t}} \cdot \mathbf{v}_G^E ds \quad (2.18)$$

onde $\tilde{\mathcal{X}}_G^E(\Omega_G)$ corresponde ao espaço $\tilde{\mathcal{X}}_G^0(\Omega_G)$ acrescido hierarquicamente com as funções de enriquecimento construídas pela estratégia global-local, Equação (2.16), onde I_l é o conjunto de nós aos quais a solução local é associada:

$$\tilde{\mathcal{X}}_G^E(\Omega_G) = \tilde{\mathcal{X}}_G^0(\Omega_G) \cup \left\{ \sum_{j \in I_l} \mathcal{N}_j(\mathbf{x})\tilde{\mathbf{u}}_L \right\} \quad (2.19)$$

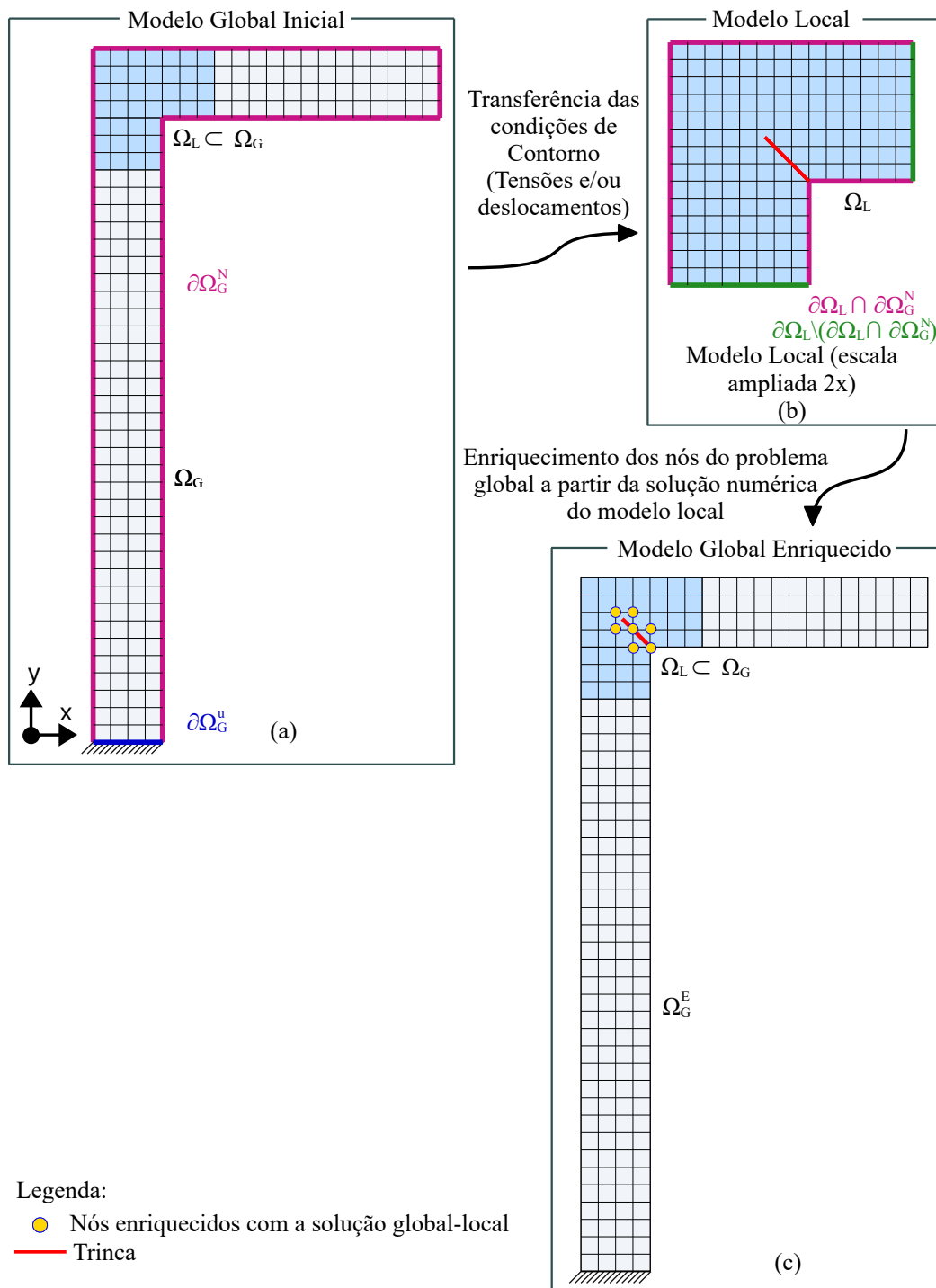


Figura 2.5: Representação esquemática do processo de enriquecimento Global-Local.

Segundo Fonseca (2019), no caso de problemas de Elasticidade Bidimensional, as funções de enriquecimento global-local adicionam apenas dois graus de liberdade a cada nó global enriquecido, independentemente da discretização utilizada no problema local. Kim *et al.* (2010) destacam esse aspecto como a principal vantagem do MEEG, uma vez que não resulta em um aumento significativo na dimensão da matriz de rigidez do problema global final.

A qualidade da solução final obtida depende da capacidade da solução numérica local em representar adequadamente o comportamento da trinca. Essa qualidade, por sua vez, é diretamente influenciada pela solução global inicial utilizada como condição de contorno no domínio local. Conforme discutido em O'Hara *et al.* (2009), erros na solução global inicial podem gerar discrepâncias significativas, mesmo em regiões afastadas dos pontos onde ocorrem os fenômenos analisados. Deste modo, a taxa de convergência do MEFG^{gl} está relacionada a convergência da solução local, que por sua vez, depende das condições de contorno impostas e da discretização adotada neste domínio. Recomenda-se, portanto, a realização de ciclos Global-Local adicionais, nos quais as condições de contorno impostas no domínio local são obtidas a partir da solução do problema global enriquecido derivado do ciclo anterior. Essa estratégia será detalhada na Seção 2.3.1.1.

De acordo com Duarte e Kim (2008), o MEFG^{gl} apresenta elevada eficiência computacional, pois o custo associado a resolução de um único sistema de equações, caso apenas um modelo devidamente discretizado fosse adotado, cresce mais rapidamente que o tamanho do problema. Em outras palavras, a decomposição do domínio em subdomínios menores, em vez de realizar uma análise refinada do problema global, revela-se mais eficiente. Somado a este aspecto, essa abordagem reduz a demanda por memória e favorece o uso de técnicas de paralelismo, conforme explorado nesta pesquisa.

2.3.1 Principais aspectos inerentes a estratégia Global-Local

A presente seção expõe as principais considerações a cerca da abordagem global-local no âmbito do Método dos Elementos Finitos Generalizados.

2.3.1.1 Ciclos Global-Local

A precisão dos resultados obtidos por meio do MEFG^{gl} é fortemente condicionada pela qualidade das condições de contorno impostas ao problema local. Conforme indicado por O'Hara *et al.* (2009), é recomendada a execução de ciclos Global-Local, nos quais a solução do problema global enriquecido é utilizada como condição de contorno em uma nova análise do problema local. Este procedimento reduz o impacto causado pela baixa precisão das informações provenientes do problema global inicial, que utiliza uma discretização grosseira e não considera explicitamente o fenômeno de interesse.

Um estudo detalhado sobre a realização de ciclos de análise Global-Local de forma iterativa, até que seja alcançada a convergência da solução, foi conduzido por Gupta, Kim e Duarte (2012). Essa abordagem foi implementada de maneira automatizada no sistema *INSANE* por Malekan (2017) e Fonseca (2019), com aplicação a problemas da Mecânica da Fratura Elástica Linear. O processo está ilustrado na Figura 2.6 e é descrito a seguir.

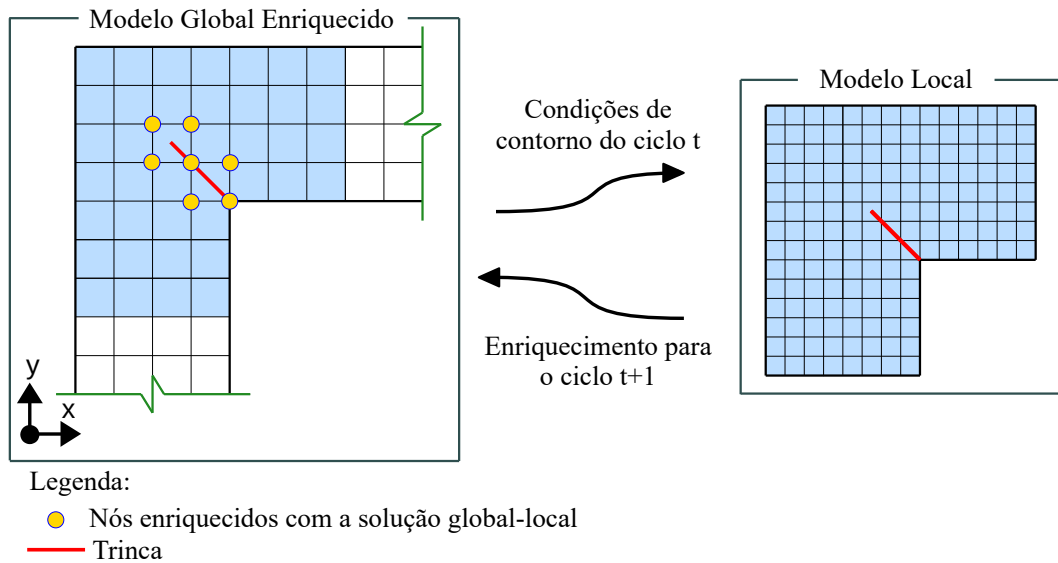


Figura 2.6: Ilustração do ciclo iterativo Global-Local.

1. Solução do problema global no ciclo t , obtendo-se $\tilde{\mathbf{u}}_G^t$. Na primeira iteração $t = 0$ e obtém-se a solução do problema global inicial, $\tilde{\mathbf{u}}_G^0$;
2. Imposição de $\tilde{\mathbf{u}}_G^t$ como condição de contorno no domínio local;
3. Solução do problema local, obtendo-se $\tilde{\mathbf{u}}_L^t$;
4. Emprego de $\tilde{\mathbf{u}}_L^t$ na construção do enriquecimento do problema global;
5. Solução do problema global enriquecido, obtendo-se $\tilde{\mathbf{u}}_G^{E,t}$;
6. Incrementa-se $t = t + 1$;
7. Inicia o processo iterativo, retornando ao passo 2 e empregando $\tilde{\mathbf{u}}_G^{E,t}$ como $\tilde{\mathbf{u}}_G^{t+1}$, a nova condição de contorno.

O número de ciclos global-local a serem realizados pode ser pré-definido, como neste trabalho, ou definido a partir de critérios de tolerância para a grandeza em análise.

2.3.1.2 Dimensão do domínio local

A dimensão do domínio local no MEF^{gl} não é intrinsecamente definida pelo método, sendo um parâmetro controlado pelo analista. Estudos sobre a influência da adição de camadas de elementos (*layers*) ao redor da trinca foram realizados por Duarte e Kim (2008), utilizando a norma de energia como parâmetro de erro. Sob a perspectiva do custo computacional, tais estudos indicam que o aprimoramento da solução decorrente do aumento do domínio local é pouco expressivo. Entretanto, Alves (2012) aponta que, em problemas de Mecânica da Fratura, a abrangência do domínio local pode impactar significativamente a solução, sendo recomendável o uso de zonas *buffer*, conforme discutido na seção seguinte.

2.3.1.3 Zona Buffer

Com o objetivo de reduzir os erros associados às condições de contorno inexatas aplicadas ao contorno do problema local, Gupta, Kim e Duarte (2012) propõem a expansão do modelo local por meio da adição de uma camada intermediária, denominada de zona *buffer*. Essa camada atua como uma região de amortecimento entre a área contendo singularidades, caracterizada por elevados gradientes de tensões e deslocamentos, e o restante do domínio onde a solução apresenta comportamento suave. A zona *buffer*, indicada em azul na Figura 2.7, compreende a região localizada entre a zona de enriquecimento e o contorno externo do domínio local. Já a área em laranja corresponde aos elementos utilizados na construção do enriquecimento global-local.

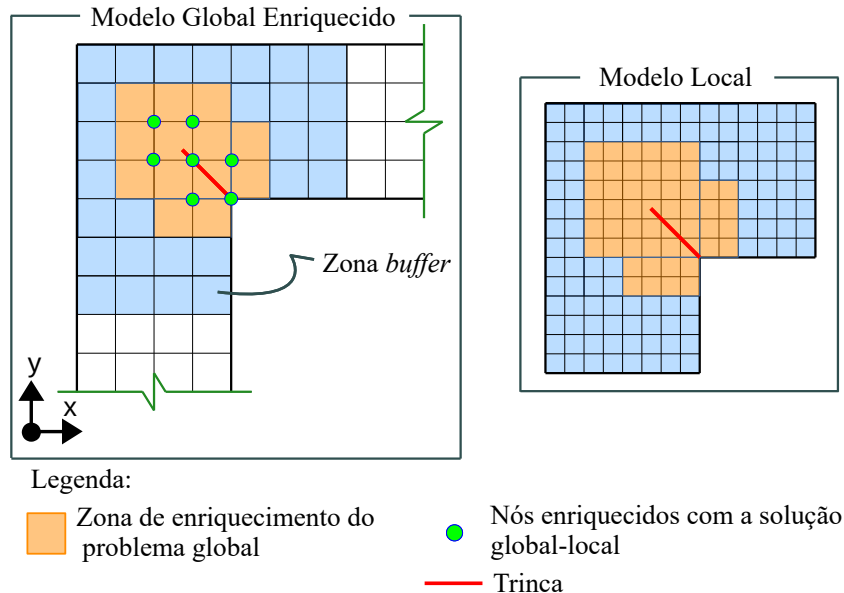


Figura 2.7: Representação esquemática da zona *buffer*.

Conforme mencionado por Fonseca (2023), a zona de enriquecimento poderia ser estendida por todo o domínio local, extinguindo a zona *buffer*. Nesse cenário, os impactos causados pela imposição de condições de contorno imprecisas seriam corrigidos por meio das iterações global-local (Seção 2.3.1.1). É importante destacar que os possíveis efeitos nocivos das condições de contorno na qualidade da aproximação estão inversamente relacionados ao tamanho da zona *buffer*, embora essa relação não seja linear. A utilização de múltiplas camadas na região de transição implica em elevados custos computacionais e não promove melhorias significativas na solução, pois acréscimos sucessivos na largura da zona geram ganhos cada vez menores em precisão, uma vez que as principais distorções já são mitigadas pela primeira camada da zona *buffer*. No presente trabalho, a zona *buffer* não foi utilizada.

2.3.2 Modelagem de trincas através do MEF^g

A aplicação do MEF^g permite, por meio do uso de funções enriquecedoras, modificar a aproximação da solução e modelar trincas sem inserir a descontinuidade geometricamente no modelo. Alves (2012) implementou no sistema *INSANE* a abordagem proposta por Moës *et al.* (1999) para problemas da Mecânica da Fratura Linear Elástica Bidimensional, que consiste na utilização de um sistema de enriquecimento composto por funções de singularidades e pela função de Heaviside.

A função de Heaviside - Equação (2.20), ou função degrau como comumente é referenciada, retorna um valor nulo quando seu argumento é negativo e assume o valor unitário quando o argumento é positivo. Dessa forma, seu uso resulta em um salto no campo de deslocamentos que representa a descontinuidade induzida pela trinca. O emprego dessas funções como enriquecimento para tratar problemas da Mecânica da Fratura foi inicialmente proposto na formulação do MEF^x (Moës *et al.* (1999)).

$$H(\xi) = \begin{cases} 1 & \forall \xi > 0 \\ 0, & \forall \xi < 0 \end{cases} \quad (2.20)$$

onde ξ é a posição relativa à descontinuidade iniciada em $\xi = 0$.

A singularidade no campo de tensões na vizinhança da ponta da trinca pode ser representada no MEF^g por meio de funções assintóticas - Equações (2.21) e (2.24) (Szabo e Babuška, 1991), ou seja, funções adequadas para tratar variáveis cujo gradiente tende ao infinito. No caso da propagação de trincas, essas singularidades correspondem aos picos de tensão na ponta da descontinuidade.

$$\mathbf{u}^{(1)} = \begin{bmatrix} u_x^{(1)} & 0 \\ 0 & u_y^{(1)} \end{bmatrix} \quad (2.21)$$

$$u_x^{(1)} = \frac{1}{2G} r^{\lambda^{(1)}} [(\kappa - Q^{(1)}(\lambda^{(1)} + 1)) \cos \lambda^{(1)}\theta - \lambda^{(1)} \cos(\lambda^{(1)} - 2)\theta] \quad (2.22)$$

$$u_y^{(1)} = \frac{1}{2G} r^{\lambda^{(1)}} [(\kappa + Q^{(1)}(\lambda^{(1)} + 1)) \sin \lambda^{(1)}\theta + \lambda^{(1)} \sin(\lambda^{(1)} - 2)\theta] \quad (2.23)$$

$$\mathbf{u}^{(2)} = \begin{bmatrix} u_x^{(2)} & 0 \\ 0 & u_y^{(2)} \end{bmatrix} \quad (2.24)$$

$$u_x^{(2)} = \frac{1}{2G} r^{\lambda^{(2)}} [(\kappa - Q^{(2)}(\lambda^{(2)} + 1)) \sin \lambda^{(2)}\theta - \lambda^{(2)} \sin(\lambda^{(2)} - 2)\theta] \quad (2.25)$$

$$u_y^{(2)} = -\frac{1}{2G} r^{\lambda^{(2)}} [(\kappa + Q^{(2)}(\lambda^{(2)} + 1)) \cos \lambda^{(2)}\theta + \lambda^{(2)} \cos(\lambda^{(2)} - 2)\theta] \quad (2.26)$$

onde:

- G é o módulo de rigidez;
- $\kappa = (3 - 4\nu)$ para o Estado Plano de Deformação, e $\kappa = \frac{(3-\nu)}{1+\nu}$ para o Estado Plano de Tensão, com ν representando o coeficiente de Poisson;
- As constantes $\lambda^{(1)} = \lambda^{(2)} = 0,5$, $Q^{(1)} = 0,333$ e $Q^{(2)} = -1,0$ foram determinadas para garantir as condições de equilíbrio e de contorno do problema, para o caso de singularidade introduzida por uma trinca fechada - $\angle\alpha = 360^\circ$ - Figura 2.8;
- r e θ são coordenadas polares relativas à ponta da trinca - Figura 2.8.

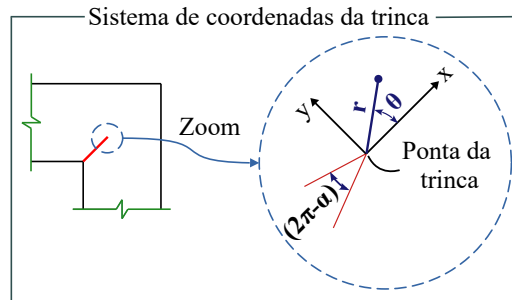


Figura 2.8: Sistema de coordenadas polares relativas à ponta da trinca

Assim, a análise de problemas da Mecânica da Fratura via MEF^{gl}, ilustrada na Figura 2.9, consiste na:

- modelagem da trinca, no problema local, por meio de combinações de funções enriquecedoras analíticas que incorporam campos descontínuos e com gradientes singulares a aproximação;
- inserção dos efeitos da trinca no problema global por meio do enriquecimento com funções construídas numericamente a partir da solução do domínio local.

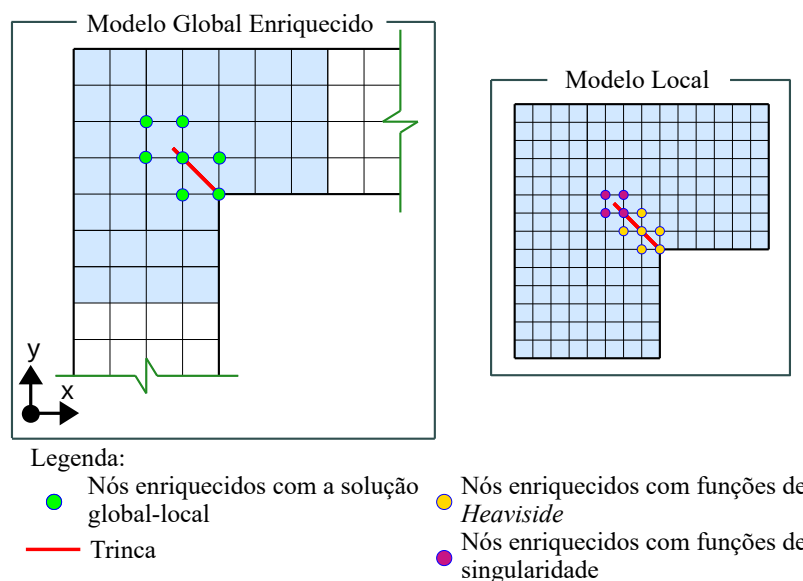


Figura 2.9: Representação esquemática da modelagem de trincas através da estratégia de enriquecimento do MEF^{gl}. A representação da trinca (linha vermelha) é apenas ilustrativa, tal como descrito na estratégia de enriquecimento do MEF^{gl} indicada na presente seção.

A formulação do MEF^{gl} aplicada na análise de trincas, conforme a estratégia apresentada nesta seção, é descrito a seguir. O processo de solução tem início com a resolução do problema global, cuja aproximação do MEF é dada por:

$$\tilde{\mathbf{u}}_g(x, y) = \sum_{j=1}^{NN_G} \mathcal{N}_j(x, y) \mathbf{u}_j^g \quad (2.27)$$

onde \mathbf{u}_j^g é o vetor dos graus de liberdade associado à partição da unidade $\mathcal{N}_j(x, y)$ para o nó j e NN_G é o número de nós no problema global. Cabe ressaltar que o emprego da aproximação do MEF em detrimento do MEF^{gl}/X não é mandatório. Neste trabalho, não foram empregados enriquecimentos no modelo global inicial objetivando compatibilizar os graus de liberdade com o modelo gerado no *ABAQUS*, conforme estratégia a ser apresentada na Seção (2.4).

No problema local, a aproximação é dada por:

$$\tilde{\mathbf{u}}_L(x, y) = \sum_{j=1}^{NN_L} \mathcal{N}_j(x, y) \mathbf{u}_j^L + \sum_{j \in I_s} \mathcal{N}_j(x, y) \{ \mathbf{u}^{(1)} \mathbf{a}_j^L + \mathbf{u}^{(2)} \mathbf{b}_j^L \} + \sum_{j \in I_d} \mathcal{N}_j(x, y) H(x, y) \mathbf{c}_j^L \quad (2.28)$$

onde \mathbf{u}_j^L é o vetor dos graus de liberdade associado à partição da unidade $\mathcal{N}_j(x, y)$ para o nó j , NN_L é o número de nós no problema local, \mathbf{a}_j^L e \mathbf{b}_j^L são os vetores dos graus de liberdade associados às funções que descrevem o campo de tensões singulares ao redor da ponta da trinca e \mathbf{c}_j^L é o vetor dos graus de liberdade associado a função de Heaviside. Deste modo, I_s e I_d denotam os conjuntos de nós onde as partições da unidade são enriquecidas com as Equações (2.21)/(2.24) e (2.20), respectivamente.

No último passo da iteração global-local, o modelo global é adicionalmente enriquecido com a solução numérica $\tilde{\mathbf{u}}_L$ obtida do problema local - Equação (2.28), resultando em:

$$\tilde{\mathbf{u}}_G^E(x, y) = \sum_{j=1}^{NN_G} \mathcal{N}_j(x, y) \mathbf{u}_j^G + \sum_{j \in I_L} \mathcal{N}_j(x, y) \tilde{\mathbf{u}}_L \mathbf{c}_j^G \quad (2.29)$$

onde \mathbf{c}_j^G é o vetor dos graus de liberdade associado as funções de enriquecimento local do nó j , e I_L é o conjunto de nós enriquecidos com a solução local.

2.4 MEF^{gl}-GLI

Tratando de problemas da mecânica da fratura no espaço tridimensional, Gupta, Pereira, J. P., Kim, Duarte e Eason (2012) apresentam um método de acoplamento não intrusivo, no qual o problema global inicial é resolvido por um programa via MEF, e o problema local é analisado por outro programa via MEF^{gl}. O problema global enrique-

cido é resolvido através da interação dos *solvers* mencionados, caracterizada pela troca de vetores de força e deslocamentos nodais, ou seja, via acoplamento não intrusivo.

Plews *et al.* (2012) aplicaram uma estratégia de solução similar para avaliar problemas de gradientes térmicos elevados. No entanto, o acoplamento entre as escalas de solução do modelo é realizado por meio da condensação estática do sistema de equações do problema global, tornando necessário o acesso à matriz de rigidez dos problemas global e local. Este aspecto limita a possibilidade de acoplamentos com programas comerciais, onde tais informações nem sempre estão disponíveis. Em problemas com grande número de graus de liberdade, o custo computacional dessa estratégia é elevado, uma vez que a determinação do vetor de forças requer o manuseio de vetores de grande dimensão, resultando em alto consumo de memória.

Filmore e Duarte (2018) propuseram um algoritmo denominado *Hierarchical Non-Intrusive Algorithm* (HNA), que realiza o acoplamento não intrusivo sem a necessidade de condensação estática do sistema de equações do problema global, desde que o programa de análise via MEF seja capaz de exportar a matriz de rigidez e o vetor de forças dessa escala do problema. Esses dados são então importados pelo programa de análise do problema local via MEFG. Esse algoritmo apresenta etapas de implementação mais simples e menor demanda computacional, mas requer que o tipo de elemento finito utilizado esteja implementado em ambos os *solvers* empregados.

Li *et al.* (2021) propuseram a estratégia MEFG^{gl}-GLI, que emprega o algoritmo iterativo global-local do MEF-GLI (Seção 2.1) com o MEFG^{gl} (Seção 2.3), exigindo apenas a transferência de forças e deslocamentos generalizados, o que elimina as limitações associadas às outras estratégias descritas nesta seção. De acordo com Duarte e Kim (2008), a propriedade hierárquica do enriquecimento global-local garante que as funções de forma convencionais do MEF permanecem inalteradas após o enriquecimento, possibilitando o acoplamento não intrusivo entre o problema global e o local (aqui denominado mesoescala).

Essa metodologia possibilita a integração de programas comerciais (usados para análise do problema inicial) com algoritmos desenvolvidos no ambiente acadêmico (utilizados para o estudo do problema local). A principal limitação deste método está relacionada ao comportamento de não convergência, uma vez que as características do GLI são herdadas (Silveira Filho, 2023).

Nas análises via MEFG^{gl}-GLI, o problema é subdividido em três escalas, sendo elas:

- **Escala Global** - Figura (2.10a): refere-se a todo o domínio, excluindo os fenômenos localizados de interesse (no caso a trinca), sendo, portanto, discretizado por meio de uma malha grosseira;
- **Mesoescala** - Figura (2.10b): refere-se a uma região específica contida na escala global, com a função de possibilitar o acoplamento não intrusivo entre as escalas global e local, atuando, portanto, como um modelo de transição [correspondendo

à escala local do algoritmo GLI (Seção 2.1) e à escala global do MEF G^{gl} (Seção 2.3)]. Comumente, utiliza-se a mesma discretização da malha do modelo global na mesoescala, dispensando a aplicação de técnicas de conformação entre as malhas desses domínios, o que facilita a transferência de informações entre os modelos global e local, além de eliminar uma das principais limitações do MEF-GLI (Whitcomb, 1991). Em geral, fenômenos localizados não são representados na mesoescala, mas não há um limitante para tal consideração (Li *et al.*, 2021);

- **Escala Local** - Figura (2.10c): refere-se às regiões do domínio onde estão representados os fenômenos locais de interesse e outras características estudadas no problema, as quais são discretizadas por meio de uma malha refinada.

Portanto, conforme pode ser visualizado na Figura 2.10, a escala global será acoplada através do algoritmo GLI à mesoescala que por sua vez, é enriquecida com a solução da escala local via MEF G^{gl} .

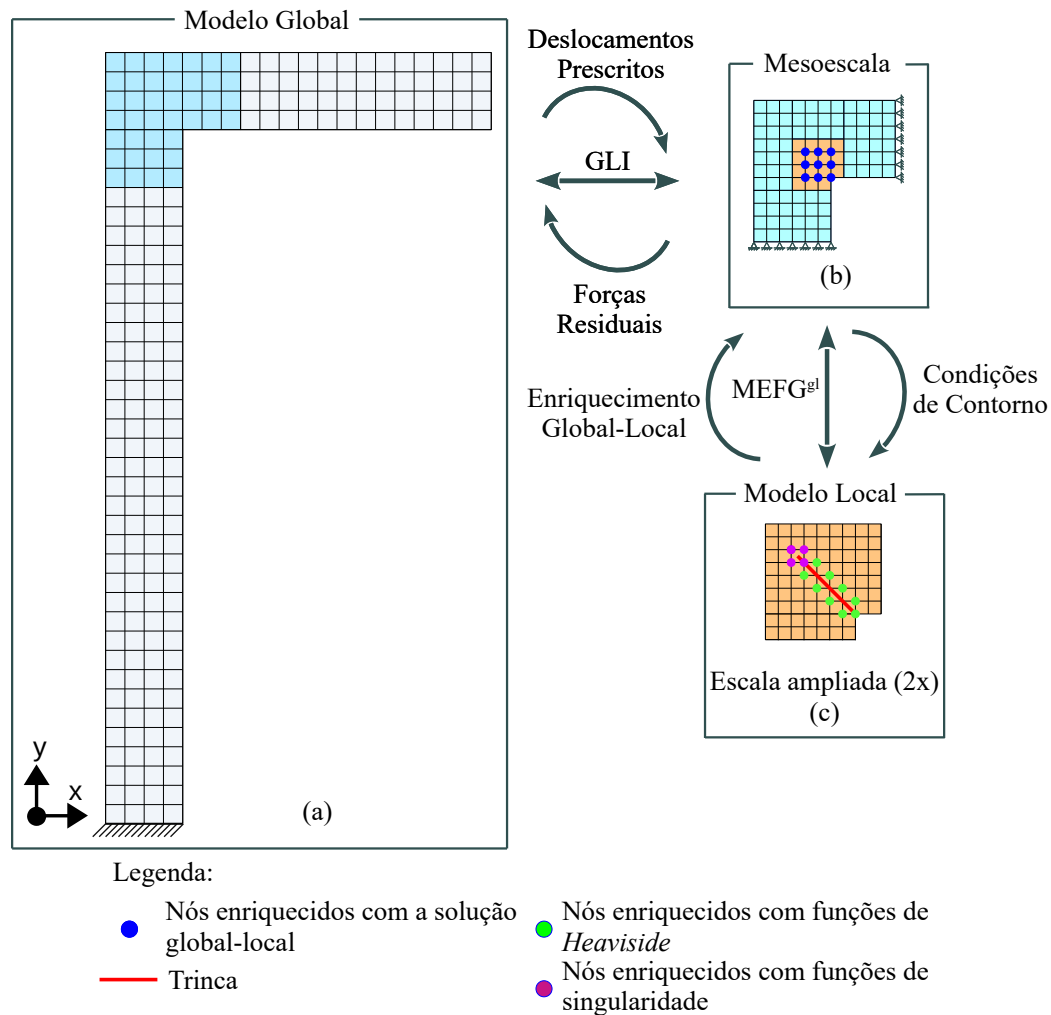


Figura 2.10: Representação esquemática da estratégia MEF G^{gl} -GLI.

Em função da estratégia de acoplamento entre o algoritmo GLI e o MEF G^{gl} , foram propostas duas abordagens de solução para o MEF G^{gl} -GLI: a monolítica e a *staggered*,

conforme detalhado nas seções 2.4.1 e 2.4.2, respectivamente.

2.4.1 Abordagem monolítica

A abordagem monolítica é a versão inicial do MEF^{gl}-GLI proposta por Li *et al.* (2021). Sua metodologia de solução consiste em realizar ao menos um ciclo global-local a cada iteração GLI, ou seja, em todas as iterações, a mesoescala é novamente enriquecida com a solução do problema local. Esta abordagem resulta em soluções mais estáveis e precisas, evitando a necessidade de iterações entre subproblemas (acoplamento exclusivo entre o problema global e a mesoescala, excluindo o problema local). As etapas do processo de solução são descritas abaixo e esquematizadas no fluxograma da Figura 2.11, onde k refere-se a cada iteração GLI.

1. Análise do modelo global: resolver o modelo global e obter a solução \mathbf{u}_G^k ;
2. Análise da(s) mesoescala(s): definir e resolver os modelos das mesoescalas impondo como condição de contorno a solução do modelo global (\mathbf{u}_G^k) - Equação (2.27). Obtêm-se a solução \mathbf{u}_M^k ;
3. Análise do modelo local: definir e resolver os modelos locais empregando a solução do modelo de mesoescala (\mathbf{u}_M^k) como condição de contorno (Neumann, Dirichlet ou Cauchy) - Equação (2.28). Obter a solução \mathbf{u}_L . Destaca-se que é possível definir múltiplos domínios locais para uma mesma mesoescala;
4. Análise da mesoescala enriquecida: aplicar o enriquecimento construído a partir da solução do problema local na mesoescala e resolver o referido domínio, obtendo \mathbf{u}_{ME}^k - Equação (2.29). Em sua apresentação original, os ciclos global-local do MEF^{gl} descritos nesta etapa e na 3 ocorreram apenas uma vez. Porém, pode-se determinar o número de iterações necessárias para aprimorar a qualidade das condições de contorno e atingir a convergência almejada;
5. Cálculo do vetor de forças residuais: calcular o vetor resíduo, \mathbf{f}_R^k , conforme Equação (2.3), empregando os valores de \mathbf{u}_G^k e \mathbf{u}_{ME}^k na interface entre a mesoescala e o modelo global;
6. Convergência: verificar se a convergência definida foi estabelecida encerrando o processo. Neste caso, têm-se como solução final \mathbf{u}_{ME}^k na região da mesoescala e \mathbf{u}_G^k no restante do domínio;
7. Iteração: em caso de não ser atingida a convergência, atualizar o vetor de forças do modelo global somando-se com sinal contrário as forças residuais, \mathbf{f}_R^k . Define-se $k = k + 1$ e retorna ao passo 1 iterativamente até que se obtenha a convergência.

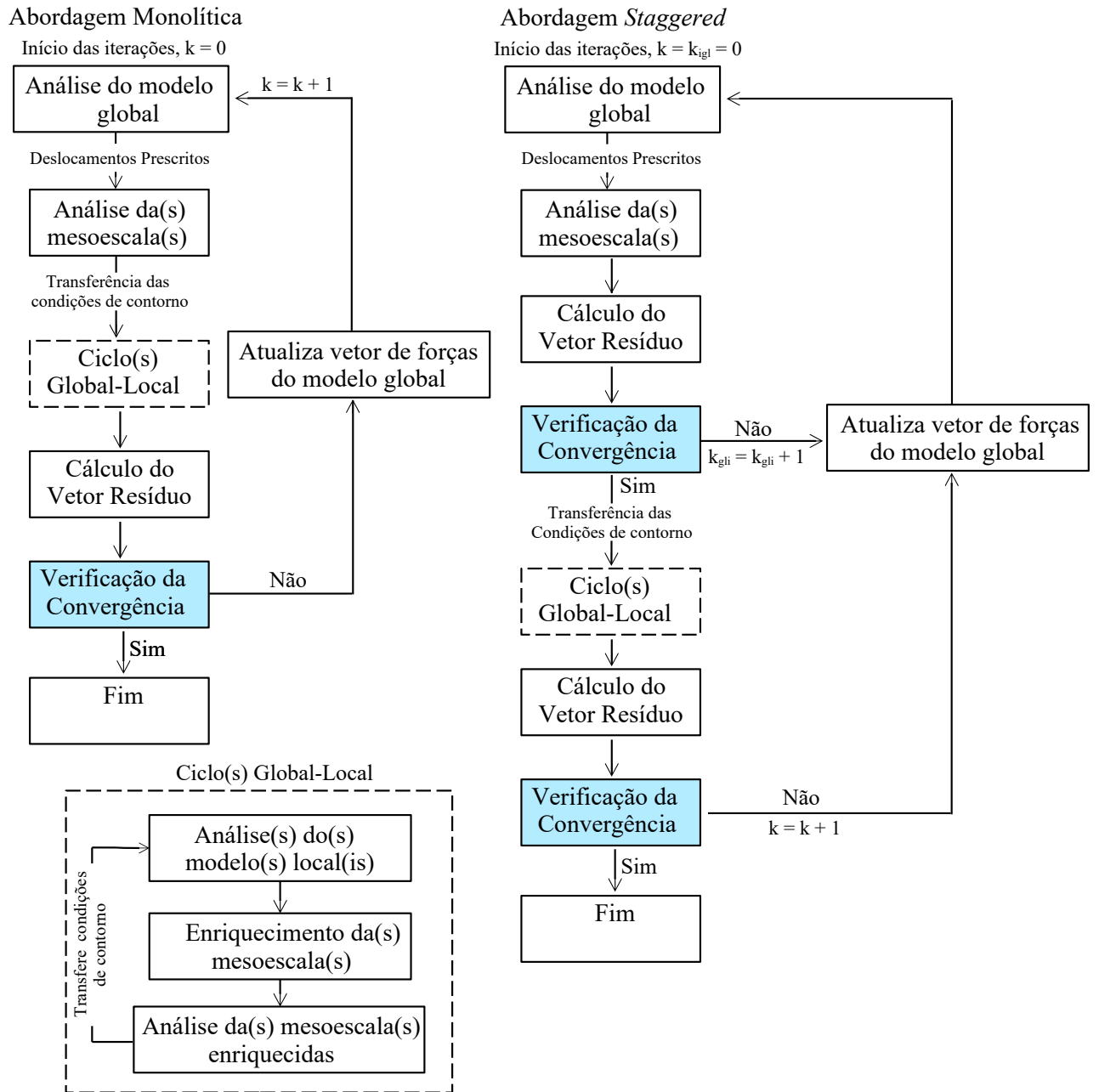


Figura 2.11: Fluxograma dos principais passos da estratégia $\text{MEEG}^{gl}\text{-GLI}$ segundo abordagem monolítica e *staggered*. Adaptado de (Bueno, 2024)

2.4.2 Abordagem *staggered*

Li *et al.* (2022) propuseram uma variante ao método MEFG^{gl}-GLI por meio de um algoritmo denominado *staggered*. O diferencial desta nova estratégia reside na execução de iterações GLI (Whitcomb, 1991) entre o modelo global e a mesoescala antes de acoplar o MEFG^{gl} a análise. Dessa maneira, no processo de solução proposto por Li *et al.* (2021), as etapas 3 e 4 são realizadas apenas após a convergência dos ciclos GLI. Somente quando se atinge o equilíbrio entre os problemas da mesoescala e da escala global, as etapas correspondentes ao MEFG^{gl} são realizadas. Após essa etapa, novas iterações GLI são realizados até que a convergência final seja alcançada. Dessa forma, ocorre uma redução nas simulações MEFG^{gl}, minimizando as possíveis alterações de rigidez da mesoescala provocadas pelo enriquecimento global-local.

Nesta abordagem são utilizados dois iteradores no gerenciamento das etapas de cálculo, k e k_{gli} , que estão associados, respectivamente, às iterações do algoritmo GLI entre o problema global e a mesoescala, e à iteração completa do método, incluindo a análise global-local. O processo de solução é descrito abaixo e esquematizado no fluxograma da Figura 2.11.

1. Início do processo iterativo: $k = 0 = k_{gli} = 0$;
2. Análise do modelo global: resolver o modelo global e obter a solução \mathbf{u}_G^k ;
3. Análise da(s) mesoescala(s): definir e resolver os modelos das mesoescalas impondo como condição de contorno a solução do modelo global (\mathbf{u}_G^k) - Equação (2.27). Obtêm-se a solução \mathbf{u}_M^k ;
4. Cálculo do vetor de forças residuais: calcular o vetor resíduo, \mathbf{f}_R^k , conforme Equação (2.3), empregando os valores de \mathbf{u}_G^k e \mathbf{u}_M^k (iteração $k = 0$) ou \mathbf{u}_{ME}^k (iteração $k \neq 0$) na interface entre a mesoescala e o modelo global;
5. Convergência do algoritmo GLI: verificar se a convergência definida para o GLI foi atingida, neste caso a análise passa para a etapa 7, e ocorre a execução dos ciclos global-local do MEFG^{gl};
6. Iteração: em caso de não ser atingida a convergência, atualizar o vetor de forças do modelo global somando-se com sinal contrário as forças residuais, \mathbf{f}_R^k . Define-se $k_{gli} = k_{gli} + 1$ e retorna ao passo 1 realizando iteração até que se obtenha a convergência;
7. Análise do modelo local: definir e resolver os modelos locais empregando a solução do modelo de mesoescala convergido (\mathbf{u}_M^k ou \mathbf{u}_{ME}^k) como condição de contorno - Equação (2.28). Obter a solução \mathbf{u}_L ;
8. Análise da mesoescala enriquecida: aplicar o enriquecimento construído a partir da solução do problema local na mesoescala e resolver novamente o referido domínio, obtendo \mathbf{u}_{ME}^k - Equação (2.29);
9. Cálculo do vetor de forças residuais: calcular o vetor resíduo, \mathbf{f}_R^k , conforme Equação

- (2.3), empregando os valores de \mathbf{u}_G^k e \mathbf{u}_{ME}^k na interface entre a mesoescala e o modelo global;
10. Convergência do método: verificar se a convergência definida foi estabelecida encerrando o processo. Neste caso, têm-se como solução final \mathbf{u}_{ME}^k na região da mesoescala e \mathbf{u}_G^k no restante do domínio;
 11. Iteração: em caso de não ser atingida a convergência, atualizar o vetor de forças do modelo global inicial somando-se com sinal contrário as forças residuais, \mathbf{f}_R^k . Define-se $k = k + 1$ e retorna ao passo 1 realizando iterações até que se obtenha a convergência.

Conforme apresentado na seção 2.1, a convergência da estratégia GLI é fortemente influenciada pela variação de rigidez na região do modelo local (correspondente à mesoescala no MEF^{gl}-GLI) entre os domínios de análise (global e local). Assim, a adoção do algoritmo *staggered* pode resultar em uma redução do custo computacional. No entanto, dependendo das condições do problema analisado, pode haver um aumento no número de iterações necessárias para alcançar a convergência, visto que a não inclusão do domínio local em todas as iterações pode afastar o modelo global da solução correta. Segundo Li *et al.* (2022), ambas as abordagens convergem e apresentam níveis de precisão idênticos.

Nos modelos analisados por Bueno (2024), a abordagem monolítica demonstrou maior eficiência computacional em comparação ao algoritmo *staggered*. Contudo, em cenários que consideraram o aumento dos custos computacionais na solução do modelo da escala local, como maior refinamento da malha e ampliação da nuvem de enriquecimento, o desempenho do algoritmo *staggered* apresentou menor sensibilidade a tais alterações.

2.5 Computação Paralela

A computação paralela consiste em uma estratégia de processamento no qual múltiplas tarefas são executadas simultaneamente em vários servidores ou processadores de computador. A resolução de um problema via computação paralela pode ocorrer por meio da Computação de Alto Desempenho (computação massivamente paralela, utilizando dezenas de milhares de processadores - Seção 2.5.1) ou pelo emprego de computadores pessoais com múltiplos núcleos de processamento (paralelismo local - Seção 2.5.2). As taxas de ganho de *performance* e o aumento da frequência de operação dos processadores têm diminuído em comparação aos níveis observados na década de 1990. Assim, a demanda por processadores mais rápidos pode ser suprida através do uso de processadores paralelos, que particionam a demanda computacional.

O paralelismo pode ser dividido em níveis, com base na granularidade e no tipo de operações a serem executadas, da seguinte maneira:

- **Nível de Instrução - ILP (*Instruction-level Parallelism*)**: as instruções executadas por um processador são reordenadas e combinadas em grupos, que são

executados em paralelo sem interferir no fluxo ou no resultado do programa (Culler *et al.*, 1998). As técnicas mais comuns para implementar e gerenciar o paralelismo de instruções são os algoritmos nomeados *marcador* e *Tomasulo*;

- **Nível de Dados - DLP (*Data-level Parallelism*)**: refere-se ao gerenciamento e manipulação de dados em operações paralelas baseadas nas mesmas instruções. O paralelismo de dados é frequentemente inerente a laços de repetição;
- **Nível de Tarefas - TLP (*Task-level Parallelism*)**: corresponde a execução simultânea de tarefas independentes entre si. As operações relacionadas a uma tarefa podem ser realizadas no mesmo ou em diferentes conjuntos de dados;
- **Nível de Processos - NLP (*Node-level Parallelism*)**: refere-se a execução simultânea de múltiplos processos independentes. Ao contrário das tarefas no TLP, cada processo possui seu próprio espaço de memória e recursos alocados, minimizando interferências entre eles.

2.5.1 Computação de Alto Desempenho

A Computação de Alto Desempenho (HPC - *High-performance computing*) refere-se à prática de agregar poder de computação para processar grandes conjuntos de dados e resolver problemas complexos com velocidades extremamente altas. Para tal, essa tecnologia faz uso de múltiplos dispositivos de armazenamento e processadores potentes que operam em paralelo. Originalmente, as cargas de trabalho de HPC eram destinadas a supercomputadores, que incorporavam milhares de processadores. Na atualidade, as tarefas são majoritariamente executadas em *clusters* de servidores de computadores de alta velocidade, usualmente hospedados na nuvem.

Um *cluster* pode ser entendido como uma arquitetura de rede que conecta vários servidores de computadores, denominados nós, e possui um agendador centralizado, cuja função primordial é gerenciar a carga de trabalho computacional entre os nós do sistema de forma paralela. Cada computador é equipado com CPUs (*Central Processing Unit*, em português Unidade Central de Processamento) multi-core ou GPUs (*Graphics Processing Units*, em português Unidade Gráfica de Processamento) de alto desempenho. A primeira utilização dessa arquitetura computacional remonta a 1994, com o projeto desenvolvido pela NASA por Donald Becker e Thomas Sterling, intitulado *Beowulf*. O objetivo principal era utilizar um conjunto de *hardwares* comerciais e *softwares* de código aberto para construir um sistema de computação de alto desempenho a baixo custo.

Portanto, a HPC pode ser vista como uma estratégia de paralelismo distribuído a nível de processos (NLP), cujo paradigma de programação envolve o uso de várias máquinas. Nessas situações, a interface de comunicação entre os processos executados por cada máquina deve empregar uma base de mensagens decodificáveis pelo sistema como um todo, como o padrão MPI (*Message Passing Interface*, em português Interface de Passa-

gem de Mensagens).

2.5.2 Paralelismo Local

O paralelismo local emprega os recursos disponíveis de um único computador, como seus múltiplos núcleos de CPU ou GPUs, no processamento simultâneo de tarefas, aqui denominadas *threads*. Um núcleo de processamento é composto por um conjunto de circuitos e transistores que realizam operações matemáticas, lógicas e de controle de dados a partir da interpretação e execução de instruções em sequência. Dessa forma, um núcleo pode processar uma tarefa (*thread*) por vez.

As *threads* de execução são tarefas concorrentes executadas por um processo, e compartilham os recursos e a memória deste. Essa característica elimina a necessidade de troca de mensagens para comunicação (as trocas ocorrem diretamente na memória), tornando o processo mais rápido, embora limite o potencial de paralelismo aos recursos físicos da máquina, como número de núcleos e capacidade de RAM. No contexto do paralelismo local na linguagem de programação *JAVA*, cada instância da JVM (*JAVA Virtual Machine*, em português Máquina Virtual *JAVA*) corresponde a um processo do sistema operacional e implementa *threads* internamente. Dessa forma, o método `main()` é executado na *thread* principal, e as demais *threads* secundárias podem ser criadas a partir desta. Através de mecanismos de sincronização e prioridade, é possível controlar a ordem de execução das tarefas, com ou sem relação de dependência entre elas. A abordagem de paralelismo via *multithreading* pode abranger os níveis DLP e TLP, sendo suas funções nativas do *JAVA*.

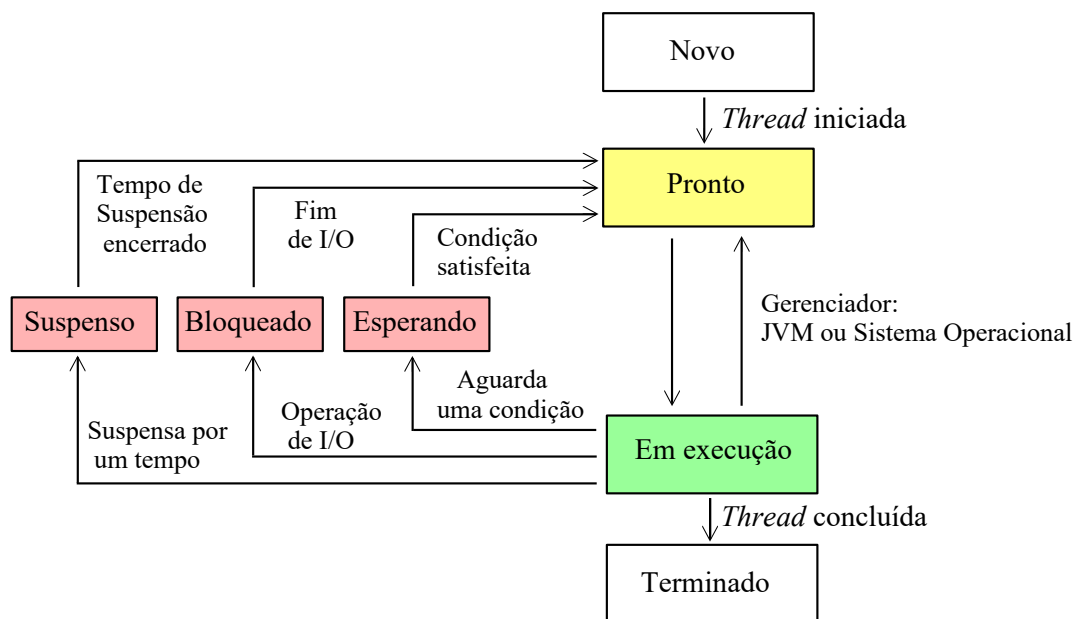


Figura 2.12: Fluxograma dos Estados do Ciclo de vida de *threads* em *JAVA*

As *threads* podem ser criadas por meio do mecanismo de herança, estendendo a classe

`Thread`, nativa do JAVA, e sobrescrevendo o método `run()`. Outra possibilidade é implementar a interface `Runnable` na classe que contém as tarefas a serem paralelizadas e passar uma instância dessa classe para um objeto `Thread`. Conforme o gerenciamento da JVM, o ciclo de vida de uma *thread* possui os estados ilustrados no fluxograma da Figura 2.12 e descritos a seguir:

- Novo: a *thread* foi criada, mas não foi iniciada;
- Pronto: a *thread* foi iniciada, está pronta para ser executada e ainda não foi alocada para um processador pelo sistema operacional, podendo ser escalonada pelo gerenciador de tarefas;
- Em execução: a *thread* é alocada para um processador e está sendo executada;
- Bloqueado: situação em que uma *thread* não pode continuar sua execução devido à necessidade de algum recurso ou condição a ser satisfeita, comumente associada a processos de entrada e saída de dados - I/O (*Input/Output*). Esse estado ocorre também quando a *thread* depende da conclusão de outra *thread* ou necessita acessar um recurso sincronizado que já está sendo utilizado, tendo que aguardar a liberação do bloqueio. O mecanismo que assegura o acesso exclusivo a um determinado recurso ou banco de dados é denominado exclusão mútua;
- Esperando: corresponde aos casos em que uma *thread* é explicitamente suspensa por outra *thread*, geralmente por meio do método `wait()`, e em sequência volta a ser executada;
- Suspenso: estado no qual uma *thread* é suspensa por um período de tempo específico ou aguarda por um tempo determinado antes de retornar. Tal ação é feita através dos métodos `sleep(long millis)` ou `join(long millis)`;
- Terminado: ocorre quando a execução da tarefa é concluída ou interrompida. Neste estado, a *thread* não pode ser reiniciada, sendo portanto, destruída.

2.5.3 Medidas de desempenho

De acordo com Azevedo (2019), a performance da execução de tarefas de forma paralela em comparação com a abordagem sequencial pode ser avaliada por meio de parâmetros de desempenho, como o número de instruções executadas, o tempo de execução, o uso de memória e a transferência de dados. Uma medida de desempenho para a aplicação paralela leva em consideração o efeito conjunto do processamento, armazenamento de dados e comunicação, sendo a métrica mais usual o *speedUp*, $SU(p)$ - Equação 2.30 (Eager *et al.*, 1989). Esse parâmetro é calculado pela razão entre o melhor tempo obtido com o processamento sequencial, T_s , e o melhor tempo da versão com n_p processadores, T_p :

$$SU(n_p) = \frac{T_s}{T_p} \quad (2.30)$$

Em programas com tarefas e processos, teoricamente, totalmente paralelizáveis, o

speedUp teria um valor igual ao número de processadores, n_p , uma vez que a carga de trabalho seria perfeitamente dividida entre estes. No entanto, esse cenário raramente é alcançado, pois, na maioria dos casos, existem trechos de código de encadeamento sequencial. Nos casos em que $SU(n_p) > 1$, a versão paralela resulta em uma redução no tempo de execução, enquanto que quando $SU(n_p) < 1$, o processamento se torna mais lento do que na abordagem sequencial. Cada aplicação possui um número ideal de unidades ativas para otimizar o desempenho em uma dada arquitetura, e não há uma relação linear entre o número de processadores e a redução do tempo de execução.

A eficiência da paralelização indica a taxa de utilização média das unidades ativas, $E(n_p)$ - Equação (2.31) (Eager *et al.*, 1989). Este parâmetro mostra se os recursos foram bem aproveitados, sendo calculado pela razão entre o *speedUp* e o número de processadores utilizadas, n_p :

$$E(n_p) = \frac{SU(n_p)}{n_p} \quad (2.31)$$

A Equação (2.31) assume que todos os n_p processadores têm capacidade equivalente. Contudo, em diversos cenários, as unidades ativas permanecem por parte do tempo aguardando os resultados de outras tarefas, o que resulta em uma redução na taxa de eficiência. A eficiência ideal corresponde a $E(n_p) = 1,0$, ou seja, cada processador permanece ativo 100% do tempo. Vale ressaltar que a melhor taxa de eficiência não implica no menor tempo de execução, ou seja, $n_{p\text{ótimo}}(E) \neq n_{p\text{ótimo}}(SU)$.

Para avaliar o desempenho da paralelização, considerando a parte do código sequencial remanescente, pode-se aplicar a Lei de Amdahl - Equação (2.32) (Amdahl, 1967). Sendo S a porcentagem do código mantido sequencial e P a porcentagem suscetível à paralelização, o *speedUp* é dado por:

$$SU_{amd}(p_n) = \frac{1}{S + \frac{P}{n_p}} \quad (2.32)$$

Embora as equações descritas acima sejam simples, a complexidade reside na métrica das variáveis envolvidas, como as porcentagens do código sequencial e paralelizado, assim como a medição do tempo. No que diz respeito ao tempo, Azevedo (2019) sugere que, para minimizar as perturbações nos resultados causadas pelo efeito observador, deve-se requisitar a duração da aplicação diretamente do sistema ou utilizar ferramentas de *profiling*, que realizam a coleta automática de dados do programa em intervalos regulares ou não.

2.5.4 Paralelização no âmbito do Método dos Elementos Finitos

O MEF utiliza equações lineares algébricas para o tratamento numérico de equações diferenciais, sendo estas organizadas em formato matricial nas implementações computa-

cionais. Técnicas de paralelização podem ser aplicadas em várias etapas da solução de um problema via MEF, como na montagem da matriz de rigidez e na resolução do sistema de equações. Diversos procedimentos podem ser utilizados, como a decomposição de domínios, que consiste na divisão das equações diferenciais parciais em problemas menores, mas acoplados. No contexto da incorporação de estratégias de paralelismo no sistema *INSANE*, destaca-se o estudo de Azevedo (2019) sobre a implementação paralela do MEF utilizando um “cluster”.

Considerando a análise multi-escala proposta pela estratégia $MEFG^{gl}$ -GLI, modelos complexos são subdivididos em sub-regiões que podem ser resolvidas de forma independente, permitindo a aplicação de estratégias de paralelismo. Essa estratégia reduz a demanda de memória e tempo computacional, porém, para garantir a eficiência desejada, é necessário distribuir a carga de trabalho de maneira equilibrada entre os processadores e minimizar o tempo gasto na transmissão de informações.

Kim *et al.* (2011) abordam a implementação e os desafios de balanceamento de carga para a versão paralelizada do $MEFG^{gl}$, apresentando uma técnica de ordenação para melhorar a eficiência do algoritmo e ferramentas para estimar o número máximo de CPUs capazes de uniformizar a distribuição de carga entre os processadores. A principal vantagem do $MEFG^{gl}$ paralelizado é a ausência de necessidade de comunicação entre processadores, uma vez que todas as informações necessárias para resolver cada problema local são obtidas do problema global inicial antes da análise paralela. Após o processamento de cada problema local, as soluções são enviadas ao problema global por meio da estratégia de enriquecimento. Nesta etapa, a influência entre os domínios locais é incorporada ao problema, inicialmente em escala global, e posteriormente em escala local por meio da execução de ciclos de análise.

Dessa forma, a estratégia proposta por Kim *et al.* (2011) envolve apenas duas comunicações do tipo *scatter-gather*. A comunicação *scatter* refere-se a um padrão de troca de informações no qual um processo (ou nó) envia dados para vários outros processos ou nós, neste caso, correspondendo ao modelo global que fornece informações (condições de contorno) para todos os domínios locais. Por outro lado, a comunicação *gather* corresponde ao conceito oposto da comunicação *scatter*, referindo-se ao padrão de troca de informações em que dados de vários processos (ou nós) são reunidos em um único processo central. Na análise global-local, tal comunicação corresponde ao enriquecimento do domínio global com as funções construídas a partir das soluções dos múltiplos domínios locais.

A implementação paralela do $MEFG^{gl}$ em plataformas de memória compartilhada (adoção de paralelismo local) segue os seguintes passos para cada ciclo global-local (Kim *et al.*, 2011):

1. Resolver o problema global inicial;
2. Criar problemas locais e armazená-los em uma lista de tarefas a serem executadas;
3. Ordenar os problemas locais na lista com base em estimativas de sua carga compu-

tacional;

4. Ativar os n_p (valor definido pelo usuário) processadores a serem empregados na análise;
5. Selecionar o primeiro problema local da lista e atribuí-lo a um único processador para resolução;
6. Executar a resolução do problema local selecionado em paralelo. Esta é a etapa onde ocorre o maior consumo de tempo de CPU;
7. Caso ainda existam problemas locais não resolvidos na lista, retornar aos passos 5 e 6 e repetir o processo. Caso contrário, finalizar a solução paralela dos problemas locais.;
8. Resolver o problema global enriquecido.

A eficiência computacional deste algoritmo é proporcionada pelo fato de nenhum processador ficar ocioso até que todos os problemas locais sejam resolvidos. O custo computacional associado ao processamento dos problemas locais pode variar em função do grau de complexidade do modelo, como refinamentos de malha, enriquecimentos e a extensão do domínio. Em problemas de mecânica da fratura, as condições de propagação de trincas podem aumentar o número de graus de liberdade e pontos de integração, resultando em desequilíbrio de carga entre processadores e, conseqüentemente, em uma redução da eficiência paralela. Para mitigar esse efeito, Kim *et al.* (2011) sugerem ordenar os problemas locais com base em suas cargas computacionais estimadas, indicando que a adoção de uma ordem decrescente de carga minimiza o desequilíbrio, enquanto a ordenação crescente agrava o problema. Os autores estabelecem como condição para o equilíbrio de carga entre os processadores a desigualdade dada pela Equação 2.33:

$$\frac{\sum_{i=1}^{N_s} Q_i}{n_p} \geq Q_{max} \quad (2.33)$$

onde N_s é o número de problemas locais, n_p é o número de processadores, Q_i é a carga computacional do i -ésimo problema local e Q_{max} é a carga do maior problema local.

Reordenando a Equação (2.33), pode-se determinar o maior número de CPUs em que o equilíbrio de carga pode ser alcançado, resultando na Equação (2.34):

$$n_p^{max} = \frac{\sum_{i=1}^{N_s} Q_i}{Q_{max}} \quad (2.34)$$

Os autores sugerem que o número de operações de ponto flutuante necessárias para fatorar a matriz de rigidez de um problema local seja uma boa estimativa para sua carga computacional, uma vez que tal operação é, frequentemente, a etapa mais demorada de uma análise via elementos finitos. Ressalta-se que outros fatores também impactam a eficiência do paralelismo na estratégia global-local, como o custo inicial de ativação de multiprocessadores e o custo de reativação de processadores.

CAPÍTULO 3

ASPECTOS COMPUTACIONAIS

Neste capítulo, são apresentados os principais aspectos da implementação computacional desenvolvida neste trabalho por meio do sistema *INSANE* e do programa *ABAQUS*. Apresenta-se uma descrição detalhada e estruturada das novas funcionalidades implementadas no sistema, organizadas conforme a seguinte categorização:

1. Apresentação dos algoritmos de solução das abordagens monolítica e *staggered* do MEFG^{gl} -GLI aplicado a problemas com múltiplas mesoescalas;
2. Aprimoramento dos algoritmos em função da taxa de convergência de cada domínio local na convergência global do problema;
3. Implementação dos métodos de leitura, gerenciamento e exportação dos parâmetros associados à análise, abrangendo tanto os dados de entrada (*input*) quanto os resultados gerados (*output*);
4. Implementação paralelizada dos algoritmos de solução da estratégia de acoplamento MEFG^{gl} -GLI em análises com múltiplas mesoescalas.

3.1 Algoritmos de solução do MEFG^{gl} -GLI aplicado a problemas com múltiplas mesoescalas

O procedimento de solução via MEFG^{gl} -GLI foi apresentado nas seções (2.4.1) e (2.4.2), referente as abordagens monolítica e *staggered*, respectivamente. Conforme implementado por Bueno (2024), as técnicas de relaxação do GLI e o algoritmo gerenciador do acoplamento entre as escalas do problema estão integrados ao sistema *INSANE*, também responsável pela solução do modelo da escala local via MEFG^{gl} .

Para a solução do modelo na escala global, emprega-se o programa comercial *Abaqus*, que utiliza o MEF e é acionado por *scripts* desenvolvidos em Python, conforme descrito em Silveira Filho (2023). A execução desses *scripts* no ambiente *INSANE* é viabilizada pelo uso do Jython (Jython, 2022), uma linguagem que possibilita a integração de códigos Java e Python, assegurando a interoperabilidade entre ambas linguagens por meio de aplicações híbridas.

São apresentados a seguir os algoritmos de solução monolítica e *staggered* segundo

a implementação proposta neste trabalho para a estratégia não intrusiva MEFG^{gl}-GLI aplicada a problemas com múltiplas mesoescalas.

3.1.1 Algoritmo de solução monolítica

A abordagem de solução monolítica aplicada à estratégia MEFG^{gl}-GLI é detalhada no Algoritmo 3.1. Essa abordagem caracteriza-se por resolver, a cada iteração, as equações governantes de todas as escalas envolvidas no problema, estabelecendo o acoplamento entre estas. A eficiência/convergência do acoplamento depende ainda dos parâmetros específicos do problema e do controle rigoroso das interações entre os modelos globais e locais.

Algoritmo 3.1: Abordagem monolítica para o MEFG^{gl}-GLI aplicada a problemas com múltiplas mesoescalas.

Entrada:

- Modelo numérico referente a escala global;
- Tipo de técnica de relaxação. Em análises com emprego de relaxação estática, definir o coeficiente de relaxação - ω ;
- Tolerância do GLI (ϵ).

Saída:

- Solução final do modelo da escala global;
- Solução final do modelo das n mesoescalas enriquecidas.

```

1   $i \leftarrow 0$ ;
2  Calcular a solução inicial do modelo global :  $\mathbf{u}_G^0$ ;
3  enquanto não convergir faça
4      se  $i > 0$  então
5           $\bar{\mathbf{u}}_G^i = \mathbf{u}_G^{i-1} + \mathbf{K}_G^{-1} \mathbf{f}_R^i$  - Equação (2.4) ;           /* solução prevista */
6          se relaxação dinâmica então
7              Calcular  $\omega^i$  pela Eq. (2.10) ;                       /* atualização de  $\omega$  */
8          senão
9               $\omega^i = \omega$  ;
10         fim
11          $\mathbf{u}_G^i = \mathbf{u}_G^{i-1} + \omega^i(\bar{\mathbf{u}}_G^i - \mathbf{u}_G^{i-1})$  ;           /* solução corrigida */
12     fim
13     Processamento nas  $n$  mesoescalas - Algoritmo (3.2);           /* Trecho paralelizável da
14         abordagem monolítica */
15     Construção do vetor de resíduos global,  $\mathbf{f}_R^i = \sum_{k=1}^n \mathbf{f}_R^{k,i}$ , a partir dos componentes do vetor
16     de resíduos de todas as  $n$  mesoescalas devidamente ordenados segundo os graus de
17     liberdade compatíveis com o modelo global1;
18      $i \leftarrow i + 1$  ; ;           /* incrementa-se o iterador e retoma-se ao passo inicial */
19 fim

```

¹No Algoritmo (3.2) são calculados os vetores $\mathbf{f}_R^{k,i}$ associados a cada mesoescala. Cada vetor precisa ser mapeado para o vetor \mathbf{f}_R^i , com os graus de liberdade do modelo global (prolongamento do vetor).

Algoritmo 3.2: Trecho paralelizável da Abordagem monolítica para o MEF G^{gl} -GLI.

Entrada:

- Contador das iterações - i ;
- Solução do modelo global na iteração i - \mathbf{u}_G^i ;
- Modelos numéricos referentes as n mesoescalas;
- Lista de nós e elementos da interface, $\Gamma_I = \sum_{k=1}^n \Gamma_I^1 + \dots + \Gamma_I^n$, entre o modelo global e as n mesoescalas;
- Parâmetros da trinca;
- Número de ciclos global-local, n_{GL}^k , referente a estratégia de solução do MEF G^{gl} de cada mesoescala;

Saída:

- Vetor resíduo das n mesoescalas na iteração i .

```

1 para  $k = 0$  até  $n$  faça
2   Impor  $\mathbf{u}_G^i$  como condição de contorno na interface,  $\Gamma_I^k$ , da mesoescala  $k$  ;
3   se  $i > 0$  então
4     Resolve-se a mesoescala enriquecida  $k$  e encontra-se a solução  $\mathbf{u}_{MEnr}^{k,i}$  - Equações (2.18)
       e (2.29);
5   senão
6     Resolve-se a mesoescala  $k$  e encontra-se a solução  $\mathbf{u}_M^{k,0}$  - Equações (2.15) e (2.27);
       /* solução inicial da mesoescala */
7   fim
8   Processamento das iterações global-local do MEF $G^{gl}$ :
9   para  $j = 0$  até  $n_{GL}^k$  faça
10    se  $j > 0$  então
11      Impor  $\mathbf{u}_{MEnr,j-1}^{k,i}$  como condição de contorno do modelo local;
12    senão
13      se  $i > 0$  então
14        Impor  $\mathbf{u}_{MEnr}^{k,i}$  como condição de contorno do modelo local ; /* na primeira
           iteração global-local emprega-se a solução da mesoescala obtida
           na iteração anterior */
15      senão
16         $i = 0$  - o modelo local é construído a partir dos parâmetros da trinca e impõe-se
            $\mathbf{u}_M^{k,0}$  como suas condições de contorno ;
17      fim
18    fim
19    Calcular a solução do modelo local:  $\mathbf{u}_{L,j}^{k,i}$  - Equações (2.16) e (2.28);
20    Enriquecer o modelo da mesoescala  $k$  a partir da solução  $\mathbf{u}_{L,j}^{k,i}$  - Equação (2.19);
21    Calcular a solução do modelo enriquecido da mesoescala:  $\mathbf{u}_{MEnr,j}^{k,i}$  - Equações (2.18) e
       (2.29);
22  fim
23  Calcular  $\mathbf{f}_R^{k,i}$  pela Eq. (2.3) utilizando  $\mathbf{u}_G^i$  e a solução discreta da mesoescala enriquecida,
        $\mathbf{u}_{MEnr,n_{GL}^k}^{k,i}$ ;
24 fim

```

O Algoritmo 3.2 representa o trecho paralelizável da abordagem monolítica aplicada

ao MEF G^{gl} -GLI, responsável pela solução das mesoescalas. Esse processo envolve a construção dos modelos locais e a simulação do fenômeno local de interesse, neste caso as trincas, empregando a estratégia de enriquecimento global-local inerente ao MEF G^{gl} .

Os modelos numéricos fornecidos como entrada de dados são definidos a partir das informações de discretização do domínio, como a lista de nós e elementos devidamente referenciados pela incidência nodal. Adicionalmente, é necessário especificar o modelo de análise adotado, o modelo constitutivo do material, a ordem de integração numérica, bem como as propriedades geométricas da seção e as características do material. Esses dados asseguram a definição completa e consistente do problema para sua resolução numérica.

A lista de nós e elementos da região de interface entre o modelo global e cada mesoescala é empregada para cálculo das grandezas empregadas no acoplamento, a saber:

- Lista de nós: empregada para imposição dos deslocamentos do problema global na interface de cada mesoescala;
- Lista de elementos: empregada para cálculo das forças em cada interface Γ_I^k , sendo estas empregadas no cálculo do vetor resíduo a ser aplicado no modelo global.

A adoção de discretizações equivalentes entre o modelo global e as mesoescalas ao longo de cada interface simplifica o mapeamento entre esses domínios, eliminando a necessidade de algoritmos para compatibilização dos graus de liberdade. Esta é a principal vantagem em utilizar o MEF G^{gl} . Contudo, tal equivalência não constitui uma condição indispensável para a execução do procedimento de solução.

As etapas descritas no Algoritmo 3.1 representam o processo iterativo de solução para cada configuração pré-definida do problema, determinada pelas condições geométricas e de carregamento estabelecidas. No contexto da análise de trincas, cada etapa de propagação corresponde a uma configuração geométrica distinta, demandando a execução do algoritmo para a obtenção da solução específica de cada caso. Esse procedimento assegura a atualização contínua das condições do problema em função da evolução da trinca, mantendo a consistência do modelo ao longo do processo de análise. Deste modo, a convergência do GLI é verificada para cada mesoescala k em todas as configuração do problema, confrontando a tolerância (ϵ) ao resíduo relativo dado pela Eq. (3.1):

$$\frac{\|f_R^{k,i}\|}{\|f_M^{k,0}\|} < \epsilon \quad (3.1)$$

onde $\|f_M^{k,0}\|$ é a norma euclidiana do vetor de reações do modelo da mesoescala k na iteração inicial ($i = 0$), enquanto $\|f_R^{k,i}\|$ é a norma euclidiana do vetor resíduo da mesoescala k na iteração i corrente.

Conforme apresentado na Seção (2.1.1), a técnica de relaxação é empregada na correção da solução prevista do modelo global (Linha 11 - Algoritmo 3.1) objetivando garantir ou acelerar a convergência do GLI. O coeficiente de relaxação, ω , é aplicado a partir da segunda iteração no caso de relaxação estática, ou seja, $\omega^0 = 1,0$. Já para o caso

de relaxação dinâmica, ω é aplicado a partir da terceira iteração, o que significa que $\omega^0 = \omega^1 = 1, 0$.

3.1.2 Algoritmo de solução *staggered*

A abordagem de solução *staggered* aplicada à estratégia MEF G^{gl} -GLI é descrita no Algoritmo 3.3. Nesta estratégia, o processo de enriquecimento global-local e a subsequente atualização da mesoescala, com base na solução dos fenômenos localizados no modelo local, ocorrem somente após o acoplamento GLI, entre as escalas global e meso, alcançar o critério de convergência estabelecido. Esse procedimento assegura a consistência necessária entre o modelo global e as mesoescalas, evitando a propagação de erros decorrentes de interações não convergentes entre essas escalas antes da integração do MEF G^{gl} ao processo de solução.

Dessa forma, a proposição de Li *et al.* (2022), ao introduzir o algoritmo *staggered*, visa melhorar a eficiência computacional do MEF G^{gl} -GLI por meio da redução do número de simulações MEF G^{gl} realizadas durante a análise. Tal premissa fundamenta-se no fato dos ciclos de enriquecimento global-local serem mais onerosos computacionalmente em comparação às iterações GLI, além de exercerem um impacto mais significativo na rigidez do modelo local, afetando diretamente a convergência da estratégia de acoplamento.

De modo semelhante ao apresentado para a abordagem monolítica, o Algoritmo 3.3 corresponde ao processo de solução para cada configuração pré-definida do problema, e, portanto, é executado para cada passo de propagação em análise de trincas. Esta abordagem requer a verificação da convergência em duas etapas do algoritmo: após as iterações GLI e ao final das simulações MEF G^{gl} . Assim, é necessário o uso de dois iteradores distintos para o controle eficiente do processo de solução:

- i_{gli} – contador de iterações GLI, referindo-se às iterações associadas ao acoplamento exclusivo entre o modelo global e cada mesoescala;
- i_s – contador de iterações *staggered*, correspondendo a um ciclo completo de análise, no qual a escala local é incorporada por meio do enriquecimento do MEF G^{gl} .

A verificação da convergência para cada mesoescala k referente as iterações *staggered* e GLI são dadas, respectivamente, pelas Equações (3.2) e (3.3):

$$\frac{\|f_R^{k,i_s}\|}{\|f_M^{k,0}\|} < \epsilon_s \quad (3.2)$$

$$\frac{\|f_R^{k,i_{gli}}\|}{\|f_M^{k,0}\|} < \epsilon_{gli} \quad (3.3)$$

onde $\|f_M^{k,0}\|$ é a norma euclidiana do vetor de reações do modelo da mesoescala k na iteração inicial, $\|f_R^{k,i_s}\|$ e $\|f_R^{k,i_{gli}}\|$ são as normas euclidiana do vetor resíduo da mesoescala k nas iterações i_s e i_{gli} correntes, respectivamente. ϵ_s e ϵ_{gli} são valores predefinidos de

tolerância.

Algoritmo 3.3: Abordagem *staggered* para o MEFG^{gl}-GLI aplicada a problemas com múltiplas mesoescalas.

Entrada:

- Modelo numérico referente a escala global;
- Tipo de técnica de relaxação. Em análises com emprego de relaxação estática, definir o coeficiente de relaxação - ω ;
- Tolerância do GLI e do Staggered (ϵ).

Saída:

- Solução final do modelo da escala global;
- Solução final do modelo das n mesoescalas enriquecidas.

```

1   $i_s \leftarrow 0$ ;
2   $i_{gli} \leftarrow 0$ ;
3  Calcular a solução inicial do modelo global :  $\mathbf{u}_G^0$ ;
4  enquanto não convergir faça
5      enquanto não convergir faça
6          se  $i_{gli} > 0$  então
7               $\tilde{\mathbf{u}}_G^{i_{gli}} = \mathbf{u}_G^{i_{gli}-1} + \mathbf{K}_G^{-1} \mathbf{f}_{\underline{R}}^{i_{gli}}$  - Equação (2.4) ;           /* solução prevista */
8              se relaxação dinâmica então
9                  Calcular  $\omega^{i_{gli}}$  pela Eq. (2.10) ;           /* atualização de  $\omega$  */
10             senão
11                  $\omega^{i_{gli}} = \omega$  ;
12             fim
13              $\mathbf{u}_G^{i_{gli}} = \mathbf{u}_G^{i_{gli}-1} + \omega^{i_{gli}} (\tilde{\mathbf{u}}_G^{i_{gli}} - \mathbf{u}_G^{i_{gli}-1})$  ;           /* solução corrigida */
14         fim
15         Processamento nas  $n$  mesoescalas - Algoritmo (3.4);           /* Primeiro trecho
16         paralelizável da abordagem staggered */
17         Construção do vetor de resíduos global,  $\mathbf{f}_{\underline{R}}^{i_{gli}} = \sum_{k=1}^n \mathbf{f}_R^{k,i_{gli}}$ , a partir dos componentes
18         do vetor de resíduos de todas as  $n$  mesoescalas devidamente ordenados segundo os
19         graus de liberdade compatíveis com o modelo global2;
20          $i_{gli} \leftarrow i_{gli} + 1$ ;
21     fim
22     Processamento nas  $n$  mesoescalas - Algoritmo (3.5); /* Segundo trecho paralelizável
23     da abordagem staggered */
24     Construção do vetor de resíduos global,  $\mathbf{f}_{\underline{R}}^{i_s} = \sum_{k=1}^n \mathbf{f}_R^{k,i_s}$ , a partir dos componentes do
25     vetor de resíduos de todas as  $n$  mesoescalas devidamente ordenados segundo os graus de
26     liberdade compatíveis com o modelo global3;
27      $i_s \leftarrow i_s + 1$ ; /* incrementa-se o iterador e retoma-se ao passo inicial */
28 fim

```

Na proposição original de Li *et al.* (2022), o critério de convergência foi definido pela razão entre as normas euclidianas do vetor de forças residuais na i -ésima iteração, $\mathbf{f}_R^{k,i}$, e

³Nos Algoritmos (3.4) e (3.5) são calculados os vetores $\mathbf{f}_R^{k,i_{gli}}$ e \mathbf{f}_R^{k,i_s} associados a cada mesoescala. Cada vetor precisa ser mapeado para os vetores $\mathbf{f}_{\underline{R}}^{i_{gli}}$ e $\mathbf{f}_{\underline{R}}^{i_s}$, respectivamente, com os graus de liberdade do modelo global (prolongamento do vetor).

o vetor de forças residuais da primeira iteração, $\mathbf{f}_R^{k,0}$. Contudo, conforme discutido por Bueno (2024), foi adotada a norma euclidiana do vetor de reações do modelo da mesoescala na iteração inicial, $\mathbf{f}_M^{k,0}$, como parâmetro para verificar a diferença relativa entre o vetor de forças residuais corrente e o carregamento do problema inicial. Essa mudança foi motivada pela ausência de diferença de rigidez inicial entre os modelos da escala meso e global, o que resultaria em um denominador $\mathbf{f}_R^{k,0}$ extremamente pequeno. No presente trabalho, essa condição também se aplica, uma vez que, assim como nas simulações descritas por Bueno (2024), os mesmos tipos de elementos finitos foram utilizados nos domínios do modelo global e da mesoescala, e os fenômenos localizados foram representados exclusivamente na escala local.

Os Algoritmos 3.4 e 3.5 representam os trechos paralelizáveis da abordagem *staggered* aplicada ao MEF gl -GLI, e, portanto, dividem a solução das mesoescalas em duas partes. O primeiro trecho, Algoritmo 3.4, corresponde a imposição da solução do problema global como condição de contorno da mesoescala e sua solução via MEF, na primeira iteração, ou MEF gl nas iterações subsequentes devido a presença do enriquecimento global-local. Assim, o primeiro trecho paralelizável corresponde as iterações GLI. O segundo trecho, Algoritmo 3.5, corresponde a construção dos modelos locais e a simulação das trincas. Nesta etapa a mesoescala é resolvida através do MEF gl considerando os enriquecimentos da estratégia global-local.

Algoritmo 3.4: Primeiro trecho paralelizável da Abordagem *staggered* para o MEF gl -GLI.

Entrada:

- Contador de iterações IGL - i_{igl} ;
- Solução do modelo global na iteração i_{gli} - $\mathbf{u}_G^{i_{gli}}$;
- Contador de iterações *staggered* - i_s ;
- Modelos numéricos referentes as n mesoescalas;
- Lista de nós e elementos da interface, $\Gamma_I = \sum_{k=1}^n \Gamma_I^1 + \dots + \Gamma_I^n$, entre o modelo global e as n mesoescalas;

Saída:

- Vetor resíduo das n mesoescalas na iteração i_{igl} .

1 **para** $k = 0$ **até** n **faça**

2 Impor $\mathbf{u}_G^{i_{gli}}$ como condição de contorno na interface, Γ_I^k , da mesoescala k ;

3 **se** $i_s > 0$ **então**

4 Resolve-se a mesoescala enriquecida k e encontra-se a solução $\mathbf{u}_{MEnr}^{k,i_{gli}}$ - Equações (2.18) e (2.29);

5 **senão**

6 Resolve-se a mesoescala k e encontra-se a solução sem enriquecimento, $\mathbf{u}_M^{k,i_{gli}}$ - Equações (2.15) e (2.27);

7 **fim**

8 Calcular $\mathbf{f}_R^{k,i_{gli}}$ pela Eq. (2.3) utilizando $\mathbf{u}_G^{i_{gli}}$ e a solução discreta da mesoescala com ou sem enriquecimento, $\mathbf{u}_{MEnr}^{k,i_{gli}}$ ou $\mathbf{u}_M^{k,i_{gli}}$ se $i_s > 0$ ou $i_s = 0$, respectivamente ;

9 **fim**

Algoritmo 3.5: Segundo trecho paralelizável da abordagem *staggered* para o MEFG^{gl}-GLI.

Entrada:

- Contador de iterações IGL - i_{igl} ;
- Contador de iterações *staggered* - i_s ;
- Solução da mesoescala na iteração i_{igl} - $\mathbf{u}_{MEnr}^{k,i_{gli}}$ ou $\mathbf{u}_M^{k,i_{gli}}$;
- Modelos numéricos referentes às n mesoescalas;
- Parâmetros da trinca;
- Número de ciclos global-local, n_{GL}^k , referente à estratégia de solução do MEFG^{gl} de cada mesoescala.

Saída:

- Vetor resíduo das n mesoescalas na iteração i_s .

```

1 para  $k = 0$  até  $n$  faça
2   Processamento das iterações global-local do MEFGgl:
3   para  $j = 0$  até  $n_{GL}^k$  faça
4     se  $j > 0$  então
5       Impor  $\mathbf{u}_{MEnr,j-1}^{k,i_s}$  como condição de contorno do modelo local;
6     senão
7       se  $i_s > 0$  então
8         Impor  $\mathbf{u}_{MEnr}^{k,i_{gli}}$  como condição de contorno do modelo local; /* no primeiro
          ciclo global-local emprega-se a solução da mesoescala obtida na
          iteração anterior */
9       senão
10         $i_s = 0$  - o modelo local é construído a partir dos parâmetros da trinca e
          impõe-se  $\mathbf{u}_M^{k,i_{gli}}$  como suas condição de contorno;
11      fim
12    fim
13    Calcular a solução do modelo local:  $\mathbf{u}_{L,j}^{k,i_s}$  - Equações (2.16) e (2.28);
14    Enriquecer o modelo da mesoescala  $k$  a partir da solução  $\mathbf{u}_{L,j}^{k,i_s}$  - Equação (2.19);
15    Calcular a solução do modelo enriquecido da mesoescala:  $\mathbf{u}_{MEnr,j}^{k,i_s}$  - Equações (2.18) e
      (2.29);
16  fim
17  Calcular  $\mathbf{f}_R^{i_s}$  pela Eq. (2.3) utilizando  $\mathbf{u}_G^{i_{gli}}$  e a solução discreta da mesoescala enriquecida,
       $\mathbf{u}_{MEnr,n_{GL}^k}^{k,i_s}$ ;
18 fim

```

3.2 Aprimoramento dos algoritmos de solução do MEFG^{gl}-GLI

As condições de acoplamento no GLI estão intrinsecamente relacionadas à variação de rigidez entre as escalas de análise, global e meso, impactando diretamente o número de iterações necessárias para convergência. Em problemas que envolvem múltiplas mesoescalas, o subproblema associado ao domínio com a maior penalização de rigidez tende a governar o processo de acoplamento, determinando o número total de iterações executadas

para a solução global.

Nos algoritmos apresentados na Seção (3.1), todas as mesoescalas são processadas iterativamente ao longo de todas as etapas do método. Conseqüentemente, mesmo as mesoescalas associadas a problemas menos complexos, que convergem rapidamente, continuam sendo calculadas até que o domínio com maior penalização de rigidez atinja a convergência. Esse comportamento, embora garanta a consistência no acoplamento global-local, pode resultar em esforço computacional desnecessário para os subproblemas já convergidos.

Nesta seção são apresentados algoritmos aprimorados que implementam uma estratégia adaptativa para o processamento de mesoescalas. Nessa abordagem, o processamento de cada mesoescala é interrompido assim que o domínio associado atinge os critérios de convergência estabelecidos. Como resultado, os domínios já convergidos são excluídos das iterações subseqüentes, evitando cálculos redundantes e minorando o uso de recursos computacionais. Essa técnica pode reduzir de forma significativa o número total de iterações executadas ao longo de cada etapa de propagação da trinca, especialmente em problemas onde há grande disparidade de complexidade entre os subdomínios.

Os benefícios associados ao aprimoramento proposto tornam-se particularmente evidentes em cenários nos quais o número de mesoescalas excede a quantidade de processadores disponíveis. Nessas situações, nos algoritmos não aprimorados, mesoescalas já convergidas permanecem alocadas na fila de processamento, retardando a liberação dos recursos computacionais e prejudicando a eficiência do método, sem contribuir para o aprimoramento da solução.

Os algoritmos propostos devem preservar tanto a precisão quanto a robustez do método. Sua formulação tem como premissa que tanto a solução global, quanto as soluções dos domínios locais que convergem mais rapidamente, são pouco impactadas pelas iterações adicionais necessárias para a convergência dos demais domínios locais. Conseqüentemente, a interação mútua entre diferentes mesoescalas não constitui um aspecto de relevância considerável. Essa hipótese será rigorosamente validada por meio das simulações numéricas apresentadas na Seção (4.2).

3.2.1 Algoritmo de solução monolítica

A estratégia proposta no Algoritmo 3.6 envolve exclusivamente o processo de solução das escalas meso e local, promovendo modificações apenas no trecho paralelizável da abordagem monolítica do MEFG^{gl}-GLI, descrito no Algoritmo 3.2.

Algoritmo 3.6: Versão aprimorada do trecho paralelizável da Abordagem multinólica para o MCFG^{gl}-GLI.

Entrada:

- Contador das iterações - i ;
- Solução do modelo global na iteração i - \mathbf{u}_G^i ;
- Modelos numéricos referentes as n mesoescalas;
- Lista de nós e elementos da interface, $\Gamma_I = \sum_{k=1}^n \Gamma_I^1 + \dots + \Gamma_I^n$, entre o modelo global e as n mesoescalas;
- Parâmetros da trinca;
- Número de ciclos global-local, n_{GL}^k , referente a estratégia de solução do MCFG^{gl} de cada mesoescala;

Saída:

- Vetor resíduo das n mesoescalas na iteração i .

```

1 para  $k = 0$  até  $n$  faça
2   se O parâmetro de convergência da mesoescala  $k$  é Falso então
3     Impor  $\mathbf{u}_G^i$  como condição de contorno na interface,  $\Gamma_I^k$ , da mesoescala  $k$  ;
4     se  $i > 0$  então
5       Resolve-se a mesoescala enriquecida  $k$  e encontra-se a solução  $\mathbf{u}_{MEnr}^{k,i}$  - Equação
        (2.18);
6     senão
7       Resolve-se a mesoescala  $k$  e encontra-se a solução  $\mathbf{u}_M^{k,0}$  - Equação (2.15);
        /* solução inicial da mesoescala */
8     ;
9     fim
10    Processamento das iterações global-local do MCFGgl:
11    para  $j = 0$  até  $n_{GL}^k$  faça
12      se  $j > 0$  então
13        Impor  $\mathbf{u}_{MEnr,j-1}^{k,i}$  como condição de contorno do modelo local;
14      senão
15        se  $i > 0$  então
16          Impor  $\mathbf{u}_{MEnr}^{k,i}$  como condição de contorno do modelo local ; /* na
            primeira iteração global-local emprega-se a solução da
            mesoescala obtida na iteração anterior */
17        senão
18           $i = 0$  - o modelo local é construído a partir dos parâmetros da trinca e
            impõe-se  $\mathbf{u}_M^{k,0}$  como suas condições de contorno ;
19        fim
20      fim
21      Calcular a solução do modelo local:  $\mathbf{u}_{L,j}^{k,i}$  - Equações (2.16) e (2.28);
22      Enriquecer o modelo da mesoescala  $k$  a partir da solução  $\mathbf{u}_{L,j}^{k,i}$  - Equação (2.19);
23      Calcular a solução do modelo enriquecido da mesoescala:  $\mathbf{u}_{MEnr,j}^{k,i}$  - Equações
        (2.18) e (2.29);
24    fim
25    Calcular  $\mathbf{f}_R^{k,i}$  pela Eq. (2.3) utilizando  $\mathbf{u}_G^i$  e a solução discreta da mesoescala
        enriquecida,  $\mathbf{u}_{MEnr,n_{GL}}^{k,i}$ ;
26    se  $\mathbf{f}_R^{k,i}$  atende os critérios de convergência então
27      Define o parâmetro de convergência da mesoescala  $k$  como Verdadeiro;
28    fim
29  fim
30 fim

```

3.2.2 Algoritmo de solução *staggered*

Em relação à abordagem *staggered* do MEFG^{gl}-GLI, os Algoritmos 3.4 e 3.5 são ajustados, mantendo-se inalterado o trecho referente à análise global do problema, conforme o Algoritmo 3.4. A nova estratégia proposta é representada nos Algoritmos 3.7 e 3.8, os quais correspondem às iterações GLI e *staggered*, respectivamente, contemplando a adaptação necessária para aprimorar o processo de solução. Devido a execução do processamento da mesoescala em duas etapas distintas do código, é necessário o uso do mesmo número de parâmetros para o controle da convergência.

Algoritmo 3.7: Versão aprimorada do primeiro trecho paralelizável da Abordagem *staggered* para o MEFG^{gl}-GLI.

Entrada:

- Contador de iterações IGL - i_{igl} ;
- Solução do modelo global na iteração i_{gli} - $\mathbf{u}_G^{i_{gli}}$;
- Contador de iterações *staggered* - i_s ;
- Modelos numéricos referentes as n mesoescalas;
- Lista de nós e elementos da interface, $\Gamma_I = \sum_{k=1}^n \Gamma_I^1 + \dots + \Gamma_I^n$, entre o modelo global e as n mesoescalas;

Saída:

- Vetor resíduo das n mesoescalas na iteração i_{igl} .

```

1 para  $k = 0$  até  $n$  faça
2   se O parâmetro de convergência GLI da mesoescala  $k$  é Falso então
3     Impor  $\mathbf{u}_G^{i_{gli}}$  como condição de contorno na interface,  $\Gamma_I^k$ , da mesoescala  $k$ ;
4     se  $i_s > 0$  então
5       Resolva-se a mesoescala enriquecida  $k$  e encontra-se a solução  $\mathbf{u}_{MEnr}^{k,i_{gli}}$  - Equações
6         (2.18) e (2.29);
7     senão
8       Resolva-se a mesoescala  $k$  e encontra-se a solução sem enriquecimento,  $\mathbf{u}_M^{k,i_{gli}}$  -
9         Equações (2.15) e (2.27);
10    fim
11    Calcular  $\mathbf{f}_R^{k,i_{gli}}$  pela Eq. (2.3) utilizando  $\mathbf{u}_G^{i_{gli}}$  e a solução discreta da mesoescala com
12    ou sem enriquecimento,  $\mathbf{u}_{MEnr}^{k,i_{gli}}$  ou  $\mathbf{u}_M^{k,i_{gli}}$  se  $i_s > 0$  ou  $i_s = 0$ , respectivamente ;
13    se  $\mathbf{f}_R^{k,i_{gli}}$  atende os critérios de convergência então
14      Defina o parâmetro de convergência GLI da mesoescala  $k$  como Verdadeiro;
15    fim
16  fim
17 fim

```

Algoritmo 3.8: Versão aprimorada do segundo trecho paralelizável da abordagem *staggered* para o MEFG^{gl}-GLI.

Entrada:

- Contador de iterações IGL - i_{igl} ;
- Contador de iterações *staggered* - i_s ;
- Solução da mesoescala na iteração i_{igl} - $\mathbf{u}_{MEnr}^{k,i_{gli}}$ ou $\mathbf{u}_M^{k,i_{gli}}$;
- Modelos numéricos referentes às n mesoescalas;
- Parâmetros da trinca;
- Número de ciclos global-local, n_{GL}^k , referente à estratégia de solução do MEFG^{gl} de cada mesoescala.

Saída:

- Vetor resíduo das n mesoescalas na iteração i_s .

```

1 para  $k = 0$  até  $n$  faça
2   se O parâmetro de convergência staggered da mesoescala  $k$  é Falso então
3     Processamento das iterações global-local do MEFGgl:
4     para  $j = 0$  até  $n_{GL}^k$  faça
5       se  $j > 0$  então
6         Impor  $\mathbf{u}_{MEnr,j-1}^{k,i_s}$  como condição de contorno do modelo local;
7       senão
8         se  $i_s > 0$  então
9           Impor  $\mathbf{u}_{MEnr}^{k,i_{gli}}$  como condição de contorno do modelo local;          /* no
           primeiro ciclo global-local emprega-se a solução da
           mesoescala obtida na iteração anterior */
10        senão
11           $i_s = 0$  - o modelo local é construído a partir dos parâmetros da trinca e
           impõe-se  $\mathbf{u}_M^{k,i_{gli}}$  como suas condição de contorno;
12        fim
13      fim
14      Calcular a solução do modelo local:  $\mathbf{u}_{L,j}^{k,i_s}$  - Equações (2.16) e (2.28);
15      Enriquecer o modelo da mesoescala  $k$  a partir da solução  $\mathbf{u}_{L,j}^{k,i_s}$  - Equação (2.19);
16      Calcular a solução do modelo enriquecido da mesoescala:  $\mathbf{u}_{MEnr,j}^{k,i_s}$  - Equações
           (2.18) e (2.29);
17    fim
18    Calcular  $\mathbf{f}_R^{i_s}$  pela Eq. (2.3) utilizando  $\mathbf{u}_G^{i_{gli}}$  e a solução discreta da mesoescala
           enriquecida,  $\mathbf{u}_{MEnr,n_{GL}^k}^{k,i_s}$ ;
19    se  $\mathbf{f}_R^{i_s}$  atende os critérios de convergência então
20      Define o parâmetro de convergência staggered da mesoescala  $k$  como Verdadeiro;
21    fim
22  fim
23 fim

```

3.3 Implementação dos métodos de leitura, gerenciamento e exportação de dados

Devido a ampliação do algoritmo MEFG^{gl}-GLI no ambiente *INSANE* para o tratamento de problemas envolvendo múltiplas mesoescalas, o mecanismo de leitura de dados foi adaptado para atender às novas demandas. Nesse contexto, os dados de entrada foram reorganizados em dois grupos principais: (i) parâmetros globais da análise, associados ao acoplamento GLI, e (ii) parâmetros locais, relacionados à resolução individual de cada mesoescala pelo MEFG^{gl}. Essa separação reflete a estrutura hierárquica do problema, permitindo uma organização eficiente das informações para atender aos requisitos específicos de cada escala no processo de solução. Como parâmetros globais da análise temos:

1. Modelo numérico global;
2. Definição da estratégia de análise do modelo pelo MEFG^{gl}-GLI, podendo ser monolítica ou *staggered*;
3. Técnica de relaxação;
4. Tolerância do GLI;

Como parâmetros locais da análise temos:

1. Modelo numérico da mesoescala (lista de nós e elementos);
2. Modelo constitutivo;
3. Propriedades do material;
4. Propriedades da seção transversal;
5. Lista de nós e elementos da interface entre o modelo global e cada mesoescala;
6. Número de iterações global-local do MEFG^{gl};
7. Parâmetros das condições de contorno do MEFG^{gl};
8. Informações para construção automatizada do domínio local:
 - (a) Emprego de *Zona Buffer* e o número de camadas de elementos a serem adicionadas;
 - (b) Técnica de discretização do domínio e parâmetros de geração da malha;
 - (c) Lista de enriquecimentos.
9. Lista de descontinuidades e seus parâmetros correlatos (coordenadas, modo de abertura, etc.);
10. Parâmetros da propagação de trincas (número de passos).

A partir da implementação deste trabalho, todos os parâmetros locais da análise são dispostos em arquivos de entrada XML, um para cada mesoescala. A nova classe `InputIGLFiles` faz a coleta de todos os arquivos XML no diretório da análise, retornando a quantidade e a lista desses arquivos por meio dos métodos `getNumberOfModelFiles` e `getInputFiles`, respectivamente.

Para cada arquivo de entrada é criada uma variável que referencia um objeto da classe

`PersistenceAsXml` que implementa a interface `Persistence` (nativa do núcleo de pré-processamento do ambiente *INSANE* - Figura (3.1)), na qual estão implementados os métodos de leitura dos dados, a saber:

- `fillInsaneFromFile()`: leitura dos parâmetros locais dos tópicos 1 a 4, 9 e 10 e construção do modelo da mesoescala;
- `fillIGLGlobalLocalSolutionFromFile ()`: leitura dos parâmetros locais do tópico 5 e armazenamento em um objeto da classe `GlobalLocalSimulationManager`;
- `fillLocalInformationFromFile()`: leitura dos parâmetros locais dos tópicos 6 a 8 e armazenamento em um objeto da classe `GlobalLocalSimulationManager`.

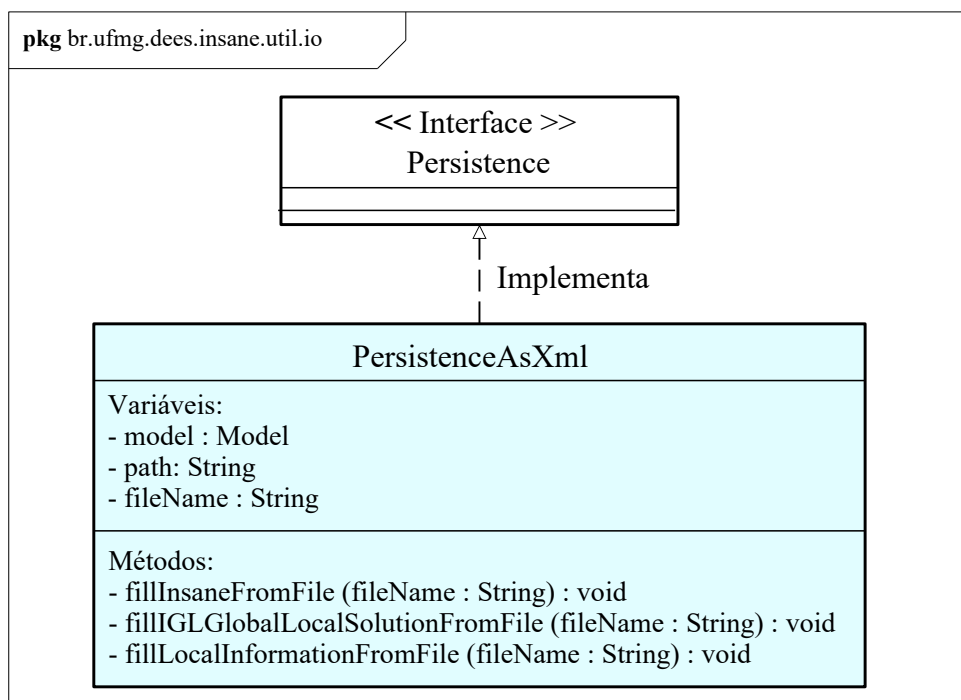


Figura 3.1: Modificações da classe `PersistenceAsXml`.

Quanto à saída de dados, esta era predominantemente realizada através do console integrado à *IDE*. No entanto, em problemas envolvendo múltiplas mesoescalas processadas em paralelo, os resultados provenientes de cada subproblema eram mesclados, o que comprometia a clareza e a organização da informação. Para resolver essa limitação, foi implementada a classe `ConsoleManager`, a qual gerencia a criação de múltiplas janelas de saída, permitindo que os resultados sejam exibidos de forma separada e ordenada. Essa funcionalidade foi implementada utilizando os recursos da biblioteca *Swing*, que proporciona uma interface gráfica para a gestão eficiente das saídas dos dados.

O construtor da classe é responsável pela criação de consoles associados a nomes específicos, correspondentes às diferentes mesoescalas, e os armazena em um *HashMap* estático. Nesse *HashMap*, o título do console é utilizado como chave (*String*), enquanto a instância da classe `ConsoleManager` é armazenada como valor. Essa estrutura permite o armazenamento e o acesso eficiente às janelas de console por meio de títulos exclusivos.

Antes de instanciar um novo objeto, o construtor realiza uma verificação no *HashMap* para confirmar se já existe um console com o título fornecido, e, caso positivo, lança uma exceção. O método `getConsole(String title)` possibilita a recuperação de um console previamente criado, utilizando seu título, o que permite que as janelas abertas sejam acessadas repetidamente ao longo do processo iterativo para registrar os resultados com o método `println(message)`. A estrutura da classe `ConsoleManager` está ilustrada na Figura (3.2).

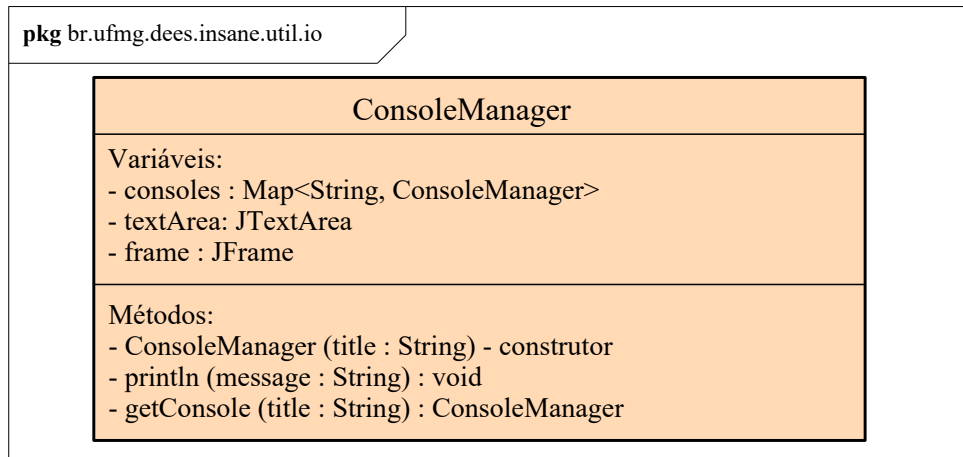


Figura 3.2: Variáveis e métodos da Classe `ConsoleManager`.

3.4 Implementação paralelizada da estratégia de acoplamento $MEFG^{gl}$ -GLI em análises com múltiplas mesoescalas

A partir da entrada dos dados, conforme descrito na Seção 3.3, os parâmetros globais são organizados e armazenados por meio da classe `IGLGlobalLocal`, que estende a classe `Solution`. Essa classe implementa os Algoritmos 3.1 e 3.3, que constituem a base da estratégia de acoplamento $MEFG^{gl}$ -GLI. Adicionalmente, a classe inclui métodos responsáveis pela interação com o software *Abaqus*, utilizando *scripts* desenvolvidos em *Python* conforme proposto por Silveira Filho (2023).

Os dados associados a cada mesoescala são armazenados em uma instância da classe `GlobalLocalSimulationManager`, que conforme descrito na Seção (3.3) desempenha a função de coordenar a interação entre o modelo de mesoescala (domínio global do $MEFG^{gl}$) e o modelo local. A criação das instâncias é realizada por meio do método `createInstance()`, que aplica o conceito de polimorfismo para instanciar diferentes tipos de objetos. Para implementar essa funcionalidade, utiliza-se o padrão de projeto *Factory*, onde o objeto gerado é determinado com base nos parâmetros fornecidos durante a execução. Esses parâmetros especificam as configurações globais (*globSetuperType*), locais (*locSetu-*

perType) e da própria simulação (*glsmType*). No contexto das análises conduzidas neste trabalho, a instância do gerenciador de simulação corresponde ao tipo `IGLGlobalLocalCrackPropagationSimulationManager`.

A estrutura organizacional proposta neste trabalho difere daquela apresentada por Bueno (2024), na qual os parâmetros locais relacionados ao acoplamento eram diretamente configurados no objeto do tipo `Solution`, enquanto os algoritmos de acoplamento estavam implementados na classe `IGLGlobalLocalCrackPropagationSimulationManager`. Tal abordagem, contudo, não era adequada para cenários envolvendo múltiplos domínios locais, o que motivou a reformulação para a estrutura apresentada neste estudo. O sequenciamento dos processos associados aos algoritmos de solução para as abordagens monolítica e *staggered*, implementadas no presente trabalho, são detalhados nos fluxogramas apresentados nas Figuras 3.3 e 3.4.

3.4.1 Classe `IGLGlobalLocalCrackPropagationSimulationManager`

A classe `IGLGlobalLocalCrackPropagationSimulationManager` é responsável por gerenciar a execução das iterações global-local do MEFG^{gl}, bem como pelo enriquecimento do modelo da mesoescala ao longo dos passos associados à propagação de trincas. Essa classe herda métodos de sua classe mãe, `GlobalLocalCrackPropagationSimulationManager`, e define métodos e variáveis adicionais específicos da estratégia de acoplamento GLI, conforme ilustrado na Figura 3.5. Os principais atributos dessa classe estão listados a seguir:

- `r:HashMap<String, double[]>` - armazena as reações residuais da simulação;
- `globalInterfaceElements:ArrayList<String>` - armazena a lista de elementos do modelo global correspondentes a região da interface entre os domínios global e meso;
- `globalInterfaceNodes:ArrayList<String>` - armazena a lista de nós do modelo global correspondentes a região da interface entre os domínios global e meso;
- `mesoInterfaceElements ArrayList<String>` - armazena a lista de elementos do modelo da mesoescala correspondentes a região da interface entre os domínios global e meso;
- `mesoInterfaceNodes:ArrayList<String>` - armazena a lista de nós do modelo da mesoescala correspondentes a região da interface entre os domínios global e meso;
- `converge:boolean` - indicador de convergência para as iterações completas, aquelas que contemplam a execução de ao menos uma iteração global-local do MEFG^{gl};
- `iglConverge:boolean` - indicador de convergência para as iterações GLI, inerentes a abordagem *staggered*.

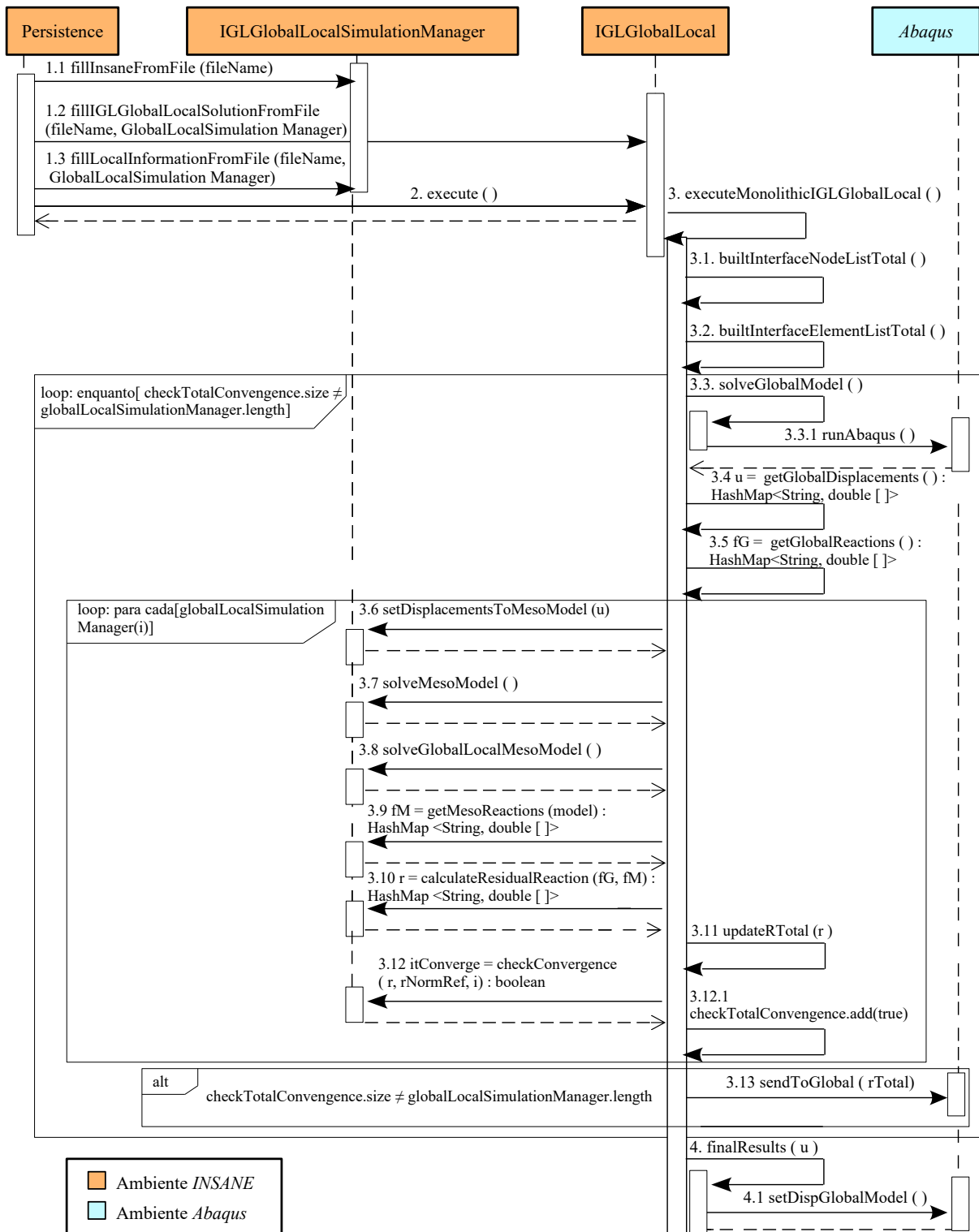


Figura 3.3: Fluxograma de solução da abordagem monolítica

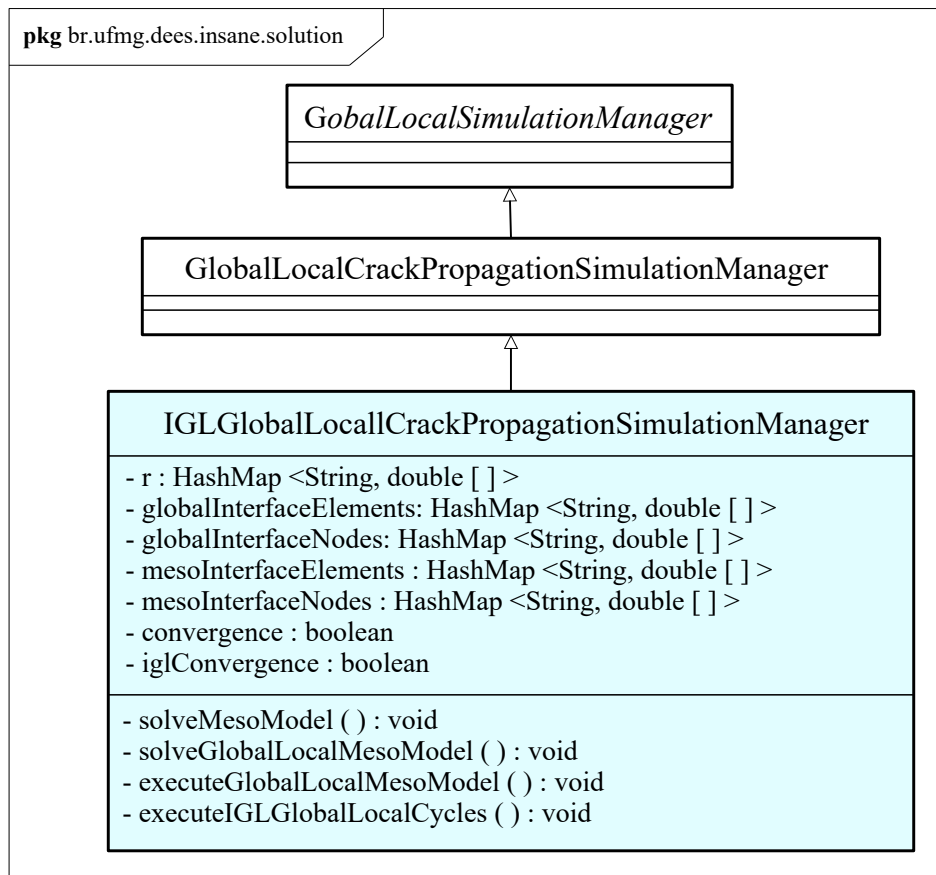


Figura 3.5: Variáveis e métodos da Classe `IGLGlobalLocalCrackPropagationSimulationManager`.

O construtor desta classe realiza a chamada ao construtor da classe mãe utilizando o comando `Super()`, assegurando que a classe `GlobalLocalCrackPropagationSimulationManager` seja devidamente inicializada e que seus atributos e métodos estejam acessíveis. Os métodos definidos nesta classe estão diretamente relacionados à solução das escalas meso e local, abrangendo o processo de enriquecimento do MEF^{gl} e o cálculo dos fatores de intensidade de tensão, conforme descrito a seguir:

- `solveMesoModel: void` - na primeira iteração, soluciona o modelo inicial da meso-escala. Nas demais iterações, resolve o mesoescala enriquecida e extrai os fatores de intensidade de tensão neste domínio;
- `solveGlobalLocalMesoModel: void` - gerencia o registro dos dados de saída, *output*, advindos das iterações global-local e aciona o método `executeGlobalLocalMesoModel`;
- `executeGlobalLocalMesoModel: void` - prepara a execução dos ciclos global-local por meio da limpeza dos fatores de intensidade de tensão advindos de análises anteriores e aciona o método `executeIGLGlobalLocalCycles`. Na primeira iteração, define os parâmetros da ponta da trinca;
- `executeIGLGlobalLocalCycles: void` - executa os ciclos global-local, nos quais o

problema local é solucionado e os fatores de intensidade de tensão calculados neste domínio. Na primeira iteração, o domínio local é construído.

3.4.2 Classe IGLGlobalLocal

A classe `IGLGlobalLocal` é uma subclasse de `GlobalLocal`, a qual, estende a classe `Solution`. Seus principais atributos estão associados aos parâmetros do acoplamento GLI, parte dos quais é obtida a partir dos dados de entrada, incluindo a definição do tipo de acoplamento e os parâmetros de relaxação, ambos especificadas por meio da `PersistenceAsXml`. As variáveis adicionais da classe estão relacionadas à gestão do fluxo iterativo e aos valores gerados e manipulados durante o processo de solução, particularmente no contexto do acoplamento entre o modelo global processado no *Abaqus* e o modelo de mesoescala calculado no *INSANE*.

Entre os principais atributos, destaca-se o `globalLocalSimulationManager:GlobalLocalSimulationManager[]`, uma lista que armazena os objetos instanciados a partir da classe `IGLGlobalLocalCrackPropagationSimulationManager`, representando o conjunto de informações referentes a cada mesoescala envolvida no problema analisado. Os principais atributos dessa classe estão listados a seguir:

- `globalLocalSimulationManager:GlobalLocalSimulationManager[]` - armazena o conjunto de mesoescalas da análise;
- `type:String` - define a estratégia de acoplamento, monolítica ou *staggered*;
- `relaxType:String` - define a técnica de relaxação, empregada para otimizar a velocidade de convergência e/ou assegurar a convergência numérica do processo iterativo;
- `relaxCoefficient:double` - parâmetro de relaxação;
- `iglTolerance:double` - valor da tolerância para o acoplamento;
- `abaqusInfo:HashMap<String, String>` - armazena informações essenciais para a interação com o programa *Abaqus*, incluindo o nome do modelo global e os respectivos casos de carregamento;
- `iterator` - contador das iterações completas, da abordagem monolítica ou *staggered*;
- `iglIterator` - contador das iterações GLI;
- `u:HashMap<String, double[]>` - armazena o somatório dos deslocamentos do modelo global ao longo das iterações;
- `fG:HashMap<String, double[]>` - armazena o somatório das reações dos nós da interface do modelo global ao longo das iterações;
- `fM:HashMap<String, double[]>` - armazena as reações dos nós da interface de cada modelo de mesoescala;
- `du:HashMap<String, double[]>` - armazena os deslocamentos do modelo global a

cada iteração;

- `dfG:HashMap<String, double[]>` - armazena as reações dos nós da interface do modelo global a cada iteração;
- `r:HashMap<String, double[]>` - armazena as reações residuais de cada mesoescala;
- `rTotal:HashMap<String, double[]>` - armazena as reações residuais de todas as mesoescalas, organizadas de forma apropriada, com os valores ordenados de acordo com os nós da interface aos quais seus componentes estão associados;
- `interfaceNodesListTotal:ArrayList<String>` - armazena a lista de nós da interface de todas as mesoescalas;
- `interfaceGlobalElementsListTotal:ArrayList<String>` - armazena a lista de nós da interface de todas as mesoescalas.

A maior parte dos métodos desta classe é dedicada às operações de acoplamento, incluindo a interação e troca de informações de força e deslocamento com o *Abaqus*, o acionamento do processamento de cada mesoescala por meio da classe `IGLGlobalLocalCrackPropagationSimulationManager`, ao cálculo dos vetores de resíduos, e à verificação subsequente da convergência, conforme ilustrado na Figura 3.6. Esses métodos foram originalmente implementados por Bueno (2024) e passaram por modificações pouco significativas.

Por outro lado, os métodos que sofreram alterações substanciais, bem como aqueles desenvolvidos especificamente para viabilizar o processamento de modelos com múltiplas mesoescalas, estão listados a seguir:

- `executeMonolithicIGLGlobalLocal:void` - implementa o Algoritmo 3.1 e executa as etapas de solução da estratégia monolítica do $MEFG^{gl}$ -GLI;
- `executeStaggeredIGLGlobalLocal:void` - implementa o Algoritmo 3.3 e executa as etapas de solução da estratégia *staggered* do $MEFG^{gl}$ -GLI;
- `maxNumberSteps:int` - determina e retorna o número máximo de passos de propagação de trinca entre todas as mesoescalas. A partir desse parâmetro, é definido o número total de etapas de cálculo necessárias para o processo de solução;
- `builtInterfaceNodesListTotal:void` - constrói um `ArrayList` com os nós da interface de todos os modelos de mesoescala;
- `builtInterfaceGlobalElementsListTotal:void` - constrói um `ArrayList` com os elementos da interface de todos os modelos de mesoescala;
- `updateRTotal:void` - atualiza o parâmetro `rTotal:Hashmap` com o vetor resíduo de cada mesoescala;
- `restartCheckConvergence:void` - reinicializa o parâmetro `converge:boolean` de todas as mesoescalas ao ser atingida a convergência em um dado passo de propagação;
- `restartCheckIglConvergence():void` - reinicializa o parâmetro

`iglConverge: boolean` de todas as mesoescalas quando a convergência é alcançada em uma dada iteração GLI na abordagem *staggered*;

- `updateIGLCheckConvergence: ArrayList<Boolean>` - define como *true* o parâmetro `iglConverge: boolean` para as mesoescalas que atingiram a convergência na iteração anterior, conforme o algoritmo aprimorado descrito na Seção 3.2.

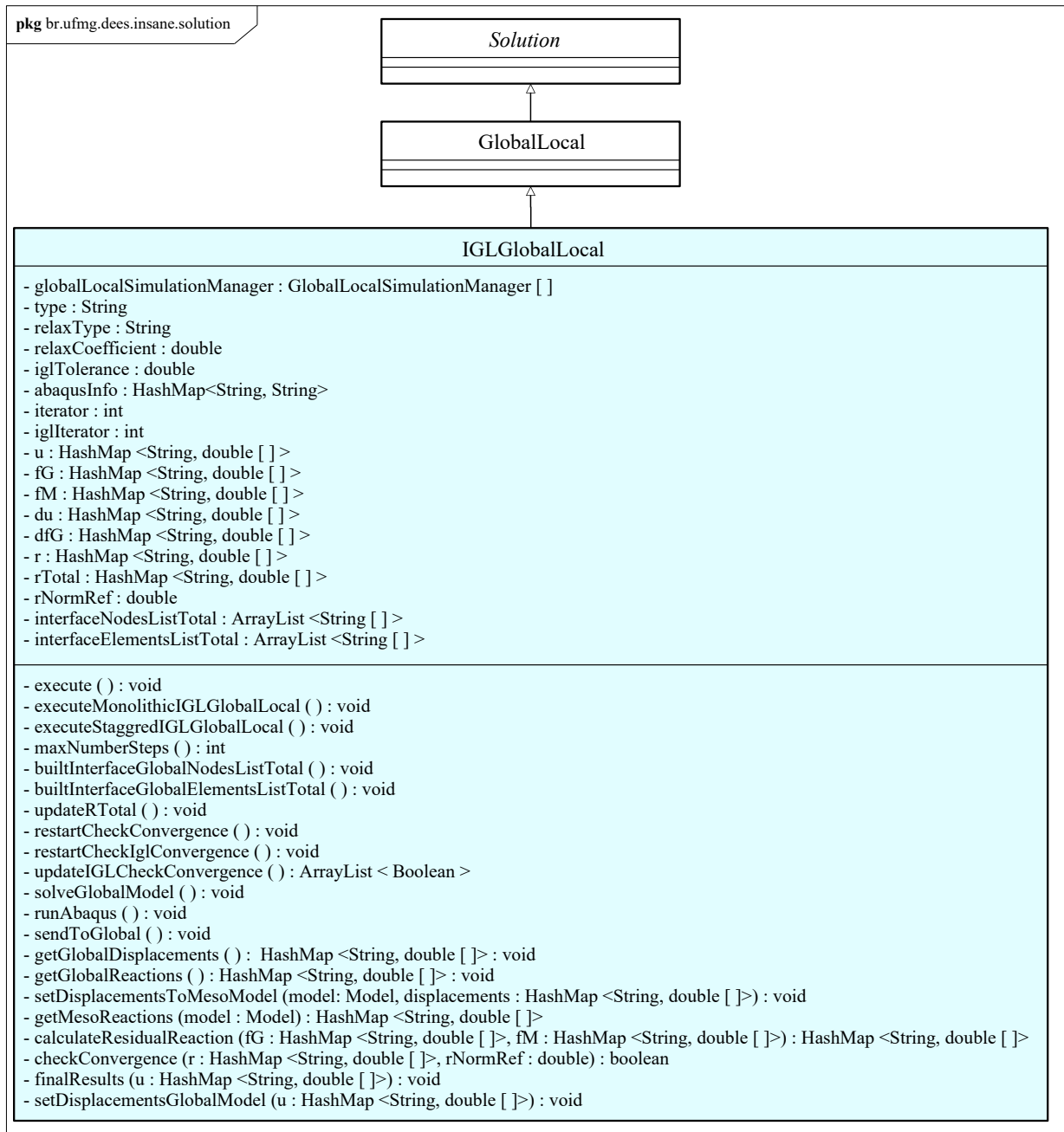


Figura 3.6: Variáveis e métodos da Classe `IGLGlobalLocal`.

Nos métodos `executeMonolithicIGLGlobalLocal: void` e `executeStaggeredIGLGlobalLocal: void`, os trechos responsáveis pelo processamento das mesoescalas foram agrupados em um laço de repetição no qual, a cada iteração, é

selecionada uma das mesoescalas do problema. Com o objetivo de aprimorar o tempo de processamento e aprimorar a eficiência computacional, esse bloco do algoritmo foi organizado em uma lista, `List<Runnable> tasks = new ArrayList<>()`, onde cada elemento `Runnable` corresponde a uma interface funcional em Java que representa uma tarefa passível de ser executada em uma *thread*. Cada índice da lista `globalLocalSimulationManager:GlobalLocalSimulationManager` é processado de forma independente em sua respectiva tarefa `Runnable`, assegurando a independência e a paralelização das operações.

Dessa forma, a construção e a solução das múltiplas mesoescalas e seus respectivos problemas locais, serão paralelizadas por meio do uso de *multithreads*, explorando a abordagem local do paralelismo. Quanto à divisão de carga e ao gerenciamento das *threads*, é adotado um conceito nativo do Java denominado *pool of threads*. O *pool of threads* consiste em um conjunto de *threads* previamente criadas e gerenciadas automaticamente pelo sistema, sendo reutilizadas ao longo da execução da análise, ao invés de gerar uma nova *thread* para cada tarefa a ser processada. Essa abordagem minimiza a sobrecarga associada a criação e destruição de *threads*, resultando em uma melhoria significativa na eficiência do sistema.

Com essa estratégia, a divisão de carga entre os processadores é realizada de forma dinâmica, ou seja, a carga é redistribuída entre as *threads* ao longo da execução. As tarefas são alocadas para as *threads* conforme elas se tornam disponíveis, sem a necessidade de uma divisão prévia e fixa das tarefas antes do início do processo. Essa abordagem torna o paralelismo mais flexível e adaptável, permitindo lidar com eventuais desbalanceamentos na carga de trabalho. O método `newFixedThreadPool()` é uma fábrica da classe `Executors`, responsável por criar um *pool* com um número fixo de *threads*, o qual pode ser configurado para utilizar o número máximo de núcleos de processadores disponíveis para a Máquina Virtual Java, valor este obtido por meio do método `Runtime.getRuntime().availableProcessors()`.

A interface `ExecutorService`, pertencente ao pacote `java.util.concurrent`, é responsável pela gestão do ciclo de vida do executor e pelo controle das *threads*. O *pool of threads* é passado como parâmetro para essa interface, permitindo que as tarefas sejam executadas por meio do método `submit`. O método `shutdown()` assegura que todas as tarefas submetidas sejam concluídas antes do encerramento do *pool*, além de interromper a aceitação de novas tarefas.

Após a solicitação de execução do *pool*, foi implementado um laço `while (!executorService.isTerminated())`, o qual verifica continuamente se todas as tarefas foram concluídas, aguardando a finalização dessas tarefas antes de prosseguir com a execução do algoritmo. Dentro desse laço, encontra-se uma estrutura *try/catch* que desempenha duas funções principais:

- determina um intervalo de tempo para que a *thread* principal verifique o estado

do executor, utilizando o método `Thread.sleep()`. Essa abordagem visa reduzir o uso de CPU, espaçando o intervalo entre as verificações e evitando sobrecarga desnecessária no processamento;

- trata as interrupções da *thread* por meio do método `Thread.currentThread().interrupt()`, permitindo que a *thread* principal seja interrompida de forma controlada, caso ocorra uma solicitação de interrupção, garantindo que o fluxo de execução seja adequadamente interrompido ou finalizado de maneira segura.

Algumas variáveis do código, relacionadas às condições de convergência, como o `rTotal`, são acessadas pelas tarefas associadas a todas as mesoescalas em processamento paralelo. Dessa forma, os trechos de código que envolvem o acesso a essas variáveis são protegidos por blocos `synchronized`, os quais garantem a segurança no acesso e modificação dos recursos compartilhados. Esse comando impede que múltiplas *threads* acessem simultaneamente o conteúdo dentro do bloco, evitando condições de corrida⁴ e garantindo a integridade dos dados.

⁴Condição inerente a tentativa de acesso e manipulação de recursos compartilhados por duas ou mais *threads* simultaneamente.

CAPÍTULO 4

SIMULAÇÕES NUMÉRICAS

Este capítulo apresenta os resultados das simulações numéricas realizadas com múltiplos domínios locais, utilizando a abordagem MEFG^{gl}-GLI. Serão avaliadas, de forma comparativa, as abordagens monolítica e *staggered*, com o objetivo de aferir a eficiência de cada estratégia em função da dimensão do problema considerado, permitindo analisar como o desempenho varia com o aumento da complexidade e do número de subdomínios. A seguir são descritos os aspectos gerais aplicáveis a todas as simulações:

- as malhas são compostas por elementos finitos bidimensionais quadrilaterais de quatro nós (Q4). Esses elementos asseguram alta compatibilidade geométrica em configurações regulares e equilibram o custo computacional e o desempenho numérico. Os elementos Q4 empregam funções de forma bilineares, nas quais cada nó possui dois graus de liberdade correspondentes às translações nas direções x e y . Consequentemente, a interpolação dos deslocamentos nodais apresenta comportamento linear em cada uma das direções espaciais;
- a malha do domínio local é construída a partir dos elementos pertencentes a mesoescala, por meio da subdivisão de cada aresta em três partes iguais. Essa abordagem, denominada Mapeamento Transfinito Bilinear, permite que cada elemento da mesoescala seja refinado em nove elementos no nível da malha local, assegurando uma representação adequada das características geométricas nesta escala do problema;
- a ordem de integração numérica é definida com base na regra de Gauss, sendo empregada uma quadratura 2×2 para a escala global e 3×3 para a mesoescala. No domínio local, os elementos atravessados pela trinca utilizam uma ordem de integração 8×8 , enquanto os demais elementos empregam uma quadratura 4×4 . Ressalta-se que, para a mesoescala enriquecida, na região do domínio coincidente com o problema local, os próprios elementos da malha local são utilizados como células de integração. Os referidos parâmetros advêm do estudo realizado por Bueno (2024);
- empregaram-se condições de contorno de Cauchy, possibilitando a transferência de tensões e deslocamentos de cada mesoescala para os respectivos domínios locais. O campo de tensões recuperado ZZ-BD foi utilizado para aprimorar os dados transferidos, conforme proposto por Marques *et al.* (2025). Conforme indicado na Seção

(2.3), o parâmetro $\bar{\kappa}$ foi determinado numericamente a cada ciclo de iteração global-local;

- baseado nos resultados obtidos por Bueno (2024), definiu-se a realização de apenas uma iteração global-local do MEF^{gl} em cada iteração do acoplamento GLI. Portanto, para todas as mesoescalas, adota-se $n_{GL} = 1$;
- conforme apresentado na Seção (2.1), adota-se como tolerância para o acoplamento $\epsilon_s = \epsilon_{gli} = 10^{-5}$;
- em todas as análises adotou-se relaxação dinâmica;
- os fatores de intensidade de tensão são calculados por meio da estratégia da integral de interação, implementada no *INSANE* por Malekan (2017) e Fonseca (2019);
- as grandezas apresentadas estão expressas em unidades consistentes, as quais serão omitidas ao longo do texto para simplificação;
- as medidas de tempo apresentadas correspondem a média do tempo computacional de três simulações consecutivas.

4.1 Validação da implementação

As análises apresentadas nesta seção têm como objetivo validar a implementação realizada no sistema *INSANE*. Para isso, são calculados os Fatores de Intensidade de Tensão, parâmetros característicos na Mecânica da Fratura, a trajetória da trinca durante sua propagação e valores dos campos de deslocamentos. Os resultados obtidos são comparados com soluções de referência, provenientes da literatura ou geradas por modelos numéricos equivalentes. São discutidos dois exemplos:

- Pórtico com Trincas em Modo Misto de abertura - Seção 4.1.1;
- Chapa com Trincas em Modo I de abertura - Seção 4.1.2;

4.1.1 Pórtico com Trincas em Modo Misto de abertura

Este problema numérico consiste em um pórtico plano, no qual uma carga de $4,2 \times 10^4$ é aplicada no eixo de simetria do elemento, conforme ilustrado na Figura 4.1. O modelo constitutivo adotado é linear elástico, considerando um material homogêneo e isotrópico, com módulo de elasticidade $E = 2,0 \times 10^7$ e coeficiente de Poisson $\nu = 0,30$. O pórtico apresenta uma espessura de $t = 15,0$ e está sujeito ao Estado Plano de Tensões.

As malhas das escalas global e meso são compostas por 1008 e 96 elementos, respectivamente, com dimensões de $5,0 \times 5,0$. Os elementos da malha local possuem dimensões de $1,667 \times 1,667$. Foram avaliadas duas mesoescalas simétricas, nas quais os fenômenos locais de interesse são trincas estáticas localizadas na região de conexão viga-pilar do pórtico. Essas fissuras são representadas cinematicamente no modelo local, gerado automaticamente por meio da estratégia implementada no sistema *INSANE*, conforme descrito por

Fonseca (2019). As trincas são definidas pelos parâmetros $a = 7,9057$, $\beta_{meso1} = 2,8198$ rad e $\beta_{meso2} = 0,3218$ rad.

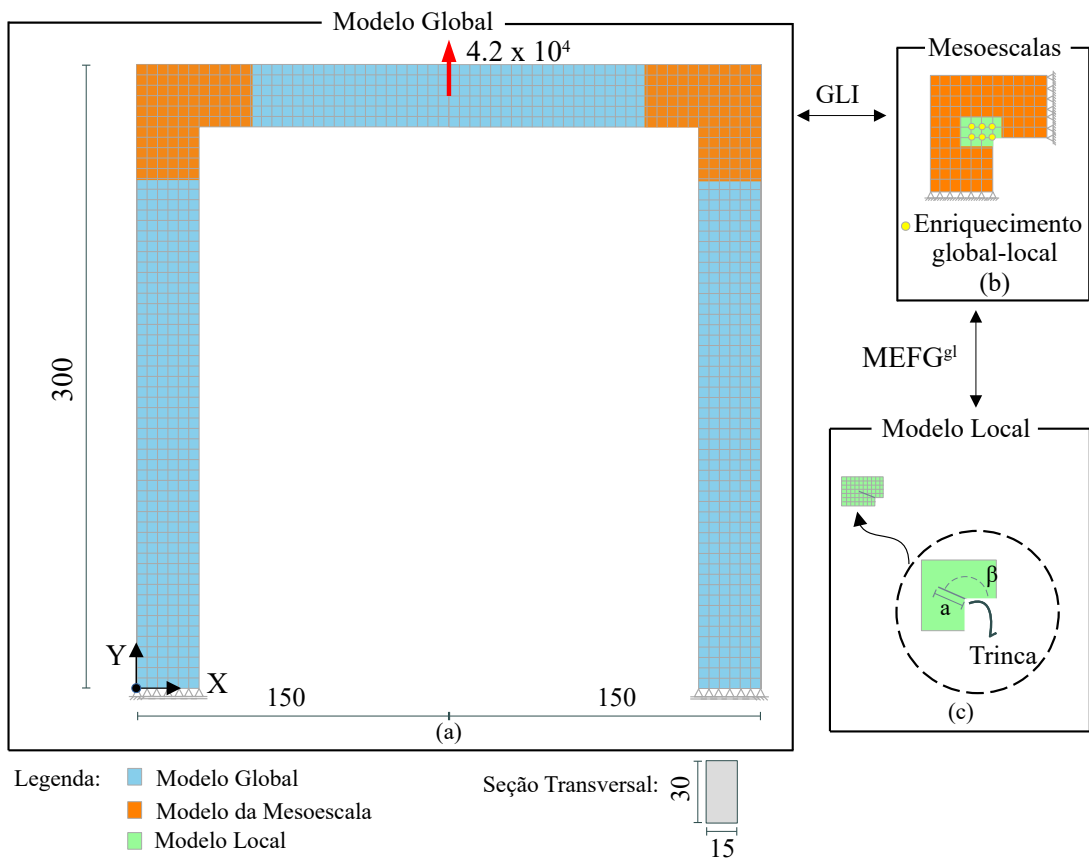


Figura 4.1: Pórtico com trincas em modo misto de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos da ligação viga-pilar à direita podem ser obtidos por simetria.

Este exemplo tem como objetivo validar a implementação dos algoritmos de solução monolítica e *staggered* aplicados ao tratamento de problemas envolvendo múltiplas mesoescalas. Um problema com dois modelos paralelizáveis e simétrico é propositalmente escolhido. Assim, para validar a implementação não apenas a precisão dos resultados é avaliada, mas também a simetria. Para tanto, comparam-se os resultados das duas mesoescalas com os de um modelo numérico equivalente, definido como aquele resolvido no ambiente *INSANE* por meio do MEFG^{gl}, no qual as discretizações global e local são as mesmas adotadas no problema do MEFG^{gl}-GLI. No modelo de referência, as iterações global-local são conduzidas até que a diferença relativa entre o valor atual (iteração i) e o valor anterior (iteração $i - 1$) dos parâmetros K_I e K_{II} seja inferior a 1%.

A avaliação da taxa de convergência do GLI é apresentada nas Figuras (4.2) e (4.3), que correspondem, respectivamente, as abordagens monolítica e *staggered*. A tolerância do acoplamento adotada, $\epsilon_s = \epsilon_{gli} = 10^{-5}$, é destacada pela linha vermelha.

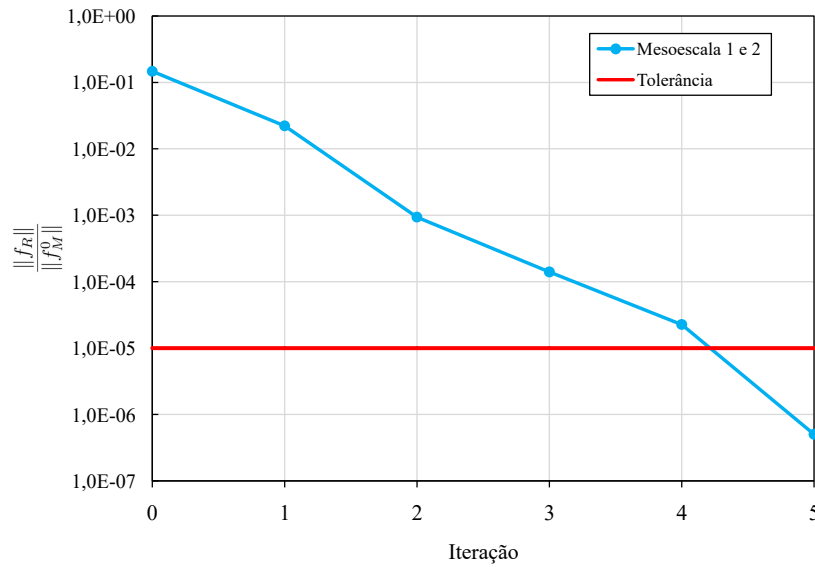


Figura 4.2: Evolução da força residual relativa ao longo dos ciclos para a abordagem monolítica.

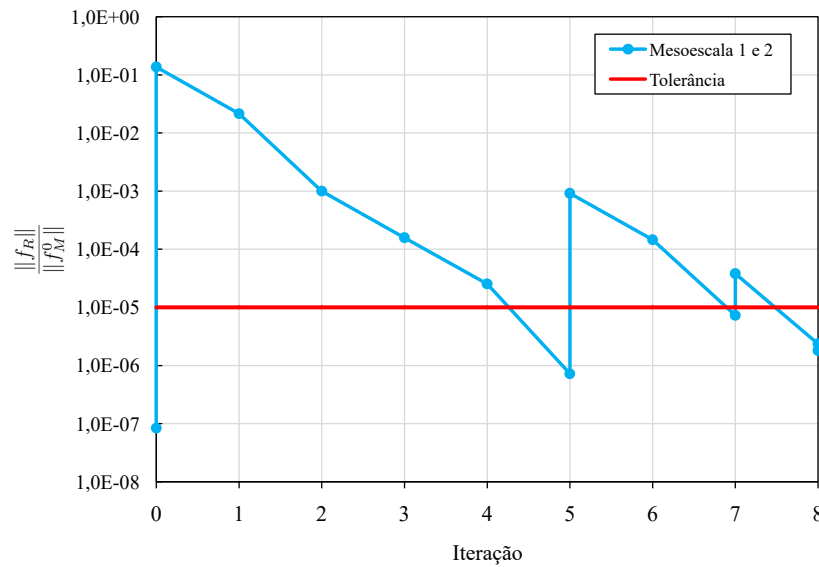


Figura 4.3: Evolução da força residual relativa ao longo dos ciclos para a abordagem *staggered*.

Na abordagem monolítica, foram necessárias 6 iterações para a convergência, enquanto, para a abordagem *staggered*, foram executadas 9 iterações GLI e 4 iterações *staggered*, correspondendo às iterações 0, 5, 7 e 8 - Figura (4.2). Por meio da representação gráfica, é possível identificar a execução das iterações *staggered*, as quais são indicadas por barras verticais. Cada barra corresponde ao par de valores obtidos para a grandeza avaliada em um mesmo ciclo GLI. O primeiro valor refere-se ao cálculo realizado antes da execução da iteração global do MEF^{gl}, e tende a seguir a taxa de convergência das iterações anteriores. O segundo valor corresponde ao cálculo realizado após a iteração global-local, considerando a atualização da rigidez e dos estados de tensão e deformação, com base no

enriquecimento oriundo da solução do problema local. O salto observado na força residual relativa, resultante da atualização das informações provenientes da trinca, é atenuado ao longo das iterações, tornando-se diminuto a medida que se atinge a convergência.

4.1.1.1 Fatores de intensidade de tensão

Os resultados obtidos em termos dos fatores de intensidade de tensão, bem como os erros em relação à solução de referência, são apresentados na Tabela (4.1) e na Figura (4.4) para a abordagem monolítica. De maneira análoga, os dados obtidos através da abordagem *staggered* são apresentados na Tabela (4.2) e na Figura (4.5).

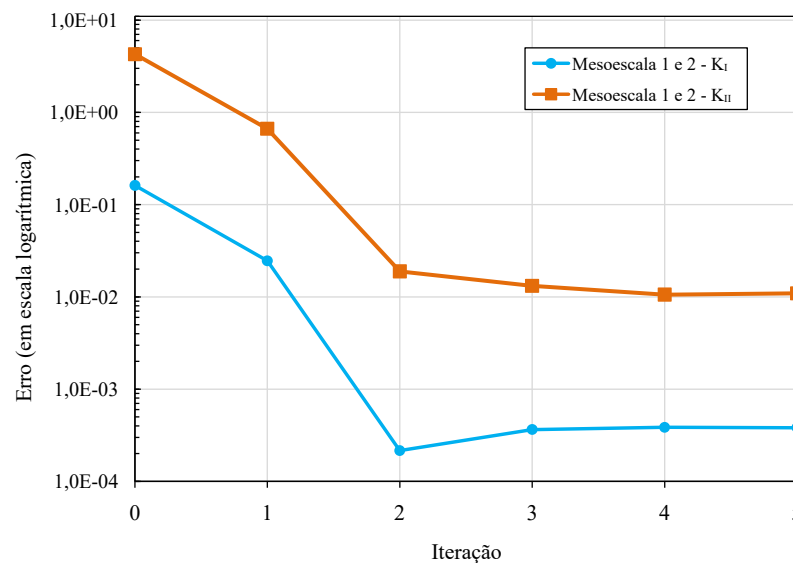


Figura 4.4: Evolução da erro dos fatores de intensidade de tensão para a abordagem monolítica.

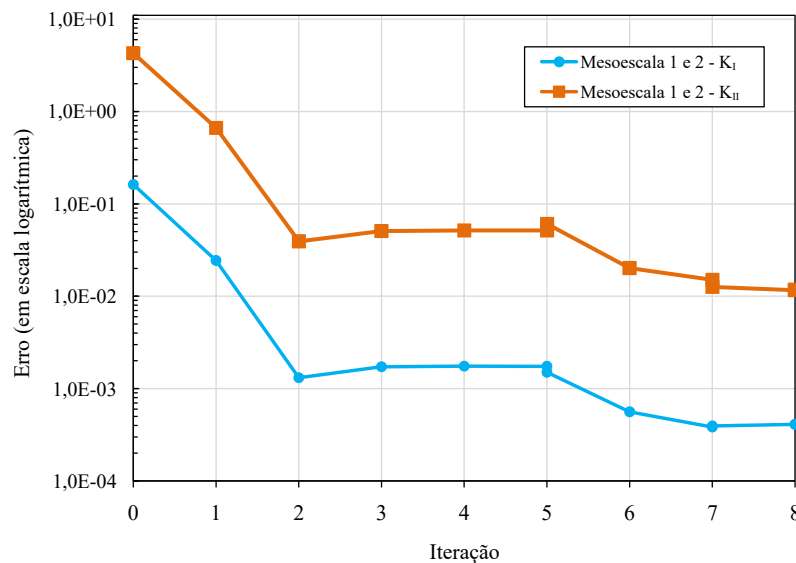


Figura 4.5: Evolução da erro dos fatores de intensidade de tensão para a abordagem *staggered*.

Tabela 4.1: Resultados em termos dos fatores de intensidade de tensão obtidos pela abordagem monolítica.

Iteração	Mesoescala 01				Mesoescala 02			
	K_I	Erro K_I (%)	K_{II}	Erro K_{II} (%)	K_I	Erro K_I (%)	K_{II}	Erro K_{II} (%)
0	1187,061	16,170	-26,376	428,922	1187,061	16,170	26,376	228,923
1	1381,246	2,457	2,674	66,652	1381,246	2,457	-2,674	66,652
2	1415,736	0,022	7,868	1,886	1415,736	0,022	-7,868	1,886
3	1416,557	0,036	8,125	1,318	1416,557	0,036	-8,126	1,335
4	1416,586	0,038	8,103	1,055	1416,586	0,038	-8,103	1,055
5	1416,582	0,038	8,107	1,094	1416,582	0,038	-8,107	1,094

Tabela 4.2: Resultados em termos dos fatores de intensidade de tensão obtidos pela abordagem *staggered*.

Iteração <i>Staggered</i>	Iteração GLI	Mesoescala 01				Mesoescala 02			
		K_I	Erro K_I (%)	K_{II}	Erro K_{II} (%)	K_I	Erro K_I (%)	K_{II}	Erro K_{II} (%)
0	0								
	0	1187,061	16,170	-26,376	428,922	1187,061	16,170	26,376	228,923
	1	1381,500	2,439	2,687	66,493	1381,500	2,439	-2,687	66,494
1	2	1417,895	0,131	8,333	3,917	1417,895	0,131	-8,333	3,917
	3	1418,478	0,172	8,427	5,090	1418,478	0,172	-8,427	5,089
	4	1418,516	0,175	8,433	5,168	1418,516	0,175	-8,433	5,167
	5	1418,511	0,174	8,432	5,156	1418,511	0,174	-8,432	5,156
	5	1418,170	0,150	8,506	6,069	1418,170	0,150	-8,505	6,068
2	6	1416,835	0,056	8,180	2,014	1416,835	0,056	-8,180	2,014
	7	1416,583	0,038	8,140	1,507	1416,583	0,038	-8,140	1,506
	7	1416,599	0,039	8,120	1,261	1416,599	0,039	-8,120	1,261
3	8	1416,621	0,041	8,112	1,164	1416,621	0,041	-8,112	1,163
	8	1416,620	0,041	8,113	1,174	1416,620	0,041	-8,113	1,173

A solução de referência para os fatores de intensidade de tensão é dada por $K_I = 1416,041$ e $K_{II} = 8,019$. Na tabela de resultados, a presença de dois valores para os fatores de intensidade de tensão em uma mesma iteração indica a execução dos ciclos *staggered*.

As duas abordagens de solução conduzem a resultados próximos, apresentando uma diferença de apenas 0,074% para K_{II} e 0,003% para K_I . Em relação à solução de referência, o maior erro encontrado é igual a 1,173%, referente a K_{II} no processo de solução *staggered*. O fato do erro mais significativo corresponder a K_{II} se deve a maior complexidade dos campos de tensão associados ao modo II de abertura da trinca, assim como a reduzida magnitude de seu valor absoluto, o que torna pequenas variações nos resultados mais impactantes na aferição deste parâmetro. A evolução do erro, exibida nas Figuras (4.4) e (4.5), evidencia a efetividade do MEF^{gl}-GLI como solucionador do problema e da funcionalidade do acoplamento entre o *Abaqus* e o *INSANE*.

Na primeira iteração, os resultados obtidos apresentam erros consideráveis, uma vez que as condições de contorno impostas ao problema local são de baixa qualidade, visto que estas são fornecidas pela mesoescala antes do enriquecimento. Portanto, a mesoescala inicial é pouco eficaz em representar adequadamente a solução no contorno local. Para a abordagem monolítica, as três primeiras iterações (0, 1 e 2) apresentam uma redução superior a 95% do erro, sendo tal aprimoramento da solução resultante dos ganhos proporcionados pelo enriquecimento e pela inclusão dos efeitos da descontinuidade nas escalas meso e global. A partir da iteração 2, a solução tende a se estabilizar e converge para o valor final. Na abordagem *staggered*, um comportamento similar é observado a partir da iteração GLI 7, que corresponde a iteração *staggered* 2. Esses resultados evidenciam uma concordância entre as duas abordagens de solução, uma vez que, em ambos os casos, o marco de aprimoramento da solução corresponde à execução do terceiro ciclo global-local do MEF^{gl}.

Avaliando a norma da força residual relativa para as iterações que marcam a estabilização do erro, iterações 2 (abordagem monolítica - Figura (4.2)) e 7 (abordagem *staggered* - Figura 4.3), obtêm-se $\frac{\|f_R^2\|}{\|f_M^0\|} = 9,365 \times 10^{-4}$ e $\frac{\|f_R^7\|}{\|f_M^0\|} = 3,818 \times 10^{-5}$, respectivamente. A magnitude dos valores obtidos pela abordagem *staggered* é dez vezes inferior àquela observada para a monolítica, evidenciando uma maior dependência em relação à tolerância (ϵ) do acoplamento GLI. Esse comportamento está intrinsecamente relacionado ao uso desse parâmetro como critério de execução dos ciclos global-local no MEF^{gl}.

4.1.1.2 Campo de deslocamentos u_y

Com o objetivo de avaliar o comportamento global do problema e a eficácia da estratégia de acoplamento em transferir para o modelo computacional no *Abaqus* os efeitos causados pela descontinuidade, foram analisados os deslocamentos do pórtico na direção de aplicação do carregamento. A Figura (4.6) ilustra o campo de deslocamentos no domínio

global, evidenciando que, ao longo do eixo de aplicação do carregamento, a variável de interesse atinge seus valores máximos.

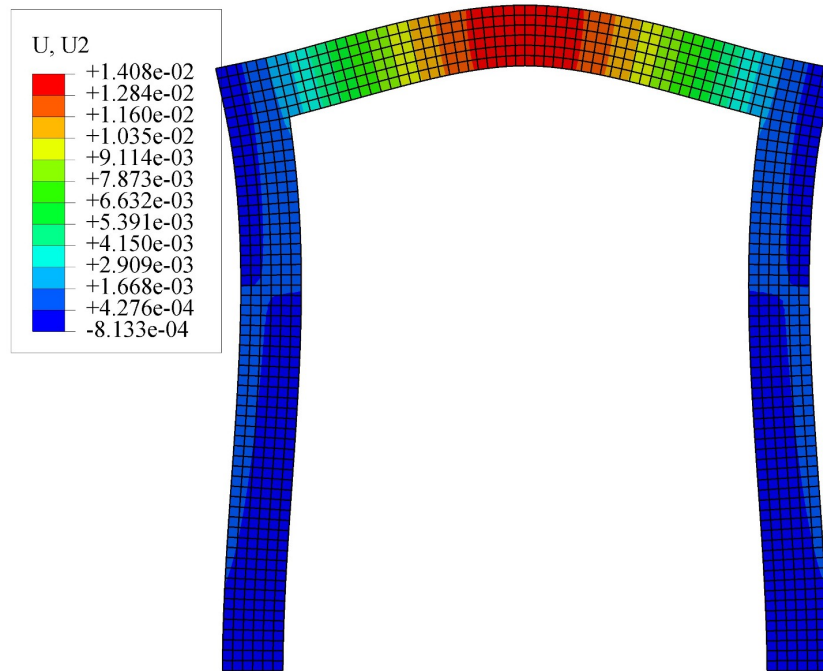


Figura 4.6: Componente de deslocamento vertical, u_y , do modelo global.

Na Figura (4.7) é apresentado o campo de deslocamentos u_y para os domínios locais, os quais exibem o mesmo comportamento observado no domínio global: um aumento dos deslocamentos na direção do centro do pórtico, ou seja, uma elevação em função da proximidade ao ponto de carregamento. A simetria do problema é refletida nos valores de K_{II} , uma vez que o Modo II de abertura da fissura está associado ao efeito de cisalhamento no plano, resultando no deslizamento das partes fendilhadas pela trinca em direções opostas.

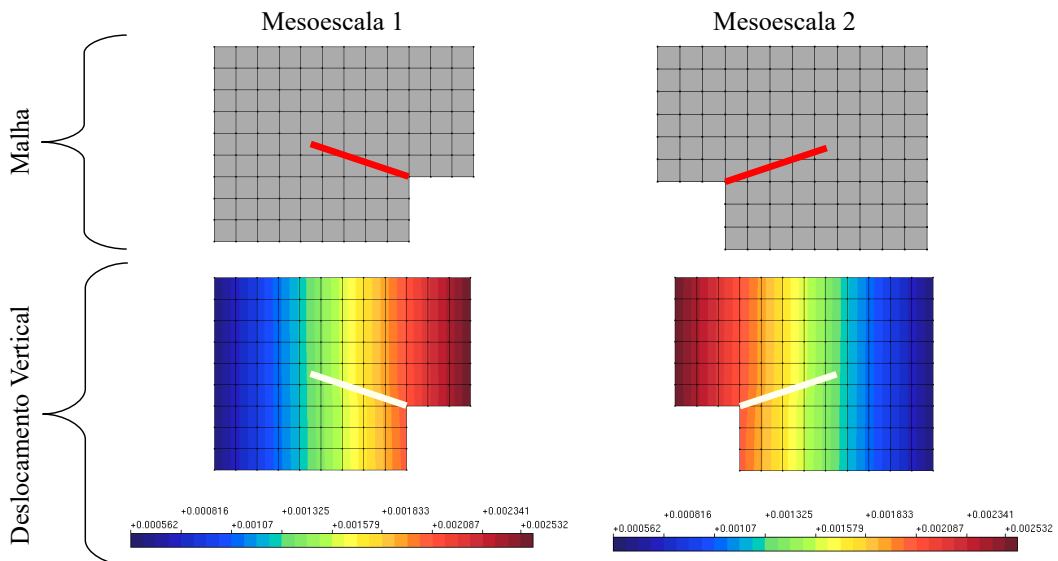


Figura 4.7: Componente de deslocamento vertical, u_y , do modelo local.

A Figura (4.8) apresenta os deslocamentos verticais, u_y , ao longo da aresta superior do pórtico, ou seja, para $y = 300$. O modelo utilizado para obter a solução de referência possui a mesma discretização do modelo global inerente ao MEFG^{gl}-GLI, portanto, os valores foram diretamente aferidos nos pontos nodais, que possuem correspondência entre os modelos. Os valores obtidos para os três modelos analisados apresentam uma alta conformidade, com diferenças a partir da sexta casa decimal. A menor variação observada entre os resultados para o campo de deslocamentos, em comparação aos fatores de intensidade de tensão, pode ser atribuída a variável analisada ser uma grandeza global, portanto, menos impactada por perturbações numéricas em grandezas de natureza local. Conseqüentemente, a estratégia de acoplamento e a execução dos ciclos global-local tem um impacto menor sobre o campo de deslocamentos.

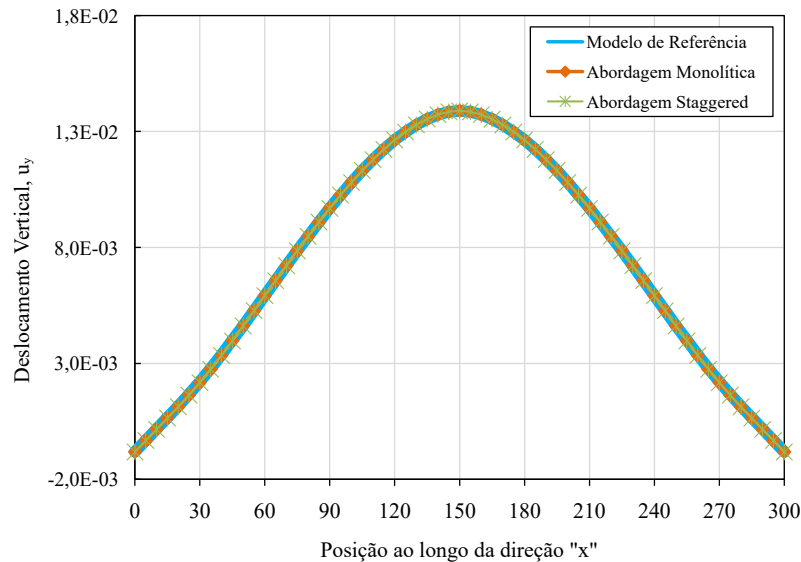


Figura 4.8: Componente de deslocamento vertical, u_y , na aresta superior do pórtico ($y = 300$).

4.1.2 Chapa com Trincas de borda em Modo I de abertura

O problema abordado nesta seção consiste em uma chapa submetida à tração em estado plano de deformação, sob uma tensão axial uniforme de $\sigma = 375$, cujas condições de contorno são representadas na Figura (4.9). A seleção deste modelo foi motivada pela necessidade de avaliar os possíveis efeitos da interação entre as mesoescalas, considerando a expectativa de propagação da trinca em Modo I puro e de forma simétrica. O material da chapa é caracterizado pelas seguintes propriedades mecânicas: módulo de elasticidade $E = 2,07 \times 10^5$ e coeficiente de Poisson $\nu = 0,30$. Mantêm-se as hipóteses de material homogêneo e isotrópico, e a espessura da chapa é definida como $t = 8,0$.

No que se refere à discretização, o modelo global é constituído por 1350 elementos com dimensões de $7,0 \times 7,0$, enquanto cada mesoescala é composta por 49 elementos de igual tamanho. Os elementos da malha local apresentam dimensões de $2,333 \times 2,333$, sendo

que sua extensão é adaptada ao longo do processamento em função da propagação da trinca. A Figura (4.9 - c) apresenta a malha gerada para o entalhe inicial, composta por 81 elementos. Essa configuração é mantida até o terceiro passo de propagação, a partir das etapas subseqüentes da análise, é adotada uma malha composta por 108 elementos.

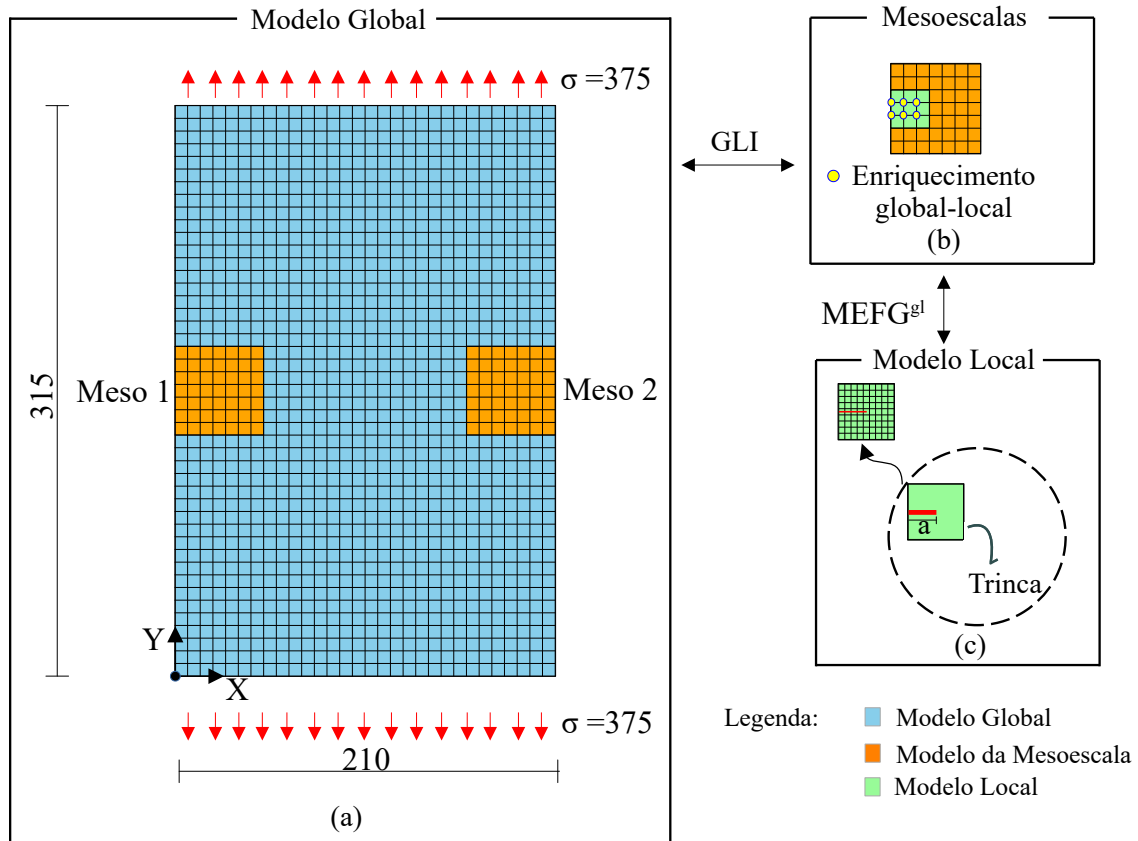


Figura 4.9: Chapa tracionada com trincas de borda em modo I de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos das escalas meso e local associados à trinca da direita, podem ser obtidos por simetria.

As mesoescalas analisadas apresentam trincas horizontais com comprimento inicial $a = 10,0$, orientadas perpendicularmente a tensão aplicada ($\beta_{meso1} = 0$ rad), configurando um cenário típico de Modo I de abertura. Neste estudo, avaliou-se a eficiência do método para a determinação dos fatores de intensidade de tensão ao longo de 5 passos de propagação, adotando um incremento de comprimento $\Delta_a = 1.750$. Para a solução de referência de K_I , utilizou-se a Equação (4.1), uma expressão empírica descrita em Dowling (2013), adequada para situações em que a relação $a/b < 0,13$. b é definido como a metade da largura da chapa.

$$K_I = 1,12\sigma\sqrt{\pi a} \quad (4.1)$$

A Tabela (4.3) apresenta os valores da solução de referência para K_I , calculados em

função do comprimento a da trinca para cada etapa de propagação. Como a propagação ocorre exclusivamente em Modo I, tem-se que $K_{II} = 0, 0$.

Tabela 4.3: Solução de referência para K_I referente aos 5 passos de propagação da trinca.

Passo de Propagação	1	2	3	4	5
Tamanho da trinca - a	10,0	11,75	13,50	15,25	17,00
$K_{I,ref}$	2354,096	2551,780	2735,213	2907,094	3069,366

A evolução da norma da força residual relativa ao longo das iterações é apresentada nas Figuras (4.10) e (4.11), correspondentes às abordagens de solução monolítica e *staggered*, respectivamente. Para melhor interpretação dos gráficos, regiões foram sombreadas para demarcar os resultados associados a cada passo de propagação da trinca. Nota-se que a primeira iteração de cada novo passo de propagação é caracterizada por um aumento significativo na força residual, decorrente das alterações nos campos de tensão e deformação provocadas pelo incremento do comprimento da fissura.

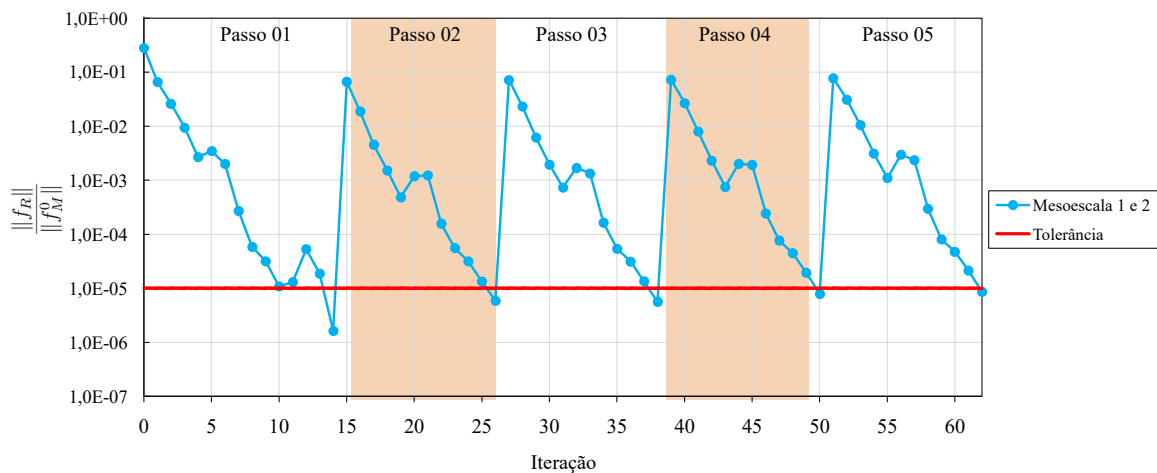


Figura 4.10: Evolução da força residual relativa ao longo dos ciclos para a abordagem monolítica.

Diferentemente do comportamento observado no problema abordado na seção 4.1.1, neste exemplo, a força residual relativa apresentou uma leve oscilação durante as iterações intermediárias de cada passo de propagação. Tal comportamento pode ser atribuído ao uso da técnica de relaxação, e decorre dos casos em que a adoção de $\omega > 1, 0$ provocou correções superestimadas na aproximação, fazendo com que os resultados da iteração corrente sejam significativamente afetados por flutuações na solução obtida na iteração anterior.

Nas iterações iniciais de cada passo a aproximação encontra-se distante da respectiva solução, e os efeitos causados pelas discontinuidades são menos precisos, o que reduz a possibilidade da solução ser excedida devido ao uso do fator de relaxação. Nas iterações subsequentes, a medida que a discontinuidade e suas interferências nos campos de tensão

e deformação são progressivamente incorporadas ao processo de solução, a aproximação torna-se mais sensível a pequenas alterações, resultando em oscilações mínimas que são observadas ao redor da iteração central de cada passo de propagação. Conforme ilustrado nas Figuras (4.10) e (4.11), esse efeito, embora presente, é sutil e não compromete o processo, pois com o aumento do número de iterações, a tendência de convergência é restabelecida e o erro diminui até alcançar a estabilização.

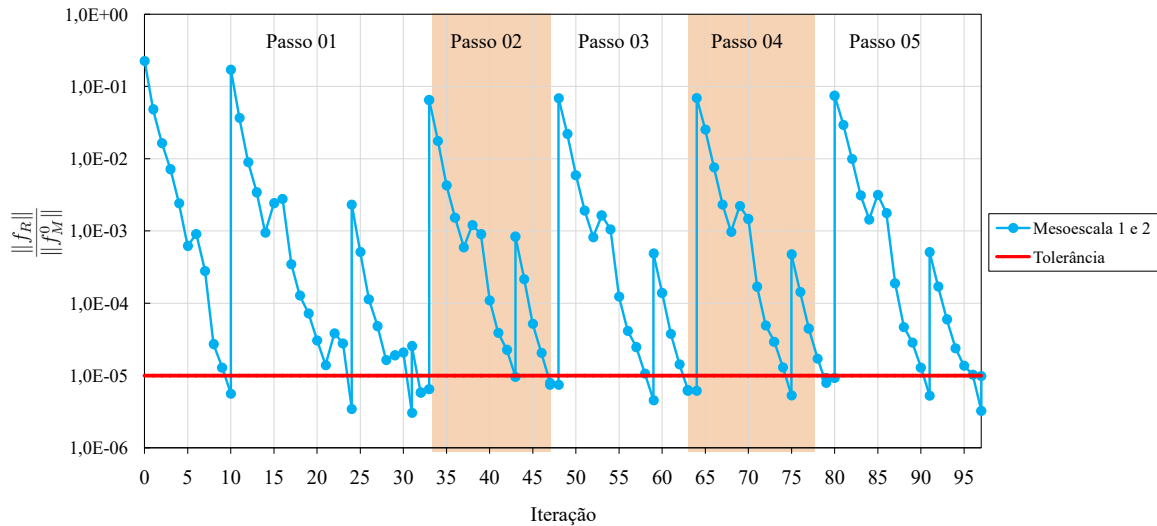


Figura 4.11: Evolução da força residual relativa ao longo dos ciclos para a abordagem *staggered*.

Nos casos em que $\omega < 1$, o incremento entre as iterações é reduzido, o que permite que a solução seja atualizada de maneira mais controlada, evitando que o método responda de forma excessiva. Esses aspectos destacam as vantagens da relaxação dinâmica, pois a capacidade de ajustar o parâmetro ω ao longo do processo, em resposta à variação das condições de rigidez do problema, proporciona maior eficiência no processo de convergência.

4.1.2.1 Fatores de intensidade de tensão

Os resultados obtidos para os fatores de intensidade de tensão (K_I) e os erros em relação à solução de referência são apresentados na Figura (4.12) e na Tabela (4.4) para a abordagem monolítica, e na Figura (4.13) e na Tabela (4.5) para a abordagem *staggered*.

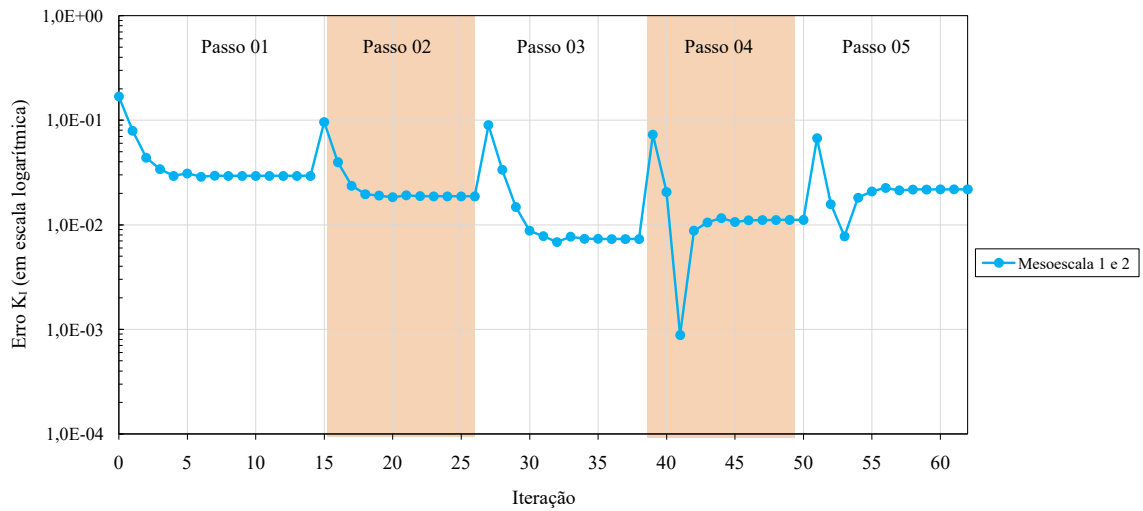


Figura 4.12: Evolução do erro de K_I para a abordagem monolítica.

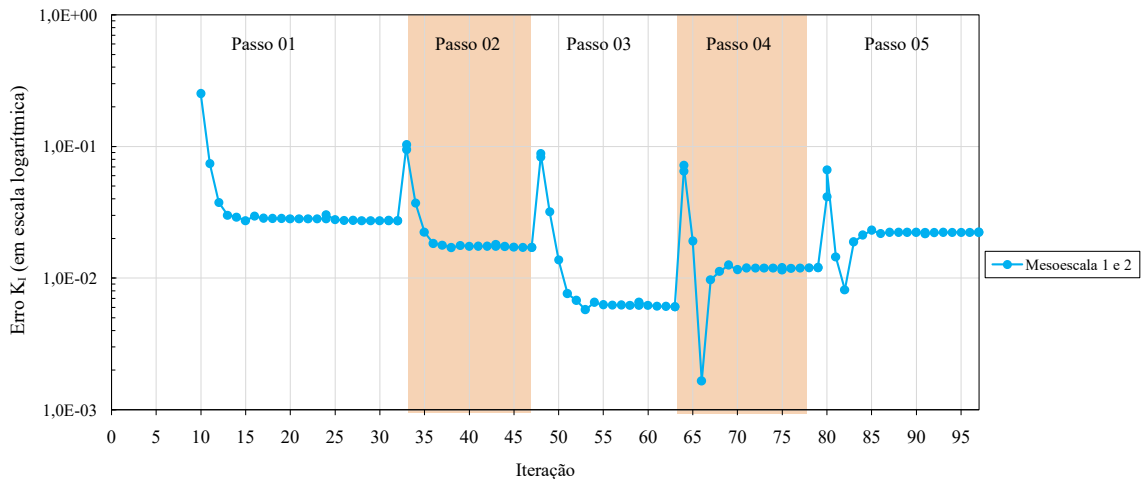


Figura 4.13: Evolução do erro de K_I para a abordagem *staggered*.

Os resultados obtidos apresentam uma boa concordância com os valores de referência, com o erro máximo observado sendo consideravelmente inferior a 10,0%, valor que corresponde à margem de erro da solução empírica em comparação aos resultados obtidos por meio de ensaios laboratoriais conforme indicado por Dowling (2013).

É importante ressaltar que não há acúmulo de erros nem comprometimento da qualidade da solução ao longo da propagação da trinca, mantendo-se os mesmos valores quando comparados os resultados das duas mesoescalas cuja solução foi paralelizada. Para a abordagem *staggered*, os primeiros valores de K_I foram obtidos na iteração GLI 10, a qual corresponde à primeira iteração *staggered*.

Tabela 4.4: Resultados de K_I obtidos pela abordagem monolítica.

Passo de Propagação	Mesoescala 1				Mesoescala 2			
	Número de Iterações	K_I	Erro de K_I (%)	K_{II}	Número de Iterações	K_I	Erro de K_I (%)	K_{II}
1	15	2285,042	2,933	9,906E-06	15	2285,042	2,933	-1,068E-05
2	12	2504,095	1,869	-4,432E-06	12	2504,095	1,869	-4,944E-06
3	12	2715,262	0,729	3,352E-06	12	2715,262	0,729	-3,666E-06
4	12	2939,454	1,113	-1,111E-05	12	2939,454	1,113	1,129E-05
5	12	3136,312	2,181	6,874E-06	12	3136,312	2,181	-6,840E-06

Tabela 4.5: Resultados de K_I obtidos pela abordagem *staggered*.

Mesoescala 1						
Passo de Propagação	Número de Iterações <i>staggered</i>	Número de Iterações GLI	K_I	Erro de K_I (%)	K_{II}	
1	4	33	2289,796	2,731	8,652E-06	
2	3	15	2508,102	1,712	-3,412E-06	
3	3	16	2718,618	0,607	2,700E-06	
4	3	16	2941,985	1,200	-1,040E-05	
5	3	18	3137,729	2,227	5,858E-06	
Mesoescala 2						
Passo de Propagação	Número de Iterações <i>staggered</i>	Número de Iterações GLI	K_I	Erro de K_I (%)	K_{II}	
1	4	33	2289,796	2,731	-9,427E-06	
2	3	15	2508,102	1,712	3,944E-06	
3	3	16	2718,618	0,607	-3,319E-06	
4	3	16	2941,985	1,200	1,108E-05	
5	3	18	3137,729	2,227	-6,097E-06	

Conforme observado nas Figuras (4.12) e (4.13), a pequena oscilação na força residual relativa não afetou as condições de acoplamento entre os modelos, nem a extração de K_I , pois não foi registrado comportamento oscilatório nos valores obtidos para este parâmetro. Ao longo de todas as iterações, os valores de K_{II} obtidos foram da ordem de 10^{-6} , entendido como zero numericamente.

Para ambas as abordagens de solução, o maior número de iterações requeridas para a convergência no primeiro passo de propagação da trinca, em comparação aos demais, pode ser atribuído ao modelo inicial não considerar a influência da descontinuidade. Conforme implementado no *INSANE* por Bueno (2024), a partir da primeira iteração global-local, o enriquecimento da mesoescala é mantido e atualizado, o que confere maior robustez ao modelo no tratamento dos fenômenos localizados.

Embora na primeira iteração de um novo passo de propagação o enriquecimento seja construído com informações derivadas da trinca de comprimento a do passo anterior, a variação de rigidez entre as escalas meso e local desse modelo é significativamente menor do que a diferença observada entre o modelo inicial, que não contempla a trinca, e o domínio da mesoescala, que inclui o entalhe inicial via enriquecimento global-local.

4.1.2.2 Propagação da Trinca

A Figura (4.14) ilustra os cinco passos da trajetória de propagação da trinca, bem como os respectivos domínios locais gerados automaticamente no ambiente *INSANE*. Dessa forma, valida-se o comportamento esperado para a ocorrência do Modo I puro, caracterizado pela abertura da trinca e propagação linear na mesma direção do entalhe inicial.

O componente do campo de deslocamentos na direção de propagação das trincas, u_x , é apresentado nas Figuras (4.16) e (4.15) para os domínios local e global, respectivamente. O comportamento do campo de deslocamentos demonstra elevada compatibilidade entre os diferentes domínios de análise, refletindo os efeitos mútuos entre as escalas global e local.

A ocorrência dos deslocamentos máximos nas bordas verticais da chapa e a direção do fluxo orientado para o eixo de simetria, localizado internamente ao elemento, resultam do comportamento global da peça, explicado pelo efeito de *Poisson* gerado pela tração perpendicular ao eixo x . A perturbação dos campos de deslocamento nas regiões onde a descontinuidade é inserida decorre da contribuição do comportamento local do problema, refletindo a redução dos deslocamentos u_x nas proximidades das regiões previamente rompidas pela trinca, nas quais surgem superfícies livres. Destaca-se a manutenção da simetria ao longo de todas as iterações.

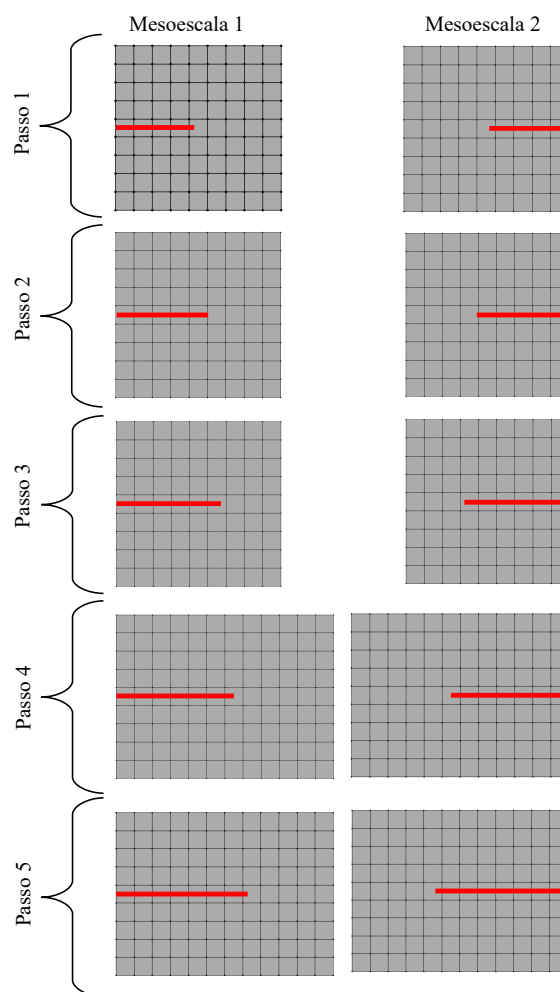


Figura 4.14: Modelos locais gerados ao longo dos passos de propagação da trinca.

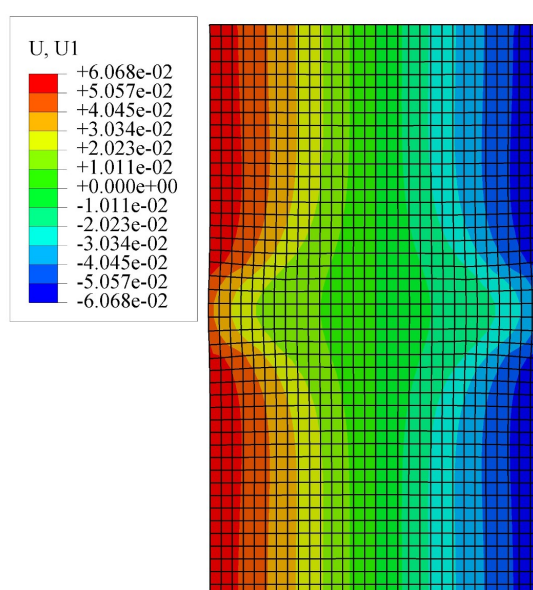


Figura 4.15: Deslocamento horizontal, u_x , no domínio global ao final do último passo de propagação.

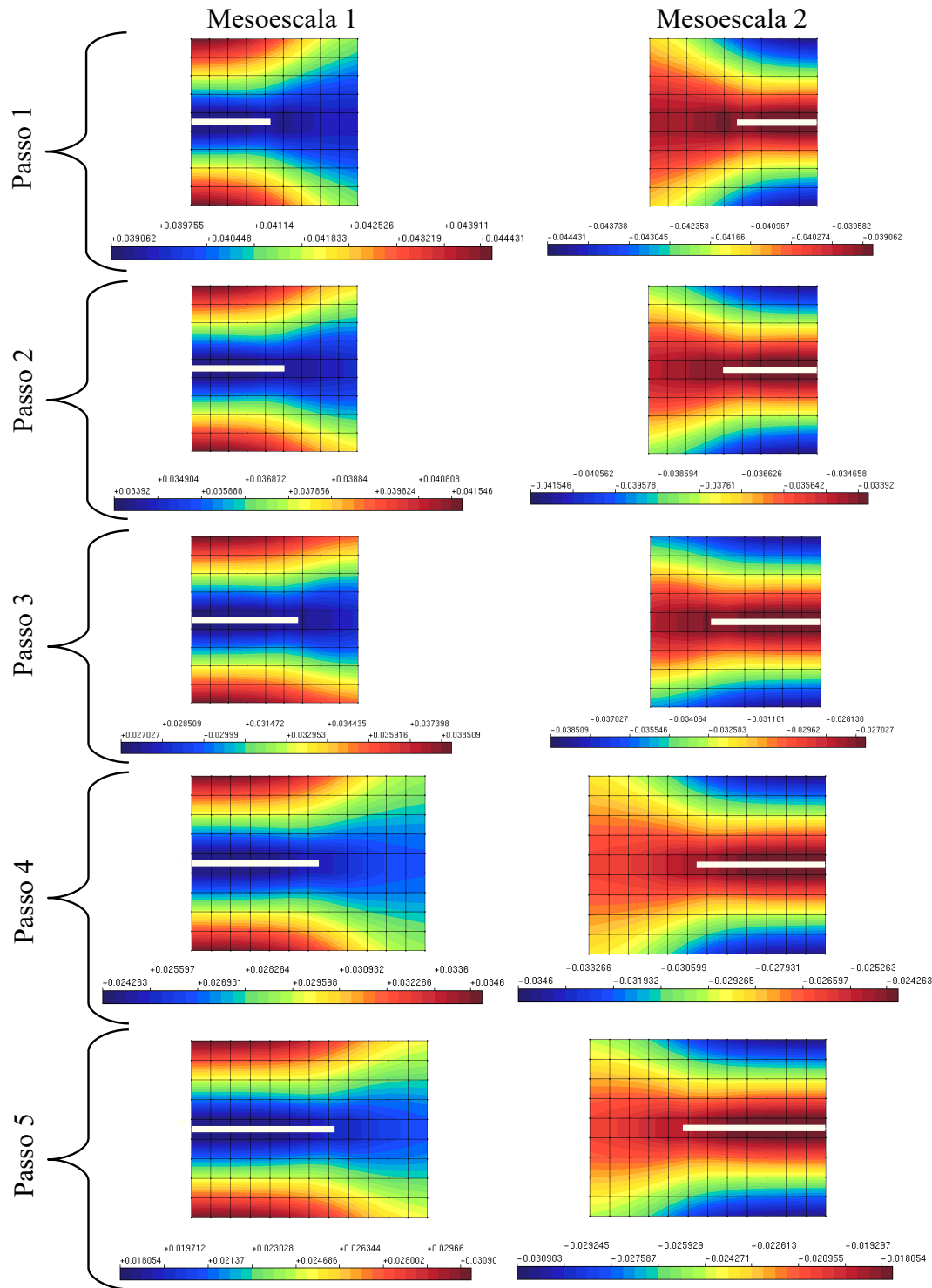


Figura 4.16: Componente de deslocamento horizontal, u_x , observado para os 5 passos de propagação da trinca no domínio local.

Quando uma trinca se propaga, as tensões normais na direção perpendicular as superfícies da trinca, σ_{YY} , tendem a se aliviar a medida que se aproximam das faces livres. Esse fenômeno ocorre devido ao reajuste do equilíbrio interno de tensões, em que a parcela do esforço antes suportada pelas regiões rompidas é assumida pelas áreas intactas do material. Assim, próximo à ponta da trinca, as tensões atingem valores elevados, carac-

terizados pelo fator de intensidade de tensão K . Esse comportamento é ilustrado para cada etapa da propagação na Figura (4.17).

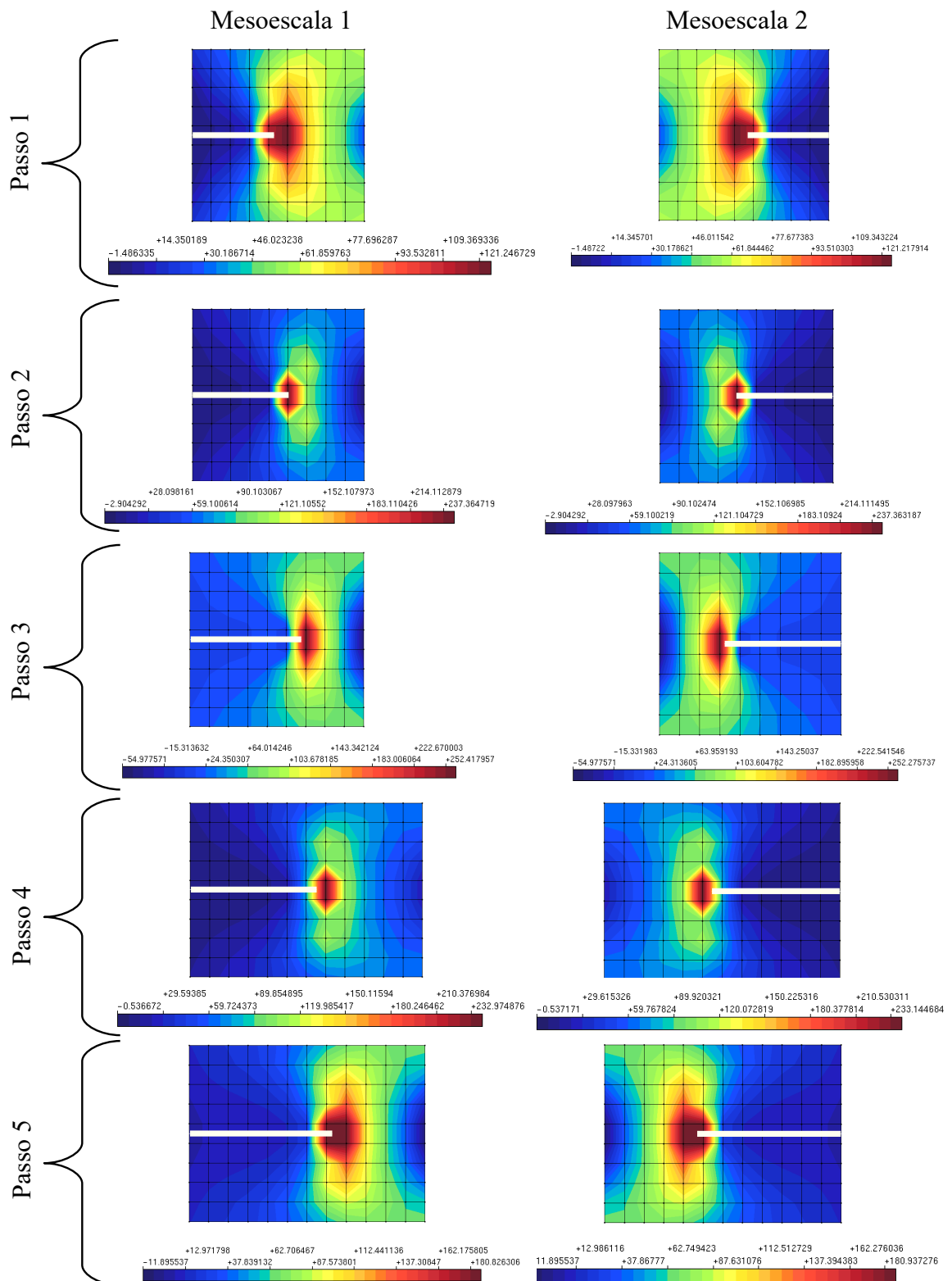


Figura 4.17: Componente de tensão σ_{YY} inerente aos 5 passos de propagação da trinca.

4.2 Desempenho Computacional

Nesta seção, realiza-se a avaliação do desempenho computacional associado à incorporação de técnicas de paralelismo no processo solutivo do MEF^{gl}-GLI. O desempenho é analisado com base nos tempos de processamento e nas correlações discutidas na Seção 2.5.4. Todas as simulações foram conduzidas sob condições controladas e no mesmo equipamento, equipado com um CPU Intel® Core™ i5-10300H @ 2.5 GHz, composto por quatro núcleos físicos e oito processadores lógicos. Para estas análises, os modelos descritos na Seção (4.1) foram ampliados pela inclusão de um maior número de domínios locais, resultando nos seguintes exemplos:

- Pórtico com oito trincas em Modo Misto de abertura sujeito ao carregamento de vento - Seção 4.2.1;
- Chapa com dez trincas em Modo I de abertura - Seção 4.2.2;

4.2.1 Pórtico com oito Trincas em Modo Misto de abertura

O problema analisado nesta seção consiste em um pórtico de oito andares, cujos pavimentos apresentam as mesmas propriedades e discretizações adotadas no modelo da Seção 4.1.1, conforme ilustrado na Figura (4.18). As malhas de cada mesoescala e seus respectivos domínios locais foram preservadas, enquanto a malha global foi ampliada de 1008 para 8064 elementos. Consequentemente, o número de nós no modelo global aumentou de 1183 para 9366, resultando em um incremento aproximado de 692% no número de graus de liberdade do sistema. A análise foi realizada sob a hipótese de Estado Plano de Tensões.

Foram avaliadas 8 mesoescalas, posicionadas de acordo com a consideração de trincas nas ligações entre vigas e pilares no lado esquerdo do pórtico. Dessa forma, o modelo não apresenta mais simetria geométrica. As fissuras consideradas nos oito domínios locais são idênticas e definidas pelos parâmetros $a = 7,9057$ e $\beta = 2,8198$ rad.

A variação nos resultados entre as mesoescalas será influenciada pelo carregamento aplicado, aqui representado pela força exercida pelo vento sobre a estrutura. O Apêndice A descreve detalhadamente as etapas de cálculo e as premissas adotadas para a determinação desse carregamento. Como as cargas aplicadas dependem da geometria do modelo, além da velocidade e pressão do vento, o carregamento é variável ao longo da altura do pórtico.

Consequentemente, as cargas concentradas aplicadas em cada andar aumentam em direção ao topo, já que a pressão distribuída ao longo das fachadas é mais intensa nos andares superiores e menos intensa nos inferiores. A Tabela (4.6) apresenta os valores das forças concentradas $F1$ a $F8$, conforme ilustrado na Figura (4.18).

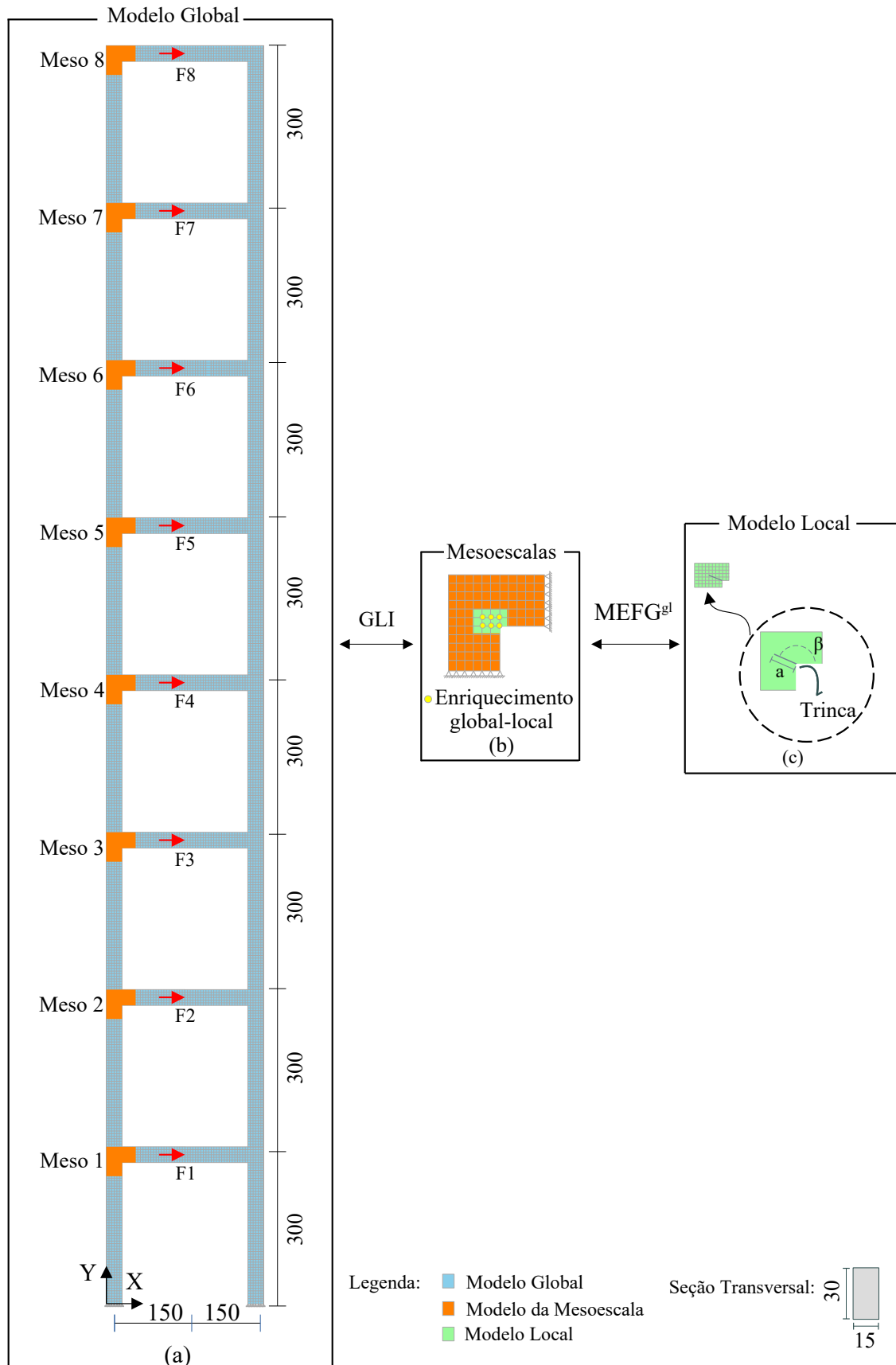


Figura 4.18: Pórtico com oito trincas em modo misto de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade.

Tabela 4.6: Forças concentradas equivalentes a solicitação de vento para o pórtico de 8 pavimentos.

Força concentrada devido à pressão do Vento	Intensidade
F1	686,338
F2	748,457
F3	787,368
F4	816,198
F5	839,284
F6	858,631
F7	875,337
F8	445,035

4.2.1.1 Validação da versão aprimorada dos algoritmos do MEFG^{gl}-GLI

Conforme descrito na Seção (3.2), este trabalho propõe uma versão aprimorada dos algoritmos de solução monolítica e *staggered* do MEFG^{gl}-GLI, aplicados a problemas com múltiplas mesoescalas. A abordagem baseia-se na premissa de interromper o processamento dos domínios que já alcançaram a convergência, permitindo a liberação de recursos computacionais para serem alocados aos modelos remanescentes na análise.

Para validar a eficiência e a precisão dessa versão aprimorada, foi utilizado o modelo de pórtico ilustrado na Figura (4.18). Os resultados foram avaliados ao longo de cinco etapas de propagação das trincas. Como parâmetro de validação da qualidade da solução, foram empregados os fatores de intensidade de tensão pois são adequados para mensurar os impactos nos resultados, devido à sua sensibilidade ao processo solutivo.

Os resultados obtidos pela abordagem monolítica (Algoritmo 3.2 *versus* Algoritmo 3.6) estão detalhados nas Tabelas (4.7) a (4.10). Para a abordagem *staggered* (Algoritmos 3.4 e 3.5 *versus* Algoritmos 3.7 e 3.8), os resultados estão apresentados nas Tabelas (4.11) a (4.14). Na maioria dos casos analisados, as diferenças entre a versão original (proposta por Li *et al.* (2021) e Li *et al.* (2022)) e a versão aprimorada dos algoritmos são observadas apenas na quinta casa decimal. A maior variação registrada foi $\Delta_{max} = 0,00767$, correspondente ao valor de K_1 da mesoescala oito no segundo passo de propagação.

Os resultados indicam que algumas mesoescalas são menos onerosas (quanto mais próximas do topo do pórtico) e convergem com um menor número de iterações, tendo esta redução um impacto no tempo de processamento de maneira diferente para as abordagens monolítica e *staggered*, conforme será discutido posteriormente neste texto. As soluções permanecem inalteradas pelas iterações adicionais necessárias para a convergência dos domínios restantes. Além disso, verifica-se que a solução global do problema não é comprometida. As soluções dos domínios que continuam interagindo com o problema global após a interrupção do processamento dos domínios menos onerosos também não apresentam variações significativas nos fatores de intensidade de tensão.

Tabela 4.7: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 1 e 2.

Mesoescala		1			2		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	625,46245	625,46245	0,00000	668,01264	668,01264	0,00000
	K_{II}	153,37244	153,37244	0,00000	148,75666	148,75666	0,00000
	N. Iter.	6	6	0	6	6	0
Passo 2	K_I	560,92101	560,92073	0,00029	588,84470	588,84467	0,00003
	K_{II}	69,32443	69,32369	0,00073	54,19531	54,19483	0,00048
	N. Iter.	5	6	1	5	6	1
Passo 3	K_I	355,65089	355,65092	0,00003	357,81523	357,81520	0,00003
	K_{II}	43,41788	43,41784	0,00005	25,24664	25,24665	0,00000
	N. Iter.	6	6	0	6	6	0
Passo 4	K_I	332,66382	332,66375	0,00007	298,05556	298,05552	0,00003
	K_{II}	88,02238	88,02189	0,00049	65,66916	65,66856	0,00060
	N. Iter.	5	6	1	5	6	1
Passo 5	K_I	415,50317	415,50054	0,00263	326,48243	326,48269	0,00026
	K_{II}	123,24140	123,24340	0,00199	140,17602	140,17607	0,00005
	N. Iter.	5	6	1	5	6	1

Tabela 4.8: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 3 e 4.

Mesoescala		3			4		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	567,40905	567,40905	0,00000	455,79470	455,79470	0,00000
	K_{II}	133,96049	133,96049	0,00000	114,56839	114,56839	0,00000
	N. Iter.	6	6	0	6	6	0
Passo 2	K_I	498,30893	498,30891	0,00003	401,40178	401,40174	0,00004
	K_{II}	63,21960	63,21915	0,00045	65,64564	65,64524	0,00040
	N. Iter.	5	6	1	5	6	1
Passo 3	K_I	301,46704	301,46696	0,00008	245,25393	245,25400	0,00007
	K_{II}	41,16163	41,16164	0,00001	45,93488	45,93485	0,00003
	N. Iter.	6	6	0	6	6	0
Passo 4	K_I	283,09989	283,09956	0,00032	254,85542	254,85648	0,00106
	K_{II}	87,20758	87,20722	0,00036	75,41672	75,41630	0,00042
	N. Iter.	5	6	1	4	6	2
Passo 5	K_I	376,69789	376,69543	0,00246	316,07294	316,07170	0,00124
	K_{II}	93,82373	93,82626	0,00253	35,78372	35,78484	0,00112
	N. Iter.	5	6	1	4	6	2

Tabela 4.9: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 5 e 6.

Mesoescala		5			6		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	347,92575	347,92575	0,00000	246,14990	246,14952	0,00038
	K_{II}	91,37780	91,37780	0,00000	64,82086	64,81950	0,00136
	N. Iter.	6	6	0	5	6	1
Passo 2	K_I	308,53309	308,53302	0,00007	217,62210	217,62167	0,00043
	K_{II}	58,28916	58,28878	0,00039	44,22018	44,21977	0,00041
	N. Iter.	5	6	1	5	6	1
Passo 3	K_I	189,46084	189,46090	0,00006	131,32092	131,32090	0,00002
	K_{II}	38,91271	38,91268	0,00002	24,64600	24,64582	0,00018
	N. Iter.	6	6	0	5	6	1
Passo 4	K_I	207,24254	207,24289	0,00034	141,07039	141,06977	0,00062
	K_{II}	55,94144	55,94083	0,00061	36,54379	36,54416	0,00037
	N. Iter.	4	6	2	4	6	2
Passo 5	K_I	246,39095	246,39098	0,00003	166,14067	166,13946	0,00121
	K_{II}	12,88690	12,88820	0,00130	10,61445	10,61550	0,00105
	N. Iter.	5	6	1	4	6	2

Tabela 4.10: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo monolítico - Valores correspondentes às mesoescalas 7 e 8.

Mesoescala		7			8		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	150,56174	150,56148	0,00027	57,32900	57,32886	0,00014
	K_{II}	36,46884	36,46822	0,00063	19,55748	19,55750	0,00002
	N. Iter.	5	6	1	5	6	1
Passo 2	K_I	130,36214	130,36209	0,00005	55,09357	55,09308	0,00049
	K_{II}	24,01013	24,01003	0,00011	29,67505	29,67430	0,00075
	N. Iter.	5	6	1	4	6	2
Passo 3	K_I	71,90495	71,90544	0,00050	29,59687	29,59392	0,00295
	K_{II}	6,47129	6,47138	0,00008	-0,02701	-0,02504	0,00197
	N. Iter.	5	6	1	3	6	3
Passo 4	K_I	64,99777	64,99774	0,00003	29,29022	29,28967	0,00055
	K_{II}	15,95325	15,95362	0,00037	3,75526	3,75322	0,00204
	N. Iter.	3	6	3	2	6	4
Passo 5	K_I	79,87915	79,87874	0,00041	32,69056	32,69064	0,00008
	K_{II}	22,00565	22,00595	0,00030	4,91693	4,91827	0,00133
	N. Iter.	5	6	1	3	6	3

Tabela 4.11: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo *staggered* - Valores correspondentes às mesoescalas 1 e 2.

Mesoescala		1			2		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	625,48695	625,48706	0,00011	668,03528	668,03539	0,00011
	K_{II}	153,37363	153,37397	0,00034	148,75795	148,75823	0,00028
	N. Iter. GLI	11	12	1	11	12	1
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 2	K_I	560,93603	560,94353	0,00751	588,87486	588,86726	0,00760
	K_{II}	69,32084	69,31994	0,00090	54,18927	54,19141	0,00213
	N. Iter. GLI	10	15	5	11	15	4
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 3	K_I	355,65848	355,66162	0,00314	357,82471	357,82779	0,00308
	K_{II}	43,41774	43,41706	0,00068	25,24565	25,24483	0,00082
	N. Iter. GLI	10	17	7	10	17	7
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 4	K_I	332,66766	332,67172	0,00405	298,06583	298,06704	0,00121
	K_{II}	88,02602	88,02358	0,00243	65,66707	65,66646	0,00060
	N. Iter. GLI	11	16	5	14	16	2
	N. Iter. <i>staggered</i>	4	4	0	4	4	0
Passo 5	K_I	415,50249	415,49962	0,00287	326,47692	326,47515	0,00177
	K_{II}	123,24989	123,25433	0,00444	140,18345	140,18552	0,00206
	N. Iter. GLI	9	9	0	6	9	3
	N. Iter. <i>staggered</i>	5	5	0	3	5	2

Tabela 4.12: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo *staggered* - Valores correspondentes às mesoescalas 3 e 4.

Mesoescala		3			4		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	567,43115	567,42820	0,00295	455,81218	455,80997	0,00221
	K_{II}	133,96192	133,96195	0,00003	114,56966	114,56966	0,00001
	N. Iter. GLI	10	12	2	9	12	3
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 2	K_I	498,32005	498,32750	0,00745	401,41023	401,41631	0,00607
	K_{II}	63,21756	63,21645	0,00110	65,64393	65,64307	0,00086
	N. Iter. GLI	10	15	5	10	15	5
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 3	K_I	301,47353	301,47643	0,00290	245,25904	245,26094	0,00190
	K_{II}	41,16238	41,16161	0,00077	45,93663	45,93613	0,00050
	N. Iter. GLI	10	17	7	10	17	7
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 4	K_I	283,10803	283,10864	0,00060	254,86144	254,85946	0,00198
	K_{II}	87,21057	87,21029	0,00029	75,41769	75,41952	0,00184
	N. Iter. GLI	11	16	5	7	16	9
	N. Iter. <i>staggered</i>	4	4	0	3	4	1
Passo 5	K_I	376,70344	376,70067	0,00276	316,07372	316,07603	0,00231
	K_{II}	93,83007	93,83531	0,00524	35,78948	35,78809	0,00139
	N. Iter. GLI	9	9	0	6	9	3
	N. Iter. <i>staggered</i>	5	5	0	3	5	2

Tabela 4.13: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo *staggered* - Valores correspondentes às mesoescalas 5 e 6.

Mesoescala		5			6		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	347,93537	347,93738	0,00201	246,14845	246,15779	0,00934
	K_{II}	91,37856	91,37882	0,00026	64,82097	64,82025	0,00072
	N. Iter. GLI	8	12	4	7	12	5
	N. Iter. <i>staggered</i>	4	5	1	3	5	2
Passo 2	K_I	308,53989	308,54416	0,00426	217,62794	217,63055	0,00260
	K_{II}	58,28753	58,28692	0,00061	44,21876	44,21876	0,00000
	N. Iter. GLI	10	15	5	11	15	4
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 3	K_I	189,46494	189,46524	0,00030	131,32651	131,32582	0,00068
	K_{II}	38,91452	38,91470	0,00018	24,64600	24,64669	0,00069
	N. Iter. GLI	10	17	7	10	17	7
	N. Iter. <i>staggered</i>	4	5	1	5	5	0
Passo 4	K_I	207,24613	207,24541	0,00072	141,07250	141,07382	0,00132
	K_{II}	55,94342	55,94434	0,00093	36,54685	36,54556	0,00129
	N. Iter. GLI	8	16	8	8	16	8
	N. Iter. <i>staggered</i>	3	4	1	4	4	0
Passo 5	K_I	246,39402	246,39486	0,00085	166,14410	166,14364	0,00046
	K_{II}	12,88819	12,88886	0,00067	10,61491	10,61592	0,00101
	N. Iter. GLI	6	9	3	6	9	3
	N. Iter. <i>staggered</i>	3	5	2	3	5	2

Tabela 4.14: Análise comparativa dos fatores de intensidade de tensão obtidos a partir das versões original e aprimorada do algoritmo *staggered* - Valores correspondentes às mesoescalas 7 e 8.

Mesoescala		7			8		
Versão do Algoritmo	Aprimorada	Original	Diferença dos valores - Δ	Aprimorada	Original	Diferença dos valores - Δ	
Passo 1	K_I	150,56548	150,56648	0,00100	57,33057	57,33038	0,00018
	K_{II}	36,46843	36,46857	0,00014	19,55769	19,55765	0,00004
	N. Iter. GLI	8	12	4	10	12	2
	N. Iter. <i>staggered</i>	4	5	1	4	5	1
Passo 2	K_I	130,36524	130,36785	0,00261	55,08662	55,09429	0,00767
	K_{II}	24,00952	24,00977	0,00026	29,67412	29,67391	0,00021
	N. Iter. GLI	10	15	5	6	15	9
	N. Iter. <i>staggered</i>	4	5	1	3	5	2
Passo 3	K_I	71,90904	71,90875	0,00029	29,59591	29,59492	0,00099
	K_{II}	6,47063	6,47155	0,00093	-0,02604	-0,02489	0,00116
	N. Iter. GLI	8	17	9	5	17	12
	N. Iter. <i>staggered</i>	4	5	1	2	5	3
Passo 4	K_I	65,00116	65,00020	0,00095	29,29275	29,29063	0,00213
	K_{II}	15,95674	15,95424	0,00250	3,75410	3,75354	0,00056
	N. Iter. GLI	16	16	0	3	16	13
	N. Iter. <i>staggered</i>	4	4	0	2	4	2
Passo 5	K_I	79,88542	79,88152	0,00390	32,69149	32,69232	0,00083
	K_{II}	22,00663	22,00671	0,00009	4,91946	4,91712	0,00234
	N. Iter. GLI	5	9	4	4	9	5
	N. Iter. <i>staggered</i>	4	5	1	2	5	3

As mesoescalas 1 e 2 correspondem aos problemas mais onerosos, visto que a influência dos campos de tensões e deformações mais pronunciados em seus domínio locais é incorporada ao modelo por meio do enriquecimento do MEFG^{gl}. Consequentemente, observa-se uma maior variação de rigidez entre essas mesoescalas e o problema global, quando comparada com a diferença aferida para as demais mesoescalas.

Ao analisar o desempenho computacional na abordagem monolítica, observa-se que o número total de iterações¹ exigidas pelas versões original e aprimorada foi de 240 e 195, respectivamente, o que corresponde a uma redução de aproximadamente 18,7%. Na abordagem monolítica todas as iterações envolvem a execução completa das etapas de acoplamento entre os modelos, portanto, a redução no número de iterações traduz-se de forma praticamente linear na diminuição do tempo de processamento, resultando em uma redução aproximada de 18,2%.

Na abordagem *staggered*, as iterações GLI foram reduzidas de 552 para 354, enquanto as iterações *staggered* diminuíram de 192 para 149, representando ganhos de aproximadamente 35,9% e 22,4%, respectivamente. O esforço computacional relacionado ao processamento do modelo global permanece constante, independentemente do número de mesoescalas consideradas, e, portanto, não é impactado pela interrupção do processamento dos domínios já convergidos. Consequentemente, como na abordagem *staggered* o modelo global é processado com maior frequência em comparação à abordagem monolítica (cerca de 80,0% mais vezes), a redução no número total de iterações não se traduz integralmente em diminuição do tempo de processamento, resultando em uma redução aproximada de 12,8% no tempo total de análise. Assim, embora a redução no número total de iterações seja mais significativa na abordagem *staggered*, o aprimoramento do algoritmo de solução proporciona maior redução de custos computacionais quando a abordagem monolítica é adotada.

Considerando que as versões aprimoradas mantiveram a precisão dos resultados e tendo sido observada uma redução do esforço computacional exigido em comparação às versões originais, conclui-se que a proposição apresentada neste trabalho está validada. Dessa forma, as versões aprimoradas serão adotadas nas análises de desempenho subsequentes.

4.2.1.2 Trincas Estáticas

Nesta seção, avalia-se o desempenho computacional do MEFG^{gl}-GLI em função do número de mesoescalas e processadores empregados na análise. A partir de um modelo com dois subdomínios, mesoescalas 1 e 2 - Figura (4.18), os demais domínios locais são progressivamente incorporados à análise, de baixo para cima. Assim, o problema analisado com três subdomínios abrange as mesoescalas 1, 2 e 3, e essa incorporação segue até a obtenção do modelo completo com as oito mesoescalas. Inicialmente, foram consideradas

¹O número total de iterações refere-se à soma algébrica das iterações realizadas por cada mesoescala, e não ao número total de iterações efetivadas na análise.

trincas estáticas em todas as análises.

Primeiramente são analisados os tempo e as iterações das simulações sequenciais, para depois apresentar o desempenho das simulações paralelizadas. As Tabelas (4.15) e (4.16) apresentam o número de iterações realizadas e o tempo total de processamento dos modelos de forma sequencial, ou seja, utilizando apenas um processador ($n_p = 1, 0$). A fim de isolar as condições de *hardware* do computador utilizado nas análises, os tempos de execução dos modelos são apresentados de maneira relativa, correspondendo a razão entre o tempo gasto em cada análise e o tempo de execução do modelo com duas mesoescalas via abordagem monolítica (primeira linha de resultados na Tabela (4.15)).

Em todos os casos, o tempo total de execução foi dividido em duas parcelas: a primeira refere-se ao tempo gasto na execução da parte sequencial dos algoritmos, correspondente ao processamento do modelo global, enquanto a segunda corresponde ao tempo dedicado ao cálculo das mesoescalas e seus respectivos domínios locais, representando o trecho paralelizável dos algoritmos.

Tabela 4.15: Pórtico com 8 mesoescalas: Número de iterações e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas estáticas.

Número de Mesoescalas	N. Iter.	Tempo relativo		
		Trecho Sequencial	Trecho Paralelizável	Total
2	6	0,361	0,639	1,000
3	6	0,350	0,951	1,301
4	6	0,358	1,234	1,592
5	6	0,356	1,466	1,822
6	6	0,352	1,787	2,139
7	6	0,375	2,224	2,599
8	6	0,357	2,276	2,633

Tabela 4.16: Pórtico com 8 mesoescalas: Número de iterações e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem *staggered*. Trincas estáticas.

Número de Mesoescalas	N. Iter. GLI	N. Iter. <i>staggered</i>	Tempo relativo		
			Trecho Sequencial	Trecho Paralelizável	Total
2	9	4	0,542	0,553	1,096
3	9	4	0,521	0,816	1,338
4	9	4	0,527	1,095	1,621
5	9	4	0,529	1,345	1,875
6	9	4	0,528	1,540	2,068
7	9	4	0,545	1,846	2,391
8	12	4	0,706	2,210	2,916

Para a abordagem monolítica, o número de iterações necessárias permanece constante, igual a seis, para todas as análises. Um comportamento semelhante é observado para a

abordagem *staggered*, na qual o número de iterações *staggered* é invariável, mantendo-se sempre em quatro, enquanto as iterações GLI variam apenas na análise do modelo com oito mesoescalas, passando de nove (valor observado para os modelos com duas a sete mesoescalas) para doze execuções.

Conforme detalhado na Seção (2.1), as condições de acoplamento entre os modelos via algoritmo GLI são determinadas pela variação de rigidez entre as escalas global e meso. Assim, infere-se que o número de iterações realizadas seja determinado pelas condições de convergência do domínio local com maior penalização de rigidez. Observa-se que a adição de novas mesoescalas, com penalização de rigidez igual ou inferior, não altera significativamente o número de iterações necessárias para a convergência. Esses dados corroboram as considerações apresentadas na Seção (4.2.1.1) sobre a versão aprimorada dos algoritmos de solução do MFG^{gl}-GLI, indicando que a interação mútua entre os diferentes modelos locais não representa uma preocupação relevante.

A Figura (4.19) ilustra o tempo total gasto em cada análise relativamente ao tempo de execução sequencial da abordagem monolítica com duas mesoescalas. Para maior clareza, esses dados estão também registrados nas Tabelas (4.17) e (4.18). Considerando as análises por processamento sequencial ($n_p = 1$), a abordagem monolítica inicialmente apresenta menores tempos de execução. Contudo, à medida que o número de mesoescalas aumenta, a diferença entre essa abordagem e a *staggered* diminui progressivamente, ocorrendo inversão do comportamento nos casos com seis e sete mesoescalas. Essa tendência não é mantida no modelo com oito mesoescalas, pois, como observado na Tabela (4.16), o número de iterações permanece constante na abordagem monolítica, enquanto aumenta na abordagem *staggered*.

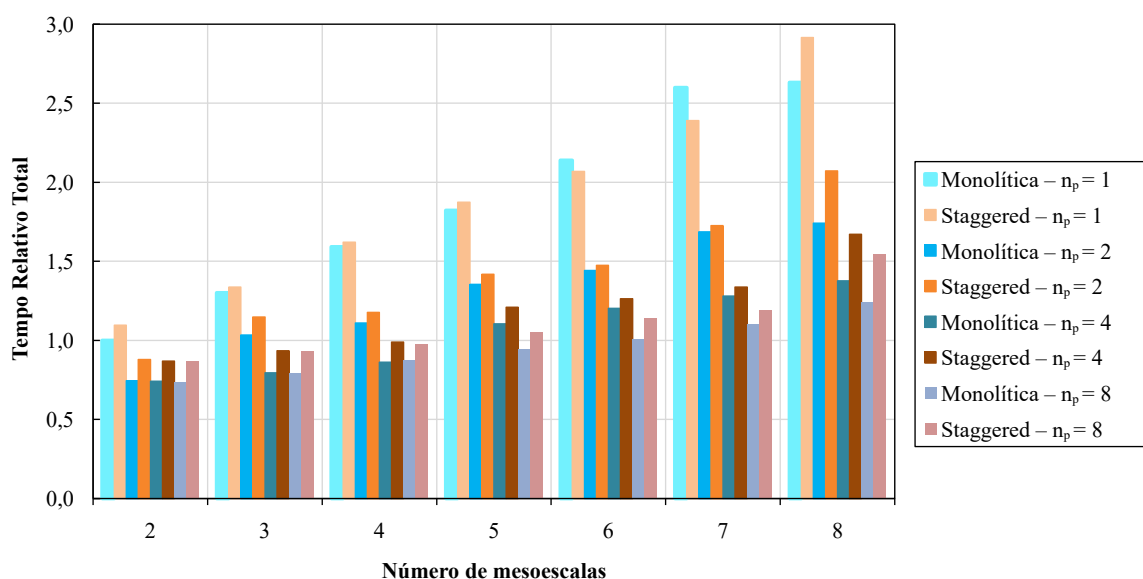


Figura 4.19: Pórtico com 8 mesoescalas: Representação gráfica do dispêndio de tempo total em função do número de mesoescalas e processadores. Trincas estáticas.

Tabela 4.17: Pórtico com 8 mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Tempo			Medidas de desempenho		
		Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	2	0,359	0,386	0,745	1,343	0,671	1,470
	3	0,361	0,673	1,035	1,257	0,629	1,576
	4	0,350	0,760	1,110	1,434	0,717	1,633
	5	0,351	1,003	1,354	1,346	0,673	1,673
	6	0,358	1,084	1,442	1,483	0,742	1,717
	7	0,354	1,333	1,687	1,541	0,770	1,748
	8	0,354	1,389	1,743	1,511	0,755	1,761
	4	2	0,359	0,382	0,742	1,349	0,337
4	3	0,361	0,434	0,794	1,637	0,409	2,215
	4	0,351	0,511	0,862	1,846	0,461	2,389
	5	0,350	0,754	1,104	1,650	0,413	2,523
	6	0,357	0,846	1,203	1,778	0,445	2,677
	7	0,359	0,922	1,280	2,030	0,507	2,793
	8	0,363	1,013	1,376	1,913	0,478	2,844
	8	2	0,361	0,376	0,737	1,358	0,170
8	3	0,360	0,433	0,793	1,641	0,205	2,777
	4	0,354	0,523	0,876	1,816	0,227	3,109
	5	0,350	0,592	0,942	1,933	0,242	3,381
	6	0,354	0,651	1,005	2,128	0,266	3,716
	7	0,376	0,725	1,101	2,360	0,295	3,982
	8	0,375	0,864	1,240	2,123	0,265	4,106

Tabela 4.18: Pórtico com 8 mesoescalas: Medidas de desempenho inerentes a abordagem *staggered* considerando o tempo total da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Tempo			Medidas de desempenho		
		Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	2	0,540	0,339	0,879	1,247	0,623	1,338
	3	0,522	0,626	1,148	1,165	0,583	1,439
	4	0,525	0,652	1,176	1,378	0,689	1,510
	5	0,530	0,888	1,418	1,322	0,661	1,560
	6	0,532	0,943	1,475	1,403	0,701	1,593
	7	0,560	1,166	1,726	1,385	0,693	1,629
	8	0,706	1,366	2,072	1,408	0,704	1,610
	4	2	0,536	0,333	0,868	1,262	0,315
3		0,537	0,398	0,934	1,432	0,358	1,844
4		0,528	0,462	0,990	1,638	0,409	2,026
5		0,524	0,686	1,211	1,549	0,387	2,165
6		0,533	0,729	1,262	1,638	0,410	2,265
7		0,539	0,797	1,337	1,788	0,447	2,375
8		0,702	0,969	1,671	1,745	0,436	2,316
8		2	0,533	0,339	0,872	1,256	0,157
	3	0,540	0,391	0,931	1,436	0,180	2,146
	4	0,526	0,449	0,976	1,662	0,208	2,444
	5	0,529	0,522	1,051	1,783	0,223	2,688
	6	0,535	0,608	1,143	1,810	0,226	2,870
	7	0,544	0,646	1,189	2,010	0,251	3,082
	8	0,729	0,819	1,547	1,885	0,236	2,968

Como as iterações GLI são as únicas a aumentar, este comportamento pode ser atribuído ao processo de imposição das condições de contorno entre as escalas global e meso. A mesoescala 8, situada no topo do pórtico, apresenta, em comparação com as demais, uma interface reduzida entre as escalas global e meso, devido à interrupção da continuidade do elemento estrutural. Como resultado, o conjunto de nós nos quais os deslocamentos são prescritos diminui aproximadamente 33,0% em relação as demais mesoescalas, penalizando as condições de acoplamento.

A convergência do GLI não é empregada como parâmetro que controla a incorporação do domínio local no algoritmo monolítico, tornando esta abordagem menos afetada pelas condições de contorno entre as escalas meso e global. As próprias iterações global-local do MEFG^{gl} tratam a baixa qualidade das informações transferidas entre as escalas do problema. Deste modo, como a mesoescala 8 é pouco onerosa, sua adição a análise não altera o número de iterações e aumenta pouco o tempo de análise, que parte de 2,599 para 2,633.

O tempo de processamento das análises apresenta uma redução progressiva a medida que mais processadores são adicionados, estabilizando-se quando sua quantidade se iguala ao número de mesoescalas. Esse comportamento é ilustrado na Figura (4.19) para os modelos com duas, três e quatro mesoescalas.

A Figura (4.20) ilustra o aumento do tempo de análise, à medida que, mais mesoescalas são incorporadas ao modelo, destacando-se a divisão do tempo entre as seções sequenciais e paralelizáveis do código. Conforme esperado, a parte relativa a resolução do modelo global (trecho sequencial) permanece constante, exceto para o caso da abordagem *staggered* com oito mesoescalas conforme discutido anteriormente. Por outro lado, o tempo dedicado ao cálculo das mesoescalas (trecho paralelizável) aumenta progressivamente.

A medida que o número de processadores se torna dominante em relação ao número de domínios locais, a taxa de variação do tempo em função do número de mesoescalas é suavizada, e a curva do tempo associado ao trecho paralelizável se aproxima de uma linha horizontal, embora nunca a atinja completamente.

Os parâmetros de desempenho, apresentados nas Tabelas (4.17) e (4.18) e na Figura (4.21), consideram, para o cálculo do *SpeedUp* (SU - Equação (2.30)) e, conseqüentemente, para a eficiência (E - Equação (2.31)), o tempo total da análise. Em todas as análises, observou-se $SU > 1,0$, indicando que a adoção das técnicas de paralelismo não comprometeram o processo. Foram alcançados valores de $SU = 2,306$ e de $SU = 2,010$ para as abordagens monolítica e *staggered*, respectivamente.

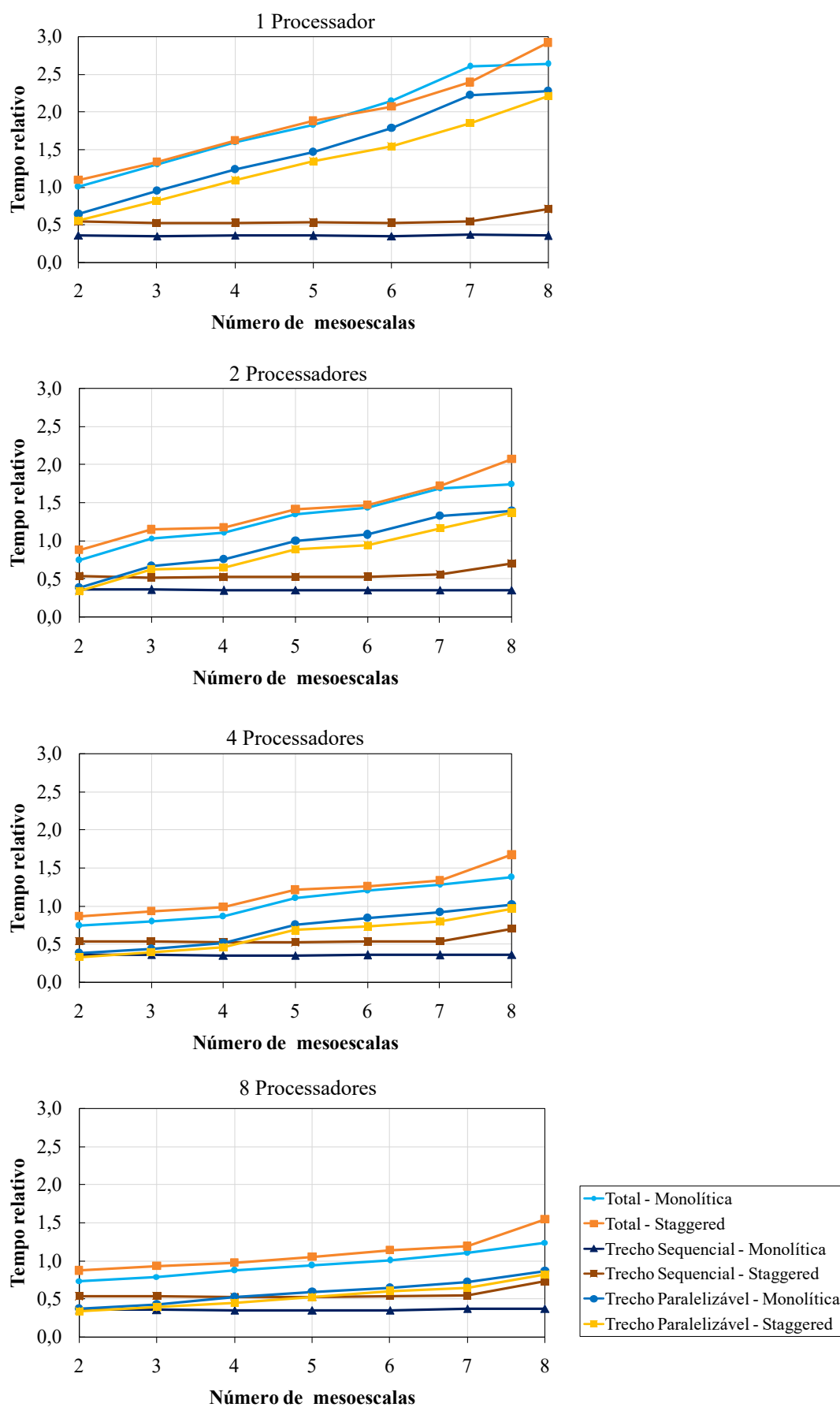


Figura 4.20: Pórtico com 8 mesoescalas: Parcelas de tempo referentes aos trechos sequenciais e paralelizáveis dos algoritmos do MEFG^{gl}-GLI. Trincas estáticas.

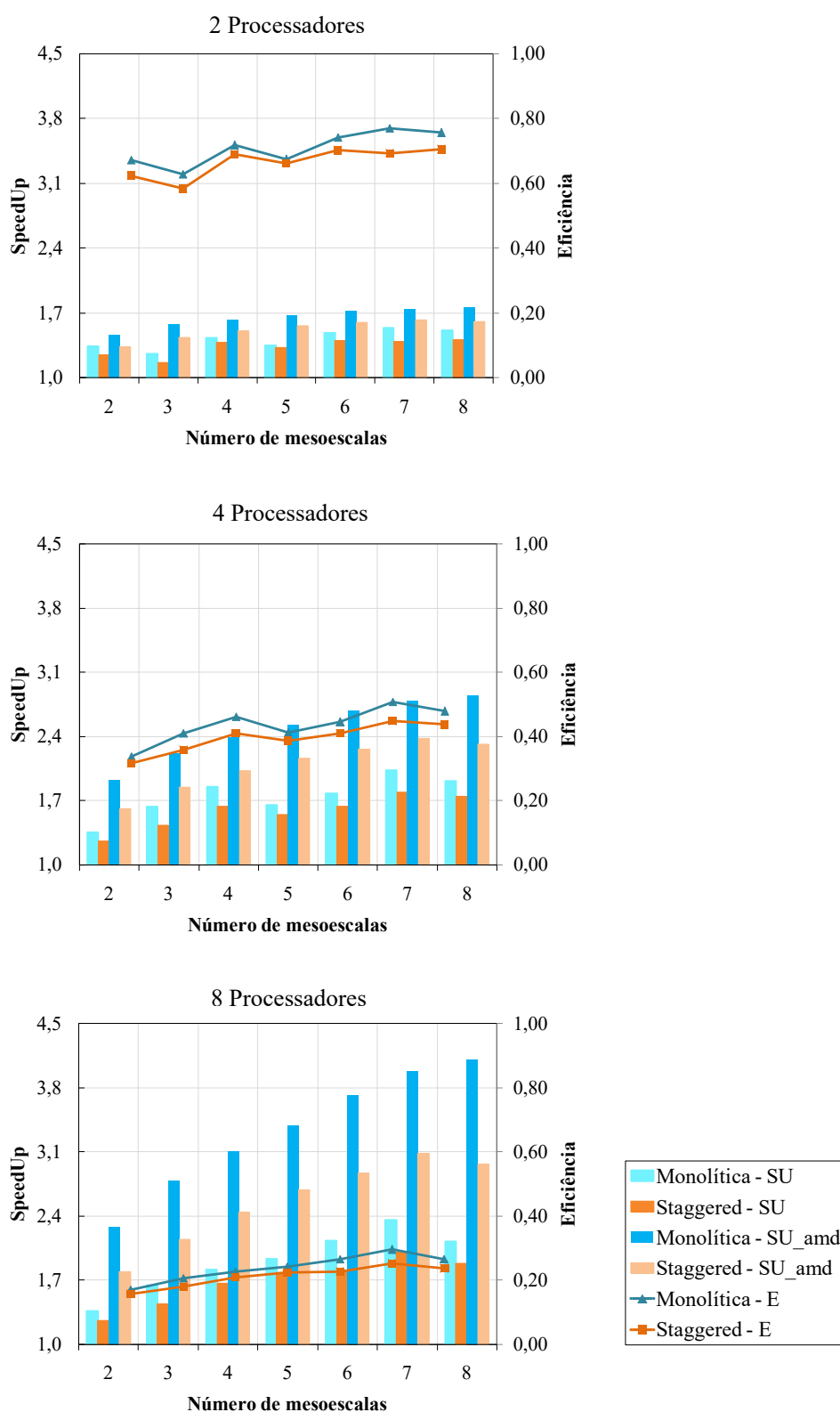


Figura 4.21: Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise para cálculo do *speedUp*. Trincas estáticas

Os dados também demonstram que o *SpeedUp* referente a abordagem monolítica é sempre superior ao observado para a abordagem *staggered*. Esse resultado é atribuído a metodologia de solução inerente as duas abordagens, no algoritmo *staggered*, o número total de análises global-local é menor, pois somente ocorre após a convergência GLI. O tempo dedicado ao trecho sequencial torna-se maior em relação ao paralelizado. São, portanto, atenuados os ganhos advindos das técnicas de paralelismo. Esse aspecto é evidenciado pelos valores do *SpeedUp* proposto por Amdahl (1967) (SU_{amd} - Equação (2.32)) calculados a partir da proposta de Amdahl (1967), que considera as frações do código passíveis de paralelização em comparação com as que são essencialmente sequenciais.

O SU_{amd} apresenta uma correção considerável ao valor idealizado do *SpeedUp*, que, em uma situação idealizada (situação na qual todo o código é paralelizado, ou seja, $S = 0$ e $P = 1$), deveria atingir um valor igual ao número de processadores empregados na análise, expressando um comportamento linear. Contudo, embora o SU_{amd} limite o ganho das técnicas de paralelismo ao considerar a parte sequencial do código, a Equação (2.32) prevê que o tempo de processamento do trecho paralelo seja reduzido de forma diretamente proporcional ao número de processadores, o que representaria uma eficiência de $E = 100\%$ para esta parte do algoritmo. Alguns fatores não corroboram essa premissa, como os casos em que o número de processadores excede o número de mesoescalas, o que impede que a eficiência acompanhe a adição de núcleos de processamento devido a ausência de tarefas a serem executadas por estes. Outro limitante ocorre nas condições de acesso as variáveis compartilhadas pelas *threads* de todos os processos de solução, uma vez que, à medida que o número de modelos processados paralelamente cresce, a probabilidade de tentativas de acesso simultâneo aumenta, resultando em maiores tempos de espera.

De maneira geral, observa-se um aumento dos parâmetros de eficiência (E) até o número de processadores igualar o número de mesoescalas, reduzindo a ociosidade dos processadores de forma que o máximo potencial da paralelização tende a ser atingido. Após esse ponto de equilíbrio, quando o número de mesoescalas excede o número de processadores, há uma queda nos parâmetros de desempenho devido a formação de filas para a execução dos processamentos. No entanto, a medida que o número de mesoescalas aumenta, tanto no *SpeedUp* quanto na eficiência (E), observa-se uma retomada do crescimento, visto que, mesmo com o surgimento das filas, a porcentagem do código passível de paralelização se amplia, gerando maiores possibilidades de ganhos.

A análise com dois processadores e cinco mesoescalas segue um comportamento diferente. Nesse caso, a distribuição das tarefas entre os processadores é de 2, 2 e 1, e observou-se que o tempo de processamento dos cinco subdomínios é bastante semelhante. Dessa forma, uma perda de eficiência é gerada pelo desbalanceamento da carga computacional. No modelo com sete mesoescalas, que também apresenta um número ímpar de problemas locais, em relação ao número par de processadores, nota-se uma tendência semelhante, mas com efeito reduzido devido à mesoescala sete, ser menos onerosa que as

anteriores adicionadas ao problema.

Nos modelos com oito mesoescalas, observa-se uma queda sutil no *SpeedUp* e, conseqüentemente, na eficiência (E - Equação (2.31)). Este comportamento é atribuído ao número de mesoescalas se igualar ao número de processadores do computador, resultando em um processador desempenhando função dupla. Além de ser responsável pela *thread* endereçada a ele, esse processador também exerce atividades de gerenciamento do processo de execução das demais *threads* nos outros processadores.

Tabela 4.19: Pórtico com 8 mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Abordagem Monolítica		Abordagem <i>staggered</i>	
		SU	E	SU	E
2	2	1,657	0,828	1,633	0,817
	3	1,413	0,706	1,303	0,652
	4	1,624	0,812	1,680	0,840
	5	1,463	0,731	1,515	0,758
	6	1,649	0,824	1,634	0,817
	7	1,669	0,835	1,583	0,791
	8	1,639	0,819	1,618	0,809
	4	2	1,673	0,418	1,664
3		2,193	0,548	2,053	0,513
4		2,412	0,603	2,370	0,593
5		1,944	0,486	1,960	0,490
6		2,113	0,528	2,112	0,528
7		2,413	0,603	2,314	0,579
8		2,246	0,561	2,281	0,570
8		2	1,700	0,213	1,634
	3	2,197	0,275	2,086	0,261
	4	2,360	0,295	2,436	0,305
	5	2,477	0,310	2,575	0,322
	6	2,743	0,343	2,535	0,317
	7	3,066	0,383	2,859	0,357
	8	2,633	0,329	2,699	0,337

Para avaliar de maneira mais direta os ganhos decorrentes das técnicas de paralelismo, a Tabela (4.19) e a Figura (4.22) apresentam as medidas de desempenho nas quais foi considerado apenas o tempo correspondente ao trecho paralelizável da análise para o cálculo do *SpeedUp* e da eficiência (E). De maneira geral, o comportamento dos parâmetros foi preservado, com seus valores absolutos aumentando consideravelmente, resultando em ganhos superiores a 20,0% para a eficiência, E .

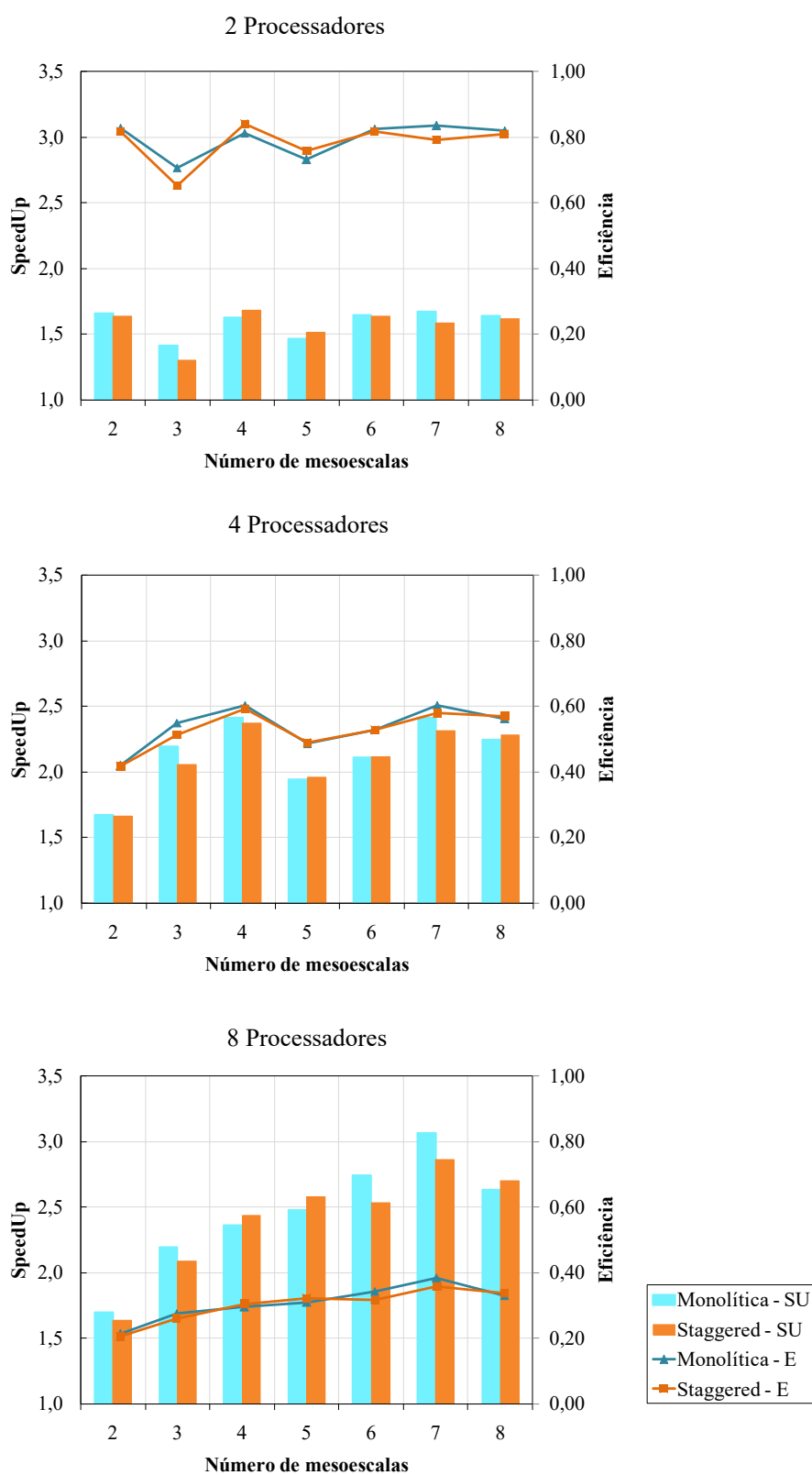


Figura 4.22: Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.

4.2.1.3 Trincas em propagação

O modelo contendo as oito mesoescalas foi reprocessado, desta vez, considerando cinco passos de propagação para todas as trincas, com o objetivo de avaliar a manutenção dos ganhos obtidos nos casos em que foram consideradas trincas estáticas. Nesse contexto, é realizada uma comparação entre o desempenho do processamento sequencial *versus* paralelo. Os tempos relativos obtidos para as abordagens monolítica e *staggered* são apresentados nas Tabelas (4.20) e (4.21), respectivamente. Considerou-se como base de referência o tempo de execução do modelo processado sequencialmente por meio da abordagem monolítica. Estes dados são representados graficamente na Figura (4.23).

Tabela 4.20: Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$) para a abordagem monolítica. Trincas em propagação

Número de Processadores	Tempo		
	Trecho Sequencial	Trecho Paralelizável	Total
1	0,091	0,909	1,000
8	0,088	0,304	0,392

Tabela 4.21: Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$) para a abordagem staggered. Trincas em propagação.

Número de Processadores	Tempo		
	Trecho Sequencial	Trecho Paralelizável	Total
1	0,243	0,921	1,165
8	0,252	0,379	0,631

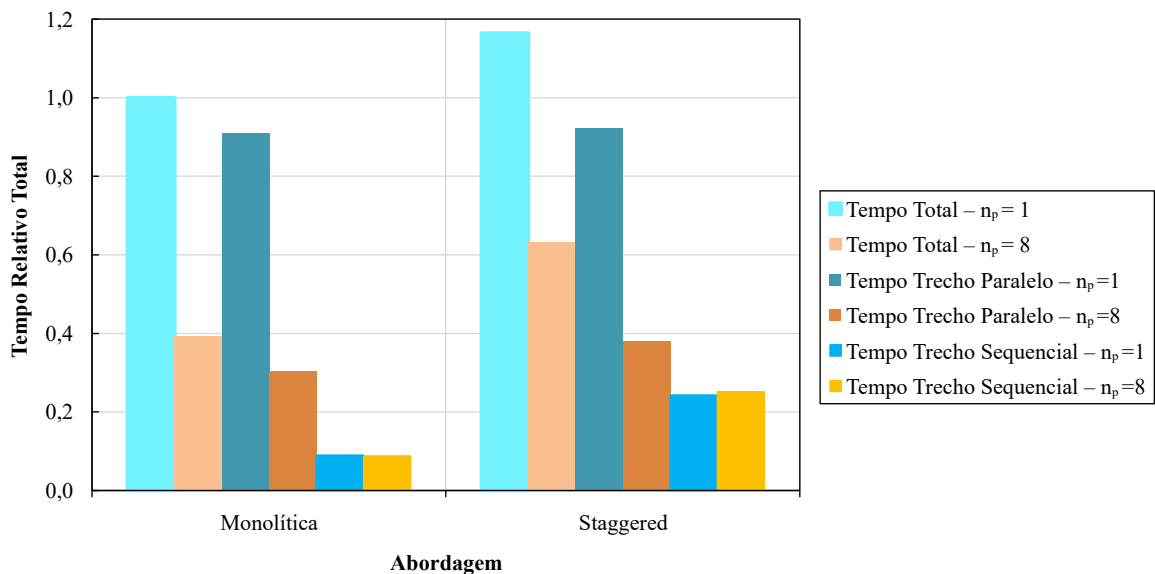


Figura 4.23: Pórtico com 8 mesoescalas: Comparativo do dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) e paralelo ($n_p = 8$). Trincas em propagação.

Com o aumento dos ciclos de solução global-local, que acompanham os passos de propagação da trinca, verifica-se uma expansão na proporção de tempo alocada ao trecho paralelizável, atingindo 90,9% e 79,1% do tempo total de análise para as abordagens monolítica e *staggered*, respectivamente. Ao comparar esses resultados com os obtidos para o caso de trincas estáticas, nota-se que a redução da parcela de tempo sequencial gerou um impacto significativo, promovendo um aumento no *SpeedUp* e na eficiência da abordagem monolítica. Esse comportamento também pode ser observado a partir do *SpeedUp* do tempo do trecho paralelizado comparativamente ao *SU* (tempo total), conforme indicado pelos parâmetros *SU_tr_paralelizável* e *E_tr_paralelizável*, apresentados na Tabela (4.22) e ilustrados na Figura (4.24).

O mesmo ganho não é observado na abordagem *staggered*, resultado de dois fatores principais: a menor proporção de tempo dedicada ao trecho paralelizável e o maior número de iterações realizadas nas mesoescalas menos onerosas em comparação à abordagem monolítica. Esse segundo fator decorre do retardamento na execução dos ciclos global-local do MEF gl na abordagem *staggered*, o que prolonga o processamento dos domínios menos onerosos, uma vez que cada domínio requer um número mínimo de iterações global-local para atingir a convergência. Essas observações estão refletidas nos parâmetros *SU_tr_paralelizável* e *E_tr_paralelizável*.

Tabela 4.22: Pórtico com 8 mesoescalas: Medidas de desempenho considerando 5 passos de propagação de trinca.

Abordagem	Medidas de desempenho				
	SU	SU_tr_paralelizável	E	E_tr_paralelizável	SU_amd
Monolítica	2,552	2,995	0,319	0,374	4,897
<i>Staggered</i>	1,846	2,429	0,231	0,304	3,248

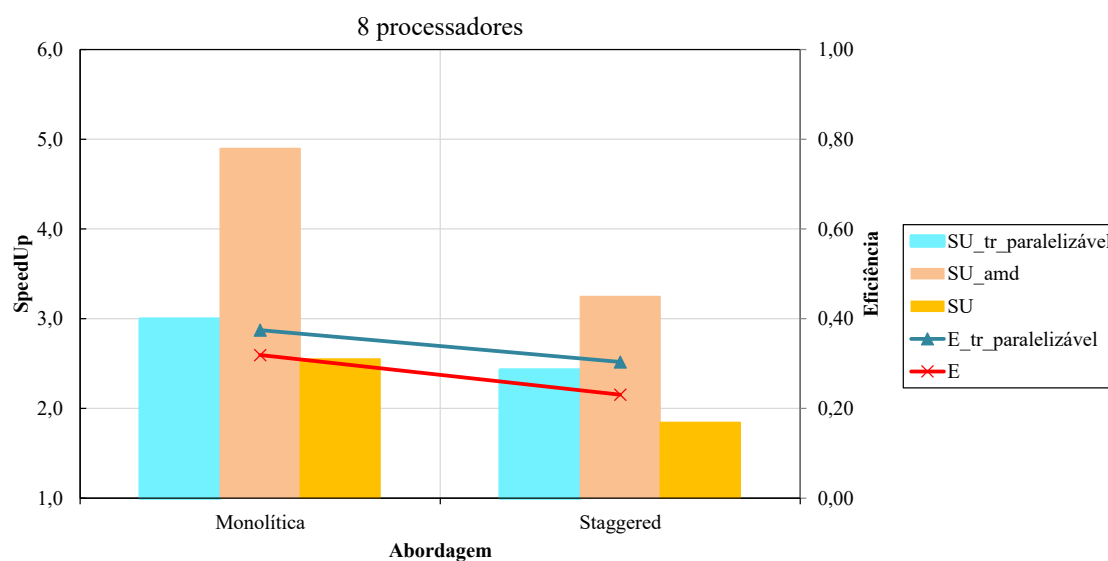


Figura 4.24: Pórtico com 8 mesoescalas: Representação gráfica das medidas de desempenho considerando 5 passos de propagação de trinca.

A eficiência total da paralelização observada é de $E = 31,9\%$ e $E = 23,1\%$ para as abordagens monolítica e *staggered*, respectivamente. Destaca-se que este parâmetro não mensura a redução do tempo proporcionada pelo aumento do número de processadores, mas sim a eficiência com que os recursos computacionais estão sendo utilizados. A eficiência reflete a proporção de ganho no tempo de execução obtido em relação ao número de processadores empregados. O *SpeedUp* é o parâmetro que quantifica o ganho de desempenho ao se executar o algoritmo em um sistema com múltiplos processadores em comparação com sua execução em um único processador, representando diretamente a aceleração conquistada na execução do código.

4.2.2 Chapa com dez trincas em Modo I de abertura

A Figura (4.25) apresenta o problema analisado nesta seção, que consiste em uma chapa submetida a tração axial, mantendo-se o mesmo estado de tensões e as condições de contorno do problema descrito na Seção (4.1.2). Dessa forma, espera-se que todas as trincas se propaguem em Modo *I* puro, apresentando comportamento similar entre si. O objetivo deste exemplo é avaliar a influência direta do aumento no número de mesoescalas. Para tal, optou-se por uma configuração geométrica e de solicitação que permite que todas as mesoescalas apresentem o mesmo comportamento. Diferentemente do problema da Seção (4.2.1), espera-se que, isoladamente, cada mesoescala tenha o mesmo impacto ou ônus sobre o processo iterativo de solução.

A distância entre as mesoescalas 1, 5, 6 e 10 em relação às bordas inferior e superior da chapa, que correspondem às arestas onde as tensões são aplicadas, é de 266. Esse mesmo espaçamento foi aplicado ao longo do eixo y entre cada mesoescala. As dimensões e a discretização de cada mesoescala, bem como de seus respectivos domínios locais, não foram alteradas em relação aos apresentados na Seção (4.1.2). A malha do modelo global foi expandida de 1350 para 7890 elementos, resultando em um aumento de aproximadamente 484% no número de graus de liberdade. A malha do domínio local ilustrada na Figura (4.25 - c) refere-se à discretização gerada no primeiro passo de propagação da trinca, correspondente à dimensão da fissura $a = 10,0$.

O exemplo analisado nesta seção foi processado considerando três configurações geométricas distintas, obtidas por meio da inclusão progressiva das descontinuidades, resultando em:

- Modelo com duas mesoescalas: considera-se as fissuras localizadas a meia altura da chapa ($y = 920,5$), abrangendo as mesoescalas 3 e 8;
- Modelo com seis mesoescalas: incorpora-se ao modelo as fissuras dispostas imediatamente acima e abaixo do modelo de duas mesoescalas, abrangendo os subdomínios 2, 3, 4, 7, 8 e 9;
- Modelo com dez mesoescalas: considera-se o modelo completo, incluindo todas as fissuras e mesoescalas representadas na Figura (4.25).

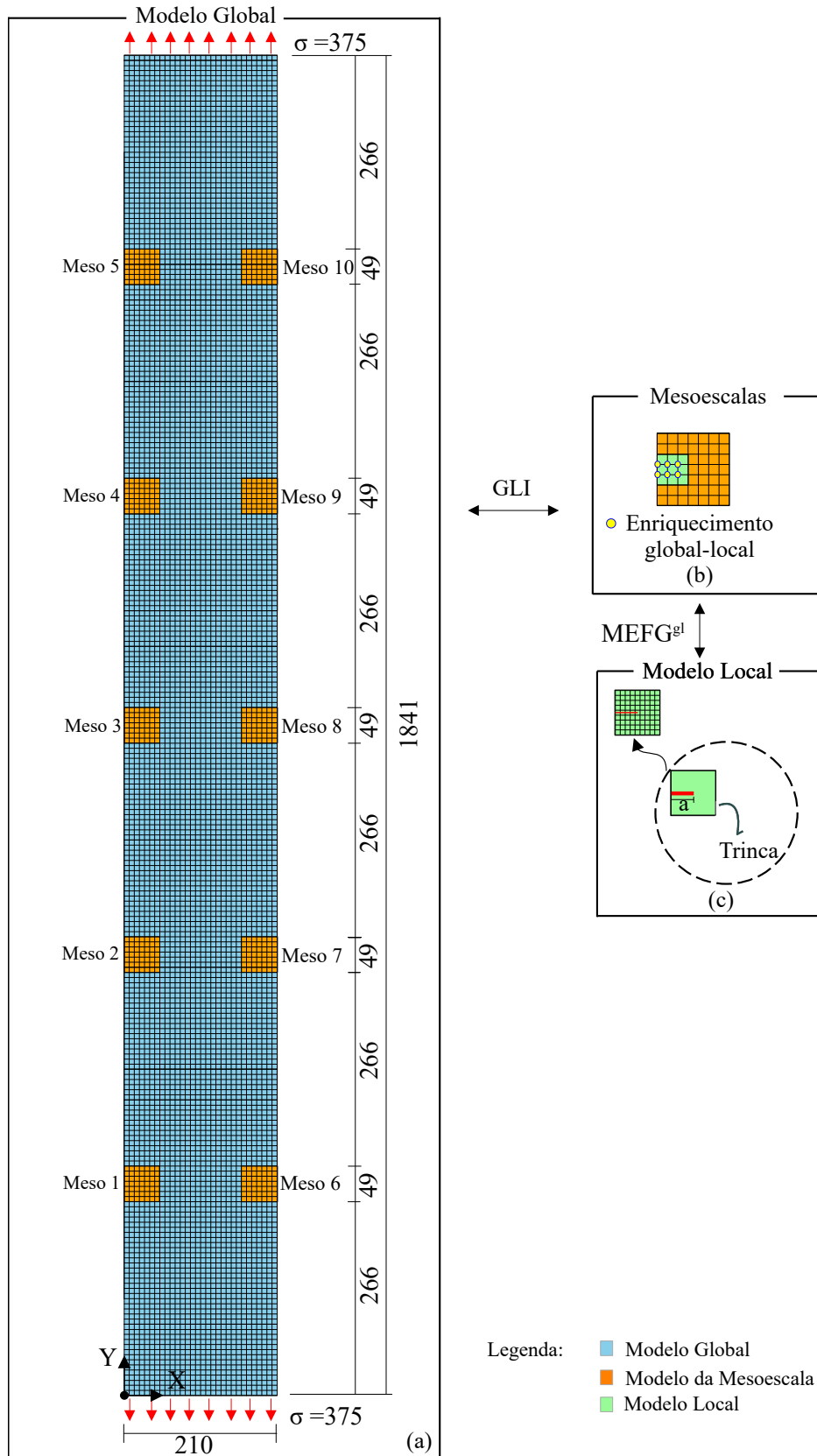


Figura 4.25: Chapa com dez trincas em Modo I de abertura: (a) Discretização do modelo global e as respectivas condições de contorno; (b) Discretização da mesoescala, com indicação esquemática dos nós enriquecidos pela solução do domínio local; (c) Discretização do domínio local, destacando os parâmetros associados à descontinuidade. Modelos das escalas meso e local associados às trincas da direita, podem ser obtidos por simetria.

4.2.2.1 Trincas Estáticas

Nas análises realizadas nesta seção, as trincas permaneceram estáticas, sendo os efeitos decorrentes da propagação avaliados em etapas posteriores. As Tabelas (4.23) e (4.24) apresentam o número de iterações realizadas e o tempo de processamento dos modelos resolvidos de forma sequencial, considerando as abordagens monolítica e *staggered*. Os tempos foram expressos de maneira relativa, tendo como referência o tempo de execução do modelo com duas mesoescalas utilizando a abordagem monolítica.

Tabela 4.23: Chapa com dez mesoescalas: Número de iterações (N. Iter.) e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas estáticas.

Número de Mesoescalas	N. Iter.	Tempo		
		Trecho Sequencial	Trecho Paralelizável	Total
2	15	0,394	0,606	1,000
6	12	0,283	1,413	1,696
10	11	0,295	2,131	2,426

Tabela 4.24: Chapa com dez mesoescalas: Número de iterações (N. Iter.) e dispêndio de tempo inerentes ao processo de solução sequencial ($n_p = 1$) via abordagem *staggered*. Trincas estáticas.

Número de Mesoescalas	N. Iter. GLI	N. Iter. <i>staggered</i>	Tempo		
			Trecho Sequencial	Trecho Paralelizável	Total
2	33	4	0,864	0,355	1,219
6	31	4	0,678	0,975	1,652
10	31	4	0,825	1,552	2,377

Considerando a abordagem monolítica, observa-se que o número de iterações necessárias para a convergência diminui a medida que são adicionados mais domínios locais. Especificamente, o número de iterações reduz-se de 15, no modelo com duas mesoescalas, para 11, no modelo completo. A partir dos dados obtidos pela abordagem *staggered*, é possível analisar este comportamento de forma detalhada, segregando as iterações relacionadas ao acoplamento GLI e ao enriquecimento global-local do MEF^{gl}. Dessa maneira, verifica-se que a redução no número de iterações ocorre exclusivamente nas iterações GLI, enquanto o número total de iterações na abordagem *staggered* permanece constante.

Esse comportamento decorre da aplicação da relaxação dinâmica, uma vez que o fator de relaxação é determinado a partir do vetor de incremento de deslocamentos calculados na interface Γ_I entre as mesoescalas e o domínio global complementar, o qual varia conforme o caso. Em análises com múltiplas mesoescalas, os vetores utilizados na Equação (2.9) são construídos a partir da contribuição de cada subdomínio e, portanto, são modificados em função das interfaces consideradas. Quando se adota a relaxação estática, o mesmo número de iterações é requerido, independentemente do número de mesoescalas. Para o

valor de $\omega = 1,0$, um total de quinze iterações foi executado na abordagem monolítica, evidenciando a eficácia da relaxação dinâmica.

A Figura (4.26) apresenta os tempos de processamento associados a cada análise, em função do número de mesoescalas e da quantidade de processadores utilizados. Para o modelo com duas mesoescalas, observa-se que a abordagem monolítica requer menor tempo de processamento em todas as simulações realizadas. A implementação de técnicas de paralelismo resultou em uma redução de aproximadamente 25,0% no tempo de análise, sendo este benefício independente do número de processadores disponíveis, uma vez que o modelo contempla apenas duas mesoescalas para paralelização. Por outro lado, na abordagem *staggered*, a redução do tempo de processamento devido ao paralelismo foi de aproximadamente 11,0%, indicando um ganho inferior em comparação com a abordagem monolítica. Ressalta-se, que no caso do modelo com duas mesoescalas, o tempo de solução utilizando a abordagem *staggered* paralelizada é 8,3% superior ao tempo observado na abordagem monolítica com processamento sequencial.

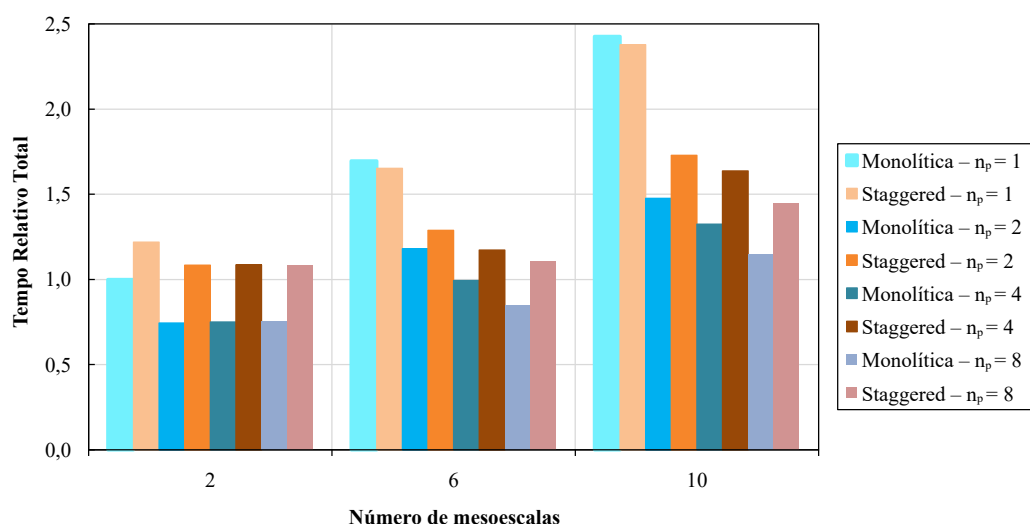


Figura 4.26: Chapa com dez mesoescalas: Representação gráfica do dispêndio de tempo em função do número de mesoescalas e processadores. Trincas estáticas.

Na simulação com apenas um processador, o aumento do número de domínios locais contribui para equilibrar a demanda computacional entre as diferentes abordagens de solução, reduzindo a discrepância no tempo gasto entre elas. Para os modelos com seis e dez mesoescalas, a abordagem *staggered* demonstra maior eficiência, alcançando a solução em menor tempo. Esses resultados podem ser atribuídos a dois fatores principais. O primeiro refere-se a manutenção da dimensão do problema global (e, conseqüentemente, do esforço computacional associado à sua resolução), uma vez que, na abordagem *staggered*, o processamento dessa escala do problema é priorizado em relação a escala local. O segundo fator está relacionado ao aumento do custo das iterações global-local do MFG^{gl} não ser linear, penalizando os índices de desempenho da abordagem monolítica.

Para os modelos com seis e dez mesoescalas, a disponibilização de um maior número de processadores para a análise faz com que a abordagem monolítica volte a apresentar o melhor desempenho em todas as simulações realizadas. Conforme ilustrado na Figura (4.27), a fração temporal de solução do código que pode ser paralelizada é significativamente maior na abordagem monolítica em comparação à *staggered*, o que resulta em uma maior eficiência no uso de processadores adicionais e em uma redução mais acentuada no tempo de análise. As curvas correspondentes a essa fração do tempo revelam uma relação praticamente linear entre essa variável e o número de subdomínios considerados, comportamento que reflete a similaridade do impacto das mesoescalas na solução iterativa do problema. O aumento no número de processadores não apenas desloca esses valores para tempos menores, como também reduz progressivamente a diferença entre as curvas associadas às duas abordagens de solução.

Considerando que, para um dado número de mesoescalas, a parcela de tempo sequencial da solução permanece inalterada e assume menores valores quando se utiliza o algoritmo monolítico, essa abordagem apresenta valores superiores de *SpeedUp* e eficiência computacional em todas as análises realizadas. Os resultados correspondentes estão apresentados nas Tabelas (4.25) e (4.26) e são representados graficamente na Figura (4.28). Resultados semelhantes foram observados no caso do pórtico analisado na Seção (4.2.1).

Observa-se que, para um dado número de processadores, o aumento no número de mesoescalas resulta consistentemente em um incremento do *SpeedUp*, evidenciando que a eficiência do paralelismo cresce à medida que o tamanho do problema aumenta. A taxa desse crescimento é determinada pela relação entre o número de processadores disponíveis, o número de mesoescalas e o grau de complexidade destas. Além da redução de situações de corrida, conforme discutido na Seção (4.2.1), um fator que contribui para os elevados índices de eficiência no uso de dois processadores é o fato de o número de mesoescalas em todos os modelos ser divisível por dois. Dessa forma, a carga de trabalho é distribuída uniformemente entre os processadores, minimizando a taxa de inatividade, diferentemente do observado na Seção (4.2.1).

Ao contrário da eficiência (E), que pode ser reduzida com o aumento do número de processadores, observa-se que o *SpeedUp* apresenta uma tendência de crescimento até que o número de mesoescalas se iguale ao número de processadores. Nesse ponto, o valor desse parâmetro tende a estabilizar, indicando uma saturação no ganho de desempenho. Neste exemplo, este comportamento é visto para o modelo com duas mesoescalas, já que os parâmetros de desempenho não sofrem variações relevantes em função do número de processadores.

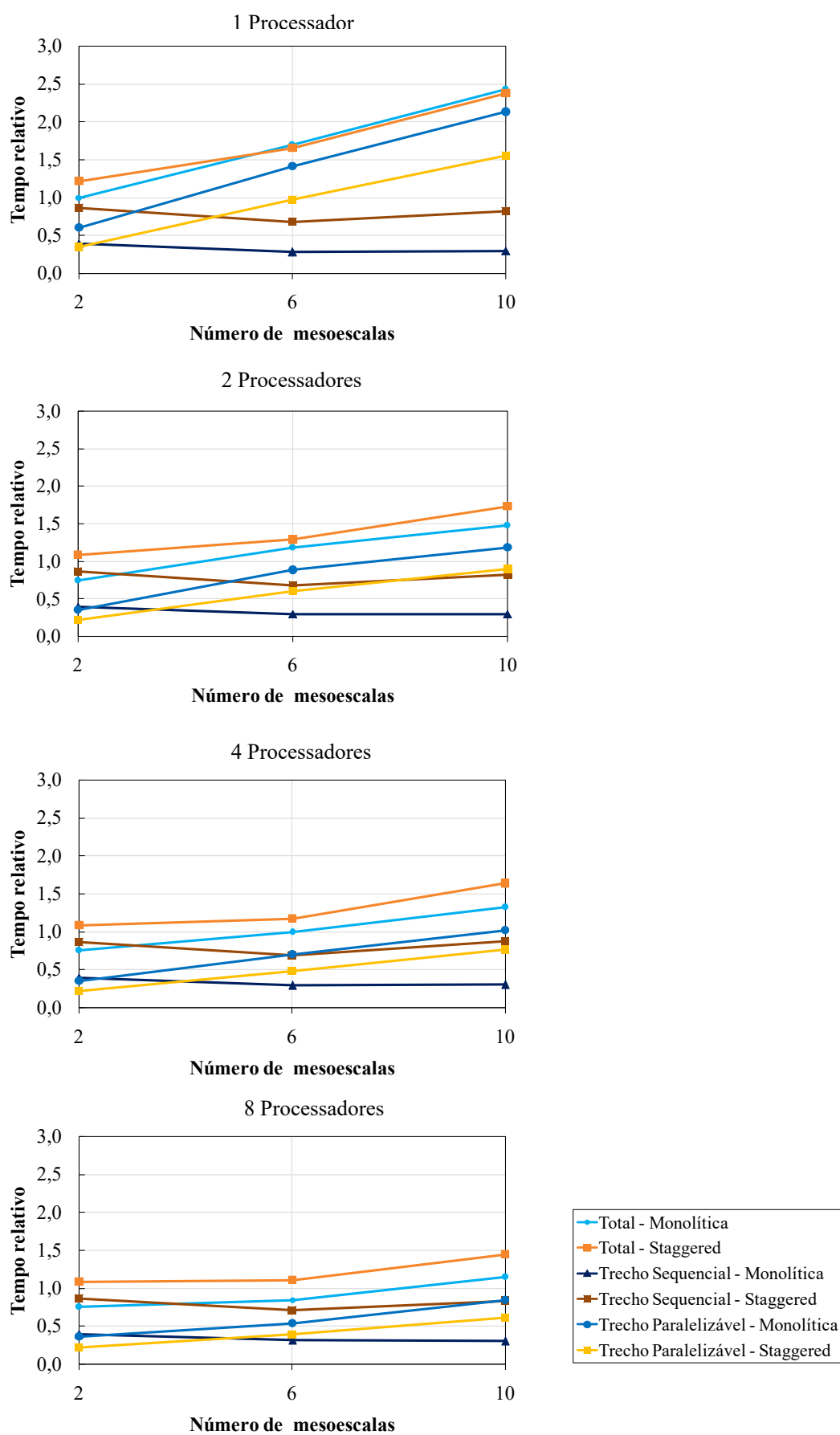


Figura 4.27: Chapa com dez mesoescalas: Parcelas de tempo referentes aos trechos sequenciais e paralelizáveis dos algoritmos do MEFG^{gl}-GLI. Trincas estáticas.

Tabela 4.25: Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Tempo Relativo			Medidas de desempenho		
		Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	2	0,393	0,350	0,743	1,345	0,673	1,435
	6	0,297	0,883	1,180	1,437	0,718	1,714
	10	0,295	1,181	1,476	1,643	0,822	1,783
4	2	0,395	0,355	0,751	1,332	0,333	1,833
	6	0,291	0,702	0,994	1,707	0,427	2,665
	10	0,304	1,021	1,325	1,831	0,458	2,932
8	2	0,395	0,363	0,757	1,320	0,165	2,129
	6	0,313	0,535	0,848	1,999	0,250	3,690
	10	0,302	0,847	1,149	2,110	0,264	4,323

Tabela 4.26: Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem *staggered* considerando o tempo total da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Tempo Relativo			Medidas de desempenho		
		Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	2	0,864	0,219	1,083	1,126	0,563	1,170
	6	0,682	0,606	1,287	1,283	0,642	1,418
	10	0,826	0,903	1,728	1,375	0,688	1,485
4	2	0,862	0,222	1,085	1,124	0,281	1,279
	6	0,690	0,483	1,173	1,409	0,352	1,793
	10	0,876	0,762	1,638	1,452	0,363	1,960
8	2	0,865	0,219	1,083	1,126	0,141	1,342
	6	0,713	0,396	1,109	1,490	0,186	2,067
	10	0,838	0,612	1,449	1,640	0,205	2,333

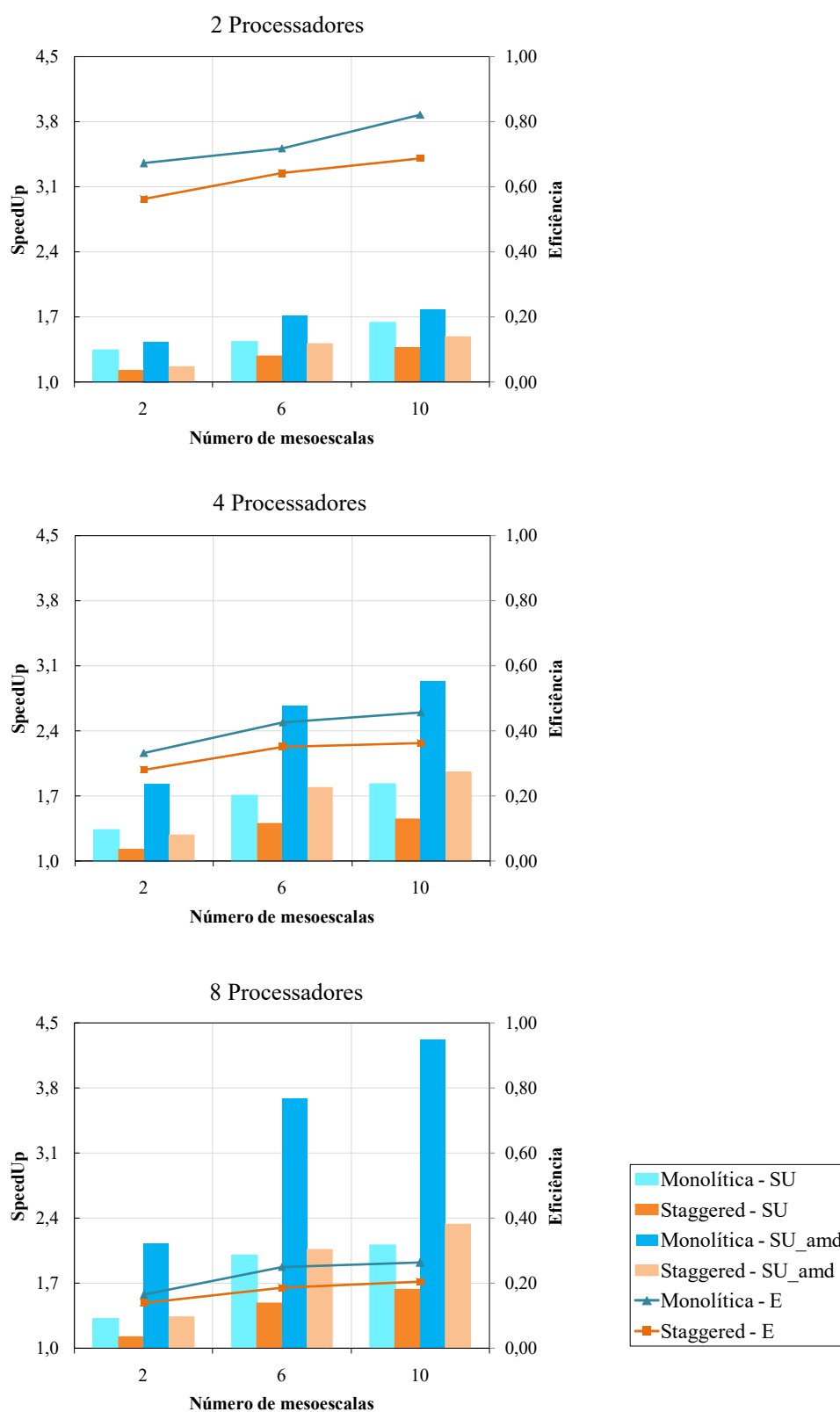


Figura 4.28: Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise. Trincas estáticas.

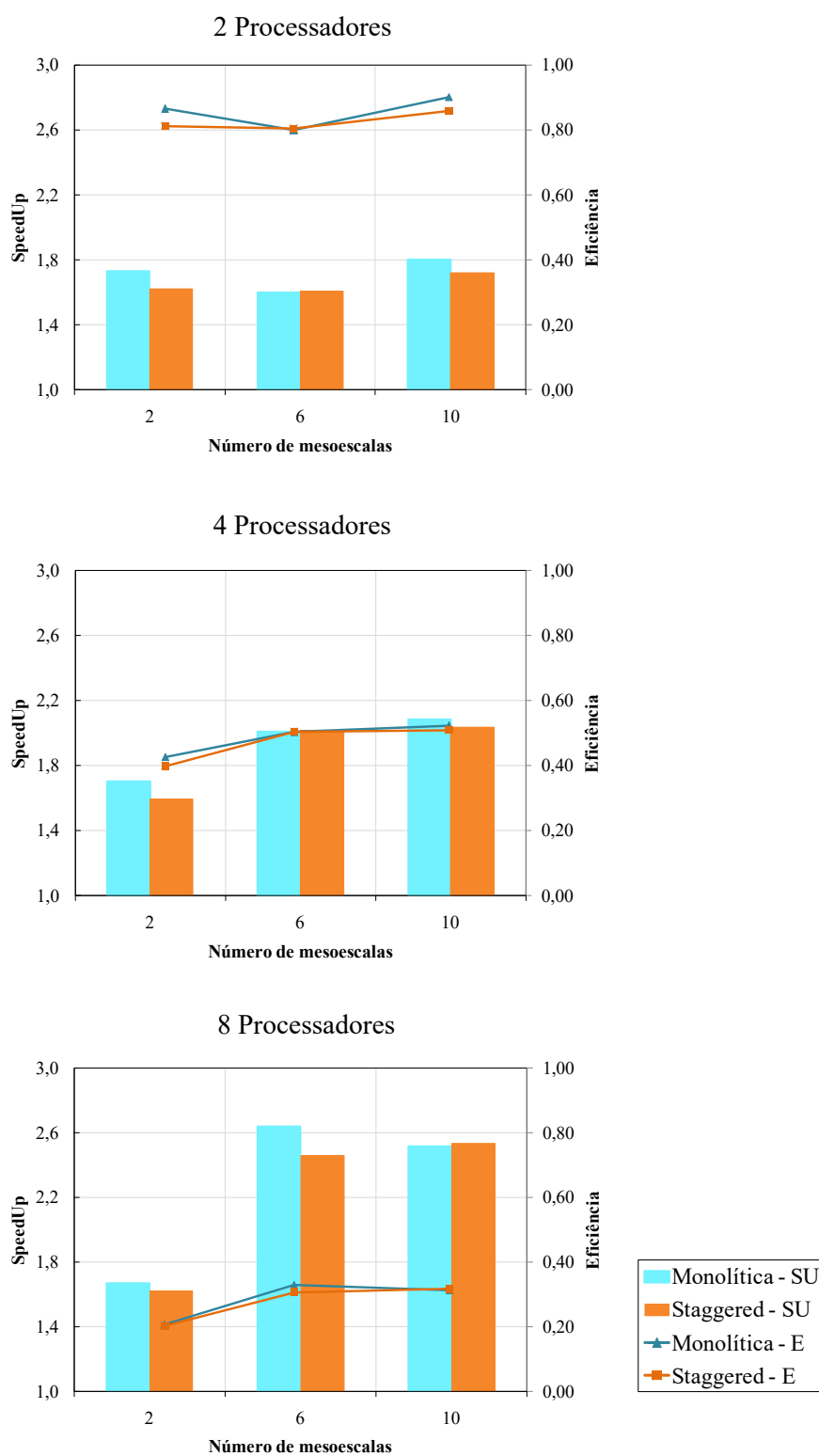


Figura 4.29: Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.

A Figura (4.29) e a Tabela (4.27) apresentam as métricas de desempenho considerando exclusivamente o tempo associado ao trecho paralelizável da análise. Observa-se que os parâmetros avaliados para as abordagens monolítica e *staggered* tornam-se significativamente próximos, indicando que, sob a perspectiva do processamento das mesoescalas, a taxa de ganho proporcionada pela paralelização é bastante semelhante entre as duas abordagens. Pequenas variações observadas são atribuídas a divisão do processo de solução de cada mesoescala em duas etapas na abordagem *staggered*, o que impacta o gerenciamento do *pool of threads*.

Tabela 4.27: Chapa com dez mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas estáticas.

Número de Processadores	Número de Mesoescalas	Abordagem Monolítica		Abordagem <i>staggered</i>	
		SU	E	SU	E
2	2	1,731	0,866	1,624	0,812
	6	1,600	0,800	1,609	0,804
	10	1,804	0,902	1,720	0,860
4	2	1,707	0,427	1,596	0,399
	6	2,012	0,503	2,018	0,505
	10	2,087	0,522	2,038	0,510
8	2	1,672	0,209	1,624	0,203
	6	2,640	0,330	2,460	0,308
	10	2,516	0,315	2,537	0,317

4.2.2.2 Trincas em propagação

Nesta seção, avalia-se o desempenho computacional dos três modelos da chapa utilizados na Seção (4.2.2.1), considerando trincas propagando-se. Inicialmente, examina-se a manutenção da premissa de equivalência do impacto sobre a solução das mesoescalas ao longo dos passos de propagação, e, conseqüentemente, a mesma penalização de rigidez atribuída no acoplamento GLI. A Tabela (4.28) apresenta os valores de K_I obtidos ao longo dos cinco passos de propagação das dez trincas. A variação máxima observada nos valores no quinto passo de propagação foi de 0,63%.

Em todos os casos, os valores de K_{II} foram substancialmente baixos, sendo numericamente considerados nulos, o que caracteriza a propagação em modo I puro. O número de iterações necessárias para a convergência permaneceu constante entre as mesoescalas, e o modelo exibiu comportamento simétrico. Assim, conclui-se que a propagação não gerou diferenciações significativas no comportamento entre as mesoescalas.

Tabela 4.28: Fator de Intensidade de Tensão extraído ao longo dos cinco passos de propagação.

Número de Mesoescalas	Abordagem	Mesoescalas	K_I				
			Passo 1	Passo 2	Passo 3	Passo 4	Passo 5
2	Monolítica	3 e 8	2284,297	2503,751	2714,806	2938,619	3135,988
	<i>Staggered</i>	3 e 8	2288,966	2507,350	2717,244	2939,450	3134,774
6	Monolítica	2 e 7	2283,112	2500,867	2709,872	2930,531	3125,102
		3 e 8	2283,440	2501,478	2710,742	2931,595	3126,584
		4 e 9	2283,077	2500,942	2709,955	2930,479	3125,260
		2 e 7	2285,406	2502,293	2710,466	2930,098	3122,771
	<i>Staggered</i>	3 e 8	2285,748	2502,909	2711,325	2931,157	3124,220
		4 e 9	2285,384	2502,401	2710,562	2930,055	3122,811
		1 e 6	2280,324	2496,918	2704,750	2924,828	3117,955
		2 e 7	2280,657	2497,671	2705,756	2926,024	3119,714
10	Monolítica	3 e 8	2280,658	2497,747	2705,838	2926,048	3119,776
		4 e 9	2280,652	2497,780	2705,880	2926,062	3119,815
		5 e 10	2280,295	2497,240	2705,088	2924,948	3118,366
		1 e 6	2283,136	2498,993	2706,067	2924,937	3116,138
		2 e 7	2283,465	2499,718	2707,069	2926,065	3117,711
	<i>Staggered</i>	3 e 8	2283,460	2499,777	2707,124	2926,040	3117,673
		4 e 9	2283,448	2499,799	2707,146	2926,023	3117,656
		5 e 10	2283,091	2499,260	2706,356	2924,904	3116,191

Os tempos correspondentes as abordagens monolítica e *staggered* nas análises realizadas com processamento sequencial estão apresentados nas Tabelas (4.29) e (4.30), respectivamente. Os tempos foram expressos de forma relativa, utilizando como referência o tempo de execução do modelo com propagação processado de maneira sequencial pela abordagem monolítica e com duas mesoescalas.

Tabela 4.29: Chapa com dez mesoescalas: Dispêndio de tempo inerente ao processo de solução sequencial ($n_p = 1$) via abordagem monolítica. Trincas em propagação.

Número de Mesoescalas	Tempo		
	Trecho Sequencial	Trecho Paralelizável	Total
2	0,305	0,695	1,000
6	0,201	1,285	1,485
10	0,220	1,984	2,205

Tabela 4.30: Chapa com dez mesoescalas: Dispêndio de tempo inerente ao processo de solução sequencial ($n_p = 1$) via abordagem *staggered*. Trincas em propagação.

Número de Mesoescalas	Tempo		
	Trecho Sequencial	Trecho Paralelizável	Total
2	0,370	0,281	0,651
6	0,310	0,820	1,130
10	0,395	1,289	1,684

A Figura (4.30) apresenta, de forma gráfica, os tempos de processamento obtidos nas análises desta seção. Nesse cenário, a abordagem *staggered* apresentou menor tempo de execução em todos os casos avaliados. A propagação das trincas gerou um aumento significativo no número total de iterações global-local do MEF G^{gl} , passando de 110 para 550 na abordagem monolítica e de 40 para 180 na *staggered*. Para as trincas estáticas, a abordagem *staggered* executou 63,6% menos iterações global-local em relação a monolítica, enquanto para as trincas em propagação, essa redução foi de aproximadamente 67,3%. Apesar da similaridade entre as porcentagens, o maior custo computacional das iterações global-local impacta de forma mais significativa o tempo de processamento do trecho paralelizável do código, conforme evidenciado nas Tabelas (4.31) e (4.32).

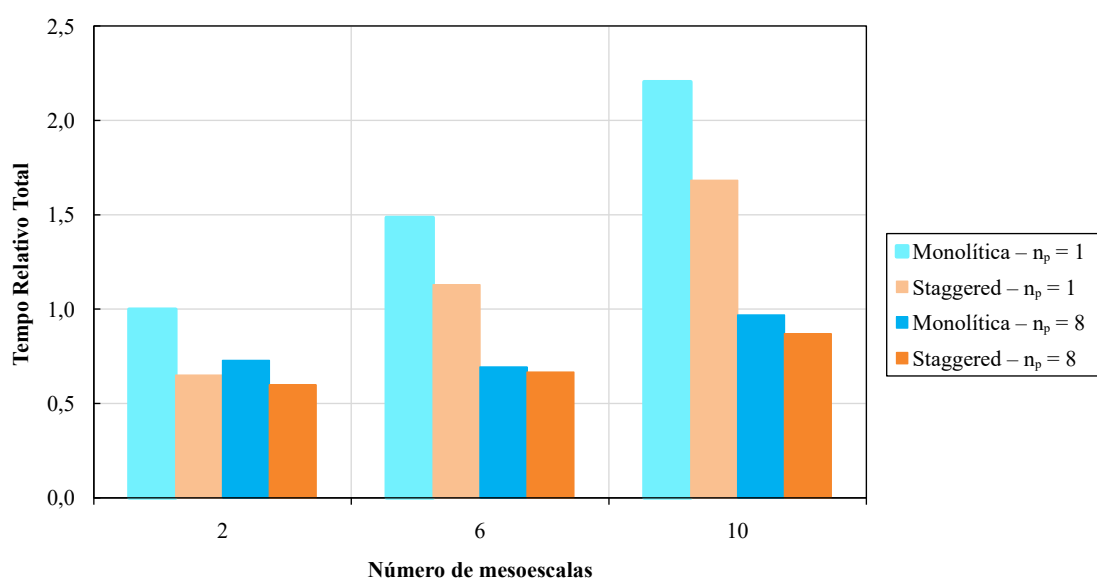


Figura 4.30: Chapa com dez mesoescalas: Representação gráfica do dispêndio de tempo total em função do número de mesoescalas e processadores considerando cinco passos de propagação.

A Figura (4.31) apresenta os parâmetros de desempenho para o modelo analisado nesta seção. Conforme era esperado, os valores de *SpeedUp* obtidos para o algoritmo monolítico são superiores, refletindo a maior eficiência na paralelização dessa abordagem de solução. Assim, a diferença de tempo despendido entre as abordagens de solução é reduzida com o aumento do número de mesoescalas, como pode ser observado comparando-se os tempos para $n_p = 1$ e $n_p = 8$ na Figura (4.30).

Portanto, por meio das técnicas de paralelismo, a abordagem monolítica atinge a convergência com um tempo mais próximo ao da abordagem *staggered*, mas sem alcançá-lo. Dessa forma, conclui-se que, além do aumento do número de mesoescalas, suas características e o número de iterações global-local necessárias para a convergência são determinantes para a obtenção do menor tempo de processamento entre as abordagens de solução avaliadas.

Tabela 4.31: Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem monolítica considerando o tempo total da análise. Trincas em propagação.

Número de Mesoescalas	Tempo			Medidas de desempenho		
	Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	0,321	0,408	0,729	1,372	0,172	2,551
6	0,227	0,466	0,694	2,142	0,268	4,110
10	0,211	0,759	0,970	2,273	0,284	4,708

Tabela 4.32: Chapa com dez mesoescalas: Medidas de desempenho inerentes a abordagem *staggered* considerando o tempo total da análise. Trincas em propagação.

Número de Mesoescalas	Tempo			Medidas de desempenho		
	Trecho Sequencial	Trecho Paralelizável	Total	SU	E	SU_amd
2	0,420	0,179	0,600	1,085	0,136	1,606
6	0,388	0,279	0,667	1,693	0,212	2,739
10	0,385	0,486	0,870	1,935	0,242	3,030

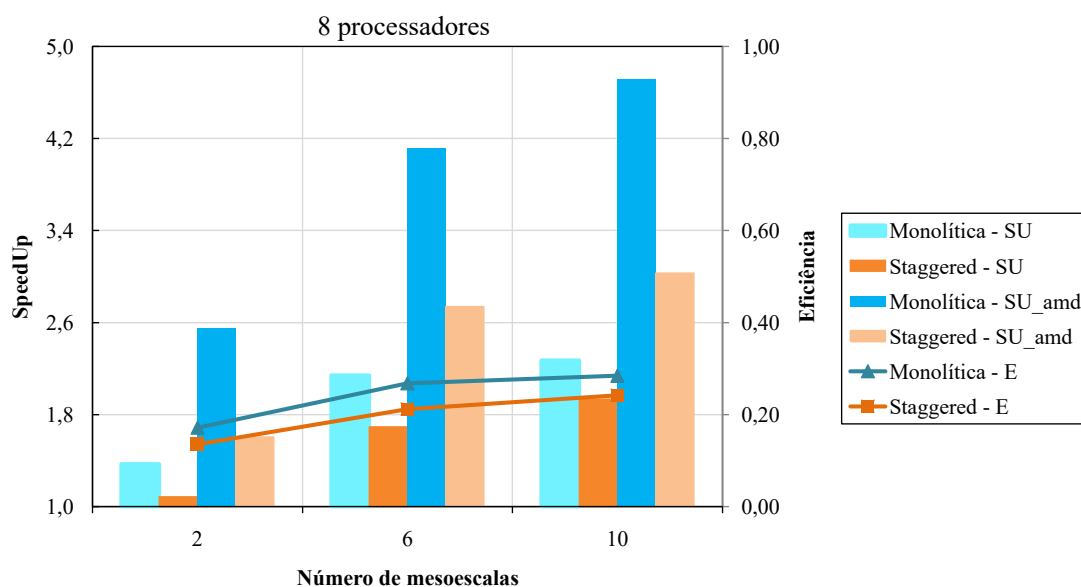


Figura 4.31: Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando o tempo total da análise. Trincas em propagação.

De maneira análoga ao que foi apresentado nas seções anteriores, a Tabela (4.33) e a Figura (4.32) apresentam as métricas de desempenho, levando em consideração exclusivamente o tempo relativo ao trecho paralelizável da análise. Com exceção do modelo contendo duas mesoescalas, os parâmetros de desempenho obtidos pela abordagem *staggered* demonstram valores superiores aos da abordagem monolítica, reforçando a conclusão de que a redução do número de iterações global-local do MEF^{gl} executados na abordagem *staggered* resulta em uma maior eficiência relativa.

Tabela 4.33: Chapa com dez mesoescalas: Medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas em propagação.

Número de Mesoescalas	Abordagem Monolítica		Abordagem <i>staggered</i>	
	SU	E	SU	E
2	1,705	0,213	1,565	0,196
6	2,756	0,344	2,936	0,367
10	2,614	0,327	2,655	0,332

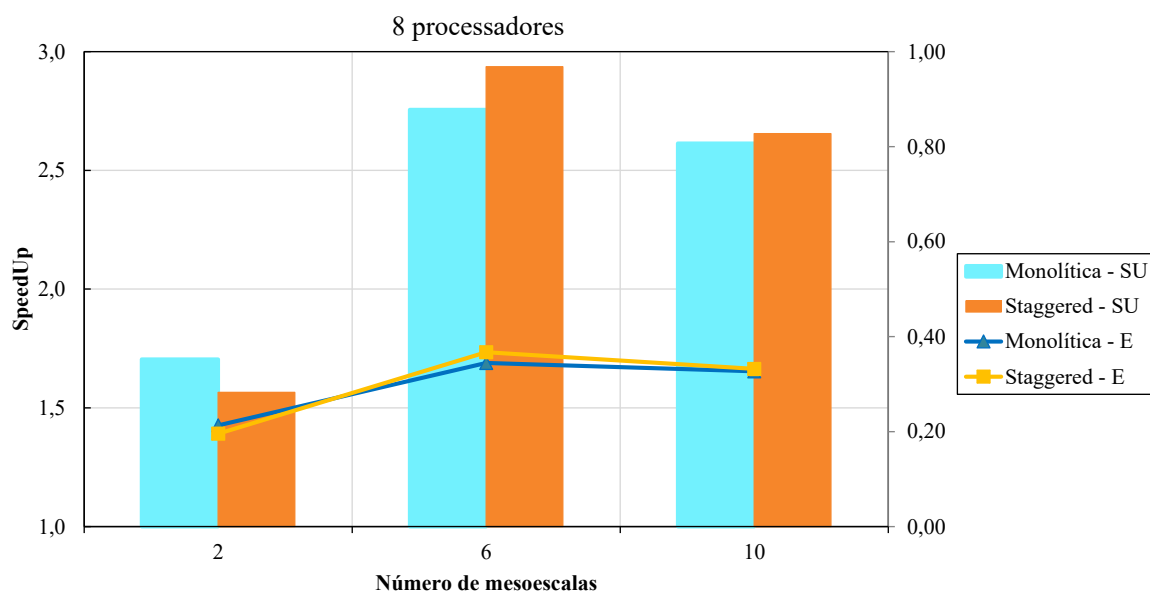


Figura 4.32: Chapa com dez mesoescalas: Representação gráfica das medidas de desempenho considerando apenas o tempo correspondente ao trecho paralelizável da análise. Trincas em propagação.

CAPÍTULO 5

CONSIDERAÇÕES FINAIS

5.1 Considerações deste Trabalho

Este trabalho se dedicou em expandir a implementação da estratégia MEFG^{gl}-GLI no ambiente computacional *INSANE*, viabilizando a análise de problemas com múltiplos domínios locais. A abordagem não intrusiva possibilita o acoplamento do *INSANE* ao *Abaqus*, resultando em uma ferramenta para análise estrutural que integra a acessibilidade e funcionalidades de um programa comercial com *solvers* numéricos avançados, como o MEFG^{gl}, disponíveis no sistema computacional acadêmico. O MEFG^{gl} viabiliza o uso de uma ampla gama de funções (definidas conforme o tipo de aplicação) para enriquecimento, aprimorando a solução de problemas que apresentam singularidades e descontinuidades.

As abordagens de solução monolítica e *staggered*, previamente implementadas no *INSANE*, foram adaptadas para abranger uma gama mais ampla de problemas, permitindo a avaliação de diversos fenômenos localizados. Neste estudo, o MEFG^{gl}-GLI foi aplicado a problemas da Mecânica da Fratura Linear Elástica envolvendo múltiplas trincas. As principais conclusões deste trabalho são apresentadas a seguir.

5.1.1 Validação da implementação

As simulações numéricas apresentadas na Seção (4.1) demonstraram a eficácia e precisão das estratégias propostas, evidenciando sua aplicação a problemas com características distintas. Os resultados obtidos para os parâmetros das trincas em modos de abertura puro e misto foram satisfatórios, com trajetórias de propagação consistentes e efeitos de simetria devidamente confirmados nos modelos analisados. Os valores calculados para os fatores de intensidade de tensão apresentaram pequenos erros, ao serem confrontados com as soluções de referência e equivalentes. Além disso, os resultados numéricos indicaram a preservação dos efeitos benéficos associados ao uso das técnicas de relaxação aplicadas no GLI, mesmo em análises envolvendo múltiplos domínios.

No que se refere ao comportamento global dos modelos, no exemplo analisado na Seção (4.1.1), os resultados obtidos para os componentes de deslocamento da aresta superior do pórtico na direção y coincidiram com aqueles verificados no modelo equivalente, solucio-

nado integralmente pelo MEFG^{gl} no *INSANE*. Ademais, constatou-se a compatibilidade entre os resultados obtidos pelas duas estratégias de solução propostas, corroborando a precisão dos métodos adotados.

As análises evidenciaram que a consideração de múltiplas mesoescalas não compromete a convergência do método, desde que a diferença de rigidez entre as escalas do problema seja adequadamente controlada. Com base nos exemplos apresentados na Seção (4.2), conclui-se que a interação mútua entre os diferentes modelos locais não constitui uma preocupação significativa. A partir desses resultados, foram propostas versões aprimoradas dos algoritmos do MEFG^{gl}-GLI, nas quais o processamento de cada mesoescala é interrompido ao atender aos critérios de convergência estabelecidos. Os resultados obtidos com essa estratégia apresentaram variações limitadas à terceira casa decimal, atestando a precisão dos algoritmos aprimorados. No que se refere ao desempenho computacional, observou-se uma redução no número total de iterações em 18,7% para a abordagem monolítica e 18,2% para a abordagem *staggered*, evidenciando ganhos significativos em eficiência computacional.

Ressalta-se que o objetivo deste trabalho não é avaliar a interferência entre trincas gerada em situações nas quais a distância entre as fissuras é relativamente pequena. Nesses casos, seria recomendável englobar ambas as descontinuidades em uma mesma mesoescala, referida como problema mestre, conforme descrito por Kim *et al.* (2011). Essa abordagem permitiria considerar de forma mais precisa a influência mútua entre as trincas. Detalhamentos adicionais sobre a interação entre trincas e a paralelização dos domínios locais do MEFG^{gl} podem ser encontrados em Kim *et al.* (2008) e Kim *et al.* (2011). A análise desse comportamento é sugerida como uma continuidade para esta pesquisa, conforme discutido na Seção (5.2).

5.1.2 Desempenho Computacional

Através da implementação de estratégias de paralelismo, foi possível distribuir os múltiplos domínios locais entre vários processadores, otimizando o uso dos recursos de *hardware* disponíveis. O emprego do paralelismo demonstra maior eficiência à medida que se incrementa o número de graus de liberdade, subdomínios e configurações de análise do modelo, sendo estes últimos correspondentes aos passos de propagação das trincas. Esses cenários são os que apresentam a maior redução relativa no tempo de análise.

O balanceamento de carga entre os processadores constitui um aspecto fundamental para a metodologia adotada nesta pesquisa. As estratégias de ordenação e distribuição de tarefas asseguraram a utilização mais eficiente dos recursos computacionais, minimizando a ociosidade dos processadores. Nesse contexto, o gerenciamento é realizado por meio do *pool of threads*, que administra um conjunto pré-alocado de *threads* reutilizáveis. Assim, em vez de criar e destruir *threads* repetidamente, o *pool* atribui tarefas às *threads*

disponíveis, que, ao concluir sua execução, retornam ao *pool*. No início de cada etapa de processamento, as *threads* que excedem o número de mesoescalas são descartadas, liberando os recursos computacionais para o sistema.

Em análises realizadas por processamento sequencial, observa-se que, na maioria dos casos, a abordagem monolítica apresenta um menor tempo de processamento quando comparada à abordagem *staggered*. Contudo, à medida que o problema analisado cresce, a abordagem *staggered* passa a apresentar o menor tempo de análise. Esse comportamento resulta do maior custo computacional das iterações global-local do MCFG^{gl}, cujo total necessário é inferior na abordagem *staggered*.

A adoção de técnicas de paralelismo confere maiores benefícios à abordagem monolítica, uma vez que a parcela de tempo paralelizável neste caso é superior à observada para o algoritmo *staggered*. Consequentemente, essa abordagem de solução apresenta os maiores valores de *speedUp*, que quantifica o ganho de desempenho ao utilizar múltiplos processadores, comparando o tempo de execução paralelo com o sequencial. A eficiência *E* segue um comportamento similar. Assim, para a grande maioria das simulações realizadas neste estudo, o paralelismo fez com que a abordagem monolítica fosse a solução mais eficiente em termos de tempo de processamento. No entanto, o aumento progressivo da escala do problema reestabelece a tendência observada na análise sequencial, na qual a abordagem *staggered* apresenta menores tempos de execução. Deste modo, verifica-se uma situação de *trade-off*, marcada pelas alternância das vantagens e desvantagens entre as abordagens monolítica e *staggered*, conforme evidenciado para a chapa com dez mesoescalas considerando as trincas estáticas (Seção (4.2.2.1)) ou em estado de propagação (Seção (4.2.2.2)).

As análises que utilizaram dois processadores demonstraram os maiores índices de eficiência, uma vez que as condições de concorrência entre as *threads* são substancialmente reduzidas, assim como o tempo de ociosidade. Este comportamento decorre de todos os casos analisados, possuem mais de dois problemas locais, permitindo uma distribuição mais equilibrada da carga de trabalho entre os processadores.

5.1.3 Relevância da pesquisa

A estratégia de acoplamento não intrusivo desenvolvida neste trabalho evidencia como ferramentas computacionais desenvolvidas em ambientes acadêmicos podem ser aplicadas a problemas práticos, criando novas oportunidades para colaboração interdisciplinar. A utilização de um programa comercial em uma das esferas de solução estimula a indústria a adotar métodos em desenvolvimento em centros de pesquisa, tornando-os acessíveis a engenheiros que desejam realizar análises mais detalhadas, sem a necessidade de modificar seus solucionadores já existentes.

Dessa forma, o presente trabalho demonstra o potencial do MCFG^{gl}-GLI para aplicações

abrangentes, evidenciando sua viabilidade em problemas com múltiplos domínios locais. A adoção de técnicas de paralelização e os aprimoramentos introduzidos no algoritmo contribuem para a redução do custo computacional, tornando a estratégia mais atrativa. Assim, as contribuições deste estudo representam um avanço significativo na linha de pesquisa relacionada ao MEF gl -GLI, especialmente no que tange à sua aplicação prática. Conclui-se ainda que, com base nos experimentos realizados, pode-se avaliar a influência dos múltiplos domínios e outros parâmetros do método na condição de convergência.

5.2 Sugestões para Trabalhos Futuros

Com base neste trabalho, podem ser sugeridos os seguintes tópicos como propostas para trabalhos futuros:

1. expandir a implementação para a consideração de múltiplos domínios locais e trincas em uma mesma mesoescala, viabilizando a análise da interferência mútua entre as fissuras. Resolver os domínios locais por meio de computação paralela;
2. expandir a implementação para análise de trincas, de modo a contemplar:
 - critérios de propagação com base na tenacidade à fratura do material;
 - análise com carregamentos cíclicos e avaliação da propagação de trincas por fadiga;
3. adaptar a implementação para consideração de fenômenos locais diversos, como furos, soldas, enrijecedores, etc.;
4. adaptar a implementação para acoplamento de diferentes tipos de elementos finitos, passíveis de serem empregados em cada escala do problema;
5. aplicar as estratégias de acoplamento em problemas com comportamento não linear;
6. avaliar os impactos na convergência devido a adoção de coeficientes de relaxação, ω , diferentes para cada mesoescala;
7. implementação de um algoritmo de geração automática de modelos da mesoescala, de modo similar ao desenvolvido por Fonseca (2019) para problemas locais do MEF gl .

BIBLIOGRAFIA

- ABNT (2023), ‘NBR 6123:2023 - forças devido ao vento em edificações’. Disponível em: <https://www.abntcatalogo.com.br/>.
- Allix, O. e Gosselet, P. (2020), ‘Non intrusive global/local coupling techniques in solid mechanics: An introduction to different coupling strategies and acceleration techniques’, *Modeling in engineering using innovative numerical methods for solids and fluids* pp. 203–220.
- Alves, P. D. (2012), Estratégia global-local aplicada ao método dos elementos finitos generalizados, Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Amdahl, G. M. (1967), Validity of the single processor approach to achieving large scale computing capabilities, *in* ‘Proceedings of the April 18-20, 1967, Spring Joint Computer Conference’, AFIPS ’67 (Spring), Association for Computing Machinery, New York, NY, USA, p. 483–485.
- Azevedo, G. S. (2019), Implementação paralela para análises estáticas lineares pelo método dos elementos finitos, Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Babuska, I., Caloz, G. e Osborn, J. E. (1994), ‘Special finite element methods for a class of second order elliptic problems with rough coefficients’, *SIAM Journal on Numerical Analysis* vol. 31(4), 945–981.
- Babuška, I. e Melenk, J. M. (1996), ‘The partition of unity method’, *International Journal for Numerical Methods in Engineering* vol. 40, 727–758.
- Barros, F. B. (2002), Métodos Sem Malha e Métodos dos Elementos Finitos Generalizados em Análise Não-Linear de Estruturas, PhD thesis, EESC - USP, São Carlos, SP, Brasil.
- Belytschko, T. e Black, T. (1999), ‘Elastic crack growth in finite elements with minimal remeshing’, *International journal for numerical methods in engineering* vol. 45(5), 601–620.

- Birner, M. e Schweitzer, M. A. (2019), *Global-Local Enrichments in PUMA*, Springer International Publishing, Cham, pp. 167–183.
- Bueno, A. C. P., Barros, F. B., Filho, N. A. S. *et al.* (2023), ‘Expansion of a non-intrusive implementation of the generalized finite element method - global-local’, *Proceedings of COBEM(International Congress of Mechanical Engineering)* .
- Chevreuril, M., Nouy, A. e Safatly, E. (2013), ‘A multiscale method with patch for the solution of stochastic partial differential equations with localized uncertainties’, *Computer methods in applied mechanics and engineering* vol. 255, 255–274.
- Culler, D., Singh, J. P. e Gupta, A. (1998), *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Dowling, N. E. (2013), *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue*, 4th edn, Pearson, Boston, MA.
- Duarte, C. A. e Kim, D. J. (2008), ‘Analysis and applications of a generalized finite element method with global-local enrichment functions’, *Computer Methods in Applied Mechanics and Engineering* vol. 197(6–8), 487–504.
- Duarte, C. A. e Oden, J. T. (1995), ‘Hp-clouds -A Meshless Method to solve Boundary-value Problems’.
- Duval, M., Passieux, J. C., Salaun, M. e Guinard, S. (2014), ‘Non-intrusive coupling: Recent advances and scalable nonlinear domain decomposition’, *Archives of Computational Methods in Engineering* vol. 23(1), 17–38.
- Eager, D., Zahorjan, J. e Lazowska, E. (1989), ‘Speedup versus efficiency in parallel systems’, *IEEE Transactions on Computers* vol. 38(3), 408–423.
- Evangelista, F., Alves, G., Moreira, F. e Paiva, G. (2020), ‘A global–local strategy with the generalized finite element framework for continuum damage models’, *Computer Methods in Applied Mechanics and Engineering* vol. 363, 112888.
- Evangelista, F., Roesler, J. e Duarte, C. A. (2013), ‘Two-scale approach to predict multi-site cracking potential in 3-d structures using the generalized finite element method’, *International Journal of Solids and Structures* vol. 50, 1991–2002.
- Filmore, T. B. e Duarte, C. A. (2018), ‘A hierarchical non intrusive algorithm for the generalized finite element method’, *Advanced Modeling and Simulation in Engineering Sciences* vol. 5(1), 2.

- Fonseca, G., Barros, F., de Oliveira, T., Monteiro, H., Novelli, L. e Pitangueira, R. (2020), ‘2-d crack propagation analysis using stable generalized finite element method with global-local enrichments’, *Engineering Analysis with Boundary Elements* vol. 118, 70–83.
- Fonseca, G. M. (2019), Propagação de trincas em meios elásticos lineares via método dos elementos finitos generalizados com estratégia global local automatizada, Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Fonseca, G. M. (2023), Estimadores de erro a posteriori e estratégias adaptativas para o Método dos Elementos Finitos Generalizados com Enriquecimento Global-Local aplicado a problemas da Mecânica da Fratura, PhD thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Fries, T. P. e Belytschko, T. (2010), ‘The extended/generalized finite element method: an overview of the method and its applications’, *International journal for numerical methods in engineering* vol. 84(3), 253–304.
- Gupta, V., Kim, D. J. e Duarte, C. A. (2012), ‘Analysis and improvements of globallocal enrichments for the generalized finite element method’, *Computer Methods in Applied Mechanics and Engineering* vol. 245, 47–62.
- Gupta, V., Pereira, J. P., Kim, D. J., Duarte, C. A. e Eason, T. (2012), ‘Analysis of three-dimensional fracture mechanics problems: A non-intrusive approach using a generalized finite element method’, *Engineering Fracture Mechanics* vol. 90, 41–64.
- Irons, B. M. e Tuck, R. C. (1969), ‘A version of the aitken accelerator for computer iteration’, *International Journal of Numerical Methods in Engineering* vol. 1, 275–277.
- Jython (2022), ‘Jython 2.7.3’. Setembro 10, 2022.
URL: www.jython.org
- Kim, D.-J., Duarte, C. A. e Pereira, J. (2008), ‘Analysis of interacting cracks using the generalized finite element method with global-local enrichment functions’, *Journal of Applied Mechanics* vol. 75.
- Kim, D.-J., Duarte, C. A. e Sobh, N. (2011), ‘Parallel simulations of three-dimensional cracks using the generalized finite element method’, *Computational Mechanics* vol. 47, 265–282.
- Kim, D. J., Pereira, J. P. e Duarte, C. A. (2010), ‘Analysis of three dimensional fracture mechanics problems: A two scale approach using coarse generalized fem meshes’, *International journal for numerical methods in engineering* vol. 81(3), 335–365.

- Kim, J. e Duarte, C. A. (2015), ‘A new generalized finite element method for two-scale simulations of propagating cohesive fractures in 3-d’, *International Journal for Numerical Methods in Engineering* vol. 104.
- Kim, J., Duarte, C. A. e Proenca, S. P. (2012), ‘A generalized finite element method with global-local enrichment functions for confined plasticity problems’, *Computational Mechanics* vol. 50.
- Li, H., Avecillas-Leon, J., Shauer, N. e Duarte, C. A. (2022), ‘A non-intrusive iterative generalized finite element method for multiscale coupling of 3-d solid and shell models’, *Computer Methods in Applied Mechanics and Engineering* vol. 402, 115408.
- Li, H., O’hara, P. e Duarte, C. A. (2021), ‘Non intrusive coupling of a 3 d generalized finite element method and abaqus for the multiscale analysis of localized defects and structural features’, *Finite Elements in Analysis and Design* vol. 193, 103554.
- Liu, Y. J., Sun, Q. e Fan, X. L. (2014), ‘A non-intrusive global/local algorithm with non-matching interface: Derivation and numerical validation’, *Computer methods in applied mechanics and engineering* vol. 277, 81–103.
- Malekan, M. (2017), Crack propagation modeling in plane structures using two-scale Generalized/Extended Finite Element Method, PhD thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Malekan, M. e Barros, F. (2018), ‘Numerical analysis of a main crack interactions with micro-defects/inhomogeneities using two-scale generalized/extended finite element method’, *Computational Mechanics* vol. 62, 783–801.
- Malekan, M., Barros, F. e Pitangueira, R. (2018), ‘Fracture analysis in plane structures with the two-scale g/xfem method’, *International Journal of Solids and Structures* vol. 155, 65–80.
- Marques, T., Fonseca, G., Lins, R. e Barros, F. (2025), ‘On the improvement of the local boundary conditions in gfemgl’, *Engineering Analysis with Boundary Elements* vol. 176, 106219.
- Bueno, A. C. P. (2024), Implementação não intrusiva do método dos elementos finitos generalizados global-local para simulação de problemas com trincas, Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Melenk, J. M. (1995), On generalized finite element methods, PhD thesis, University of Maryland, College Park, Maryland, EUA.

- Melenk, J. M. e Babuška, I. (1996), ‘The partition of unity finite element method: Basic theory and applications’, *Computer Methods in Applied Mechanics and Engineering* vol. 139(1-4), 289–314.
- Moës, N., Dolbow, J. e Belytschko, T. (1999), ‘A finite element method for crack growth without remeshing’, *International Journal for Numerical Methods in Engineering* vol. 46, 131–150.
- Monteiro, H. A. S., Novelli, L., Fonseca, G. M., Pitangueira, R. L. S. e Barros, F. B. (2020), ‘A new approach for physically nonlinear analysis of continuum damage mechanics problems using the generalized/extended finite element method with global-local enrichment’, *Engineering Analysis With Boundary Elements* vol. 113, 277–295.
- Noor, A. K. (1986), ‘Global local methodologies and their application to nonlinear analysis’, *Finite Elements in Analysis and Design* vol. 2(4), 333–346.
- Novelli, L., de Oliveira, T. S., da Silveira Monteiro, H. A., Fonseca, G. M., da Silva Pitangueira, R. L. e Barros, F. B. (2020), ‘Stable generalized/extended finite element method with global–local enrichment for material nonlinear analysis’, *Computer Methods in Applied Mechanics and Engineering* vol. 372, 113429.
- O’Hara, P., Duarte, C. A. e Eason, T. (2009), ‘Generalized finite element analysis of three-dimensional heat transfer problems exhibiting sharp thermal gradients’, *Computer Methods in Applied Mechanics and Engineering* vol. 198(21-26), 1857–1871.
- O’Hara, P., Hollkamp, J., Duarte, C. e Eason, T. (2016), ‘A two-scale generalized finite element method for fatigue crack propagation simulations utilizing a fixed, coarse hexahedral mesh’, *Computational Mechanics* vol. 57(1), 55–74.
- O’Hara, P., Duarte, C. A. e Eason, T. (2016), ‘A two-scale generalized finite element method for interaction and coalescence of multiple crack surfaces’, *Engineering Fracture Mechanics* vol. 163.
- Pereira, J. P., Kim, D. J. e Duarte, C. A. (2012), ‘A two-scale approach for the analysis of propagating three-dimensional fractures’, *Computational Mechanics* vol. 49(1), 99–121.
- Pinheiro, D. C. C. e Pitangueira, R. L. S. (2022), ‘Global local analysis with element free galerkin method’, *Engineering Analysis with Boundary Elements* vol. 136, 186–203.
- Pitangueira, R. L. S., Fonseca, F. T., Fuina, J. S., Camara, L., Ferreira, R. L., Moreira, R. N., Penna, S. S., Saliba, S. S. e Fonseca, M. T. (2008), ‘Insane - versão 2.0’, *XXVII Latin American Congress on Computational Methods in Engineering* pp. 1–20.

- Plews, J., Duarte, C. A. e Eason, T. (2012), ‘An improved nonintrusive globallocal approach for sharp thermal gradients in a standard fea platform’, *International Journal for Numerical Methods in Engineering* vol. 91(4), 426–449.
- Santos, K. F. e Barros, F. B. (2020), ‘Sgfem: a numerical investigation of the two-scale simulation of 3d elastic fracture’, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* vol. 42, 1806–3691.
- Silveira Filho, N. A. (2023), Implementação não intrusiva do método dos elementos finitos generalizados global-local, Master’s thesis, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Strouboulis, T., Babuška, I. e Copps, K. (2000), ‘The design and analysis of the Generalized Finite Element Method’, *Computer Methods in Applied Mechanics and Engineering* vol. 181(1-3), 43–69.
- Szabo, B. e Babuška, I. (1991), *Finite Element Analysis*, John Wiley & Sons, Inc.
- Whitcomb, J. D. (1991), ‘Iterative global local finite element analysis’, *Computers & Structures* vol. 40(4), 1027–1031.

Apêndices

APÊNDICE A

CÁLCULO DAS FORÇAS DE VENTO

Apresenta-se neste apêndice as etapas de cálculo para determinação das forças de vento aplicadas ao pórtico da Seção (4.2.1). Foram seguidas as orientações da NBR-6123 (ABNT (2023)). A Figura (A.1) ilustra as dimensões em planta do modelo considerado, sendo o pórtico avaliado neste trabalho locado no eixo central da estrutura, conforme destacado:

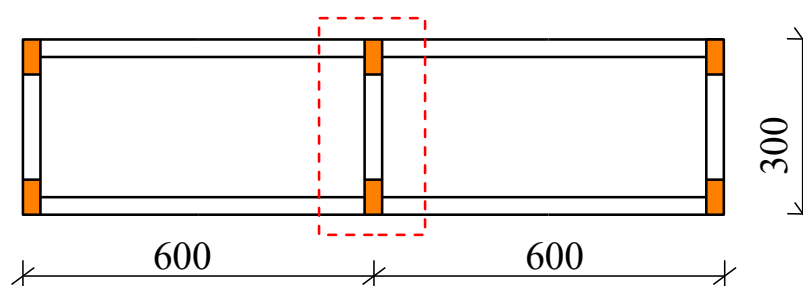


Figura A.1: Geometria em planta da estrutura analisada.

No processo de cálculo, foram considerados os seguintes parâmetros:

- edificação situada em Belo Horizonte: velocidade básica do vento - $V_0 = 31,0$;
- fator S_1 (topográfico): terreno plano ou fracamente acidentado - $S_1 = 1,0$;
- fator S_2 :

$$S_2 = b \times F_r \times (z/10)^p \quad (\text{A.1})$$

Sendo o Fator de Rajada, $F_r = 0,98$ e os coeficientes $p = 0,125$ e $b = 0,85$ visto que:

- rugosidade do terreno: categoria IV - terrenos cobertos por obstáculos;
- dimensões da edificação: classe B - maior dimensão não excede 5000;

Considerando os níveis do pórtico apresentado na Figura (4.18), obtém-se:

- $S_2(z = 300) = 0,717$;
- $S_2(z = 600) = 0,781$;
- $S_2(z = 900) = 0,822$;
- $S_2(z = 1200) = 0,852$;
- $S_2(z = 1500) = 0,876$;

- $S_2(z = 1800) = 0,897$;
- $S_2(z = 2100) = 0,914$;
- $S_2(z = 2400) = 0,929$.
- fator S_3 (estatístico): edificação residencial - $S_3 = 1,0$;

A velocidade característica e a pressão aerodinâmica podem ser obtidas pelas equações A.2 e A.3, respectivamente:

$$V_k = V_0 \times S_1 \times S_2 \times S_3 \quad (\text{A.2})$$

$$q = 0,613 \times V_k \quad (\text{A.3})$$

Resultando em:

- $q(z = 300) = 13,618$;
- $q(z = 600) = 14,850$;
- $q(z = 900) = 15,622$;
- $q(z = 1200) = 16,194$;
- $q(z = 1500) = 16,652$;
- $q(z = 1800) = 17,036$;
- $q(z = 2100) = 17,368$;
- $q(z = 2400) = 17,660$.

A partir das respectivas áreas de influência da ação do vento e do coeficiente de arrasto para regime de alta turbulência ($C_a = 1,40$) obtido através dos ábacos encontrados na (ABNT (2023)), obtém-se as forças de arrasto a serem aplicadas no modelo:

$$F_a = C_a \times q \times A_e \quad (\text{A.4})$$

Deste modo, obtém-se:

- $F1 = F_a(z = 300) = 686,338$;
- $F2 = F_a(z = 600) = 748,457$;
- $F3 = F_a(z = 900) = 787,368$;
- $F4 = F_a(z = 1200) = 816,198$;
- $F5 = F_a(z = 1500) = 839,284$;
- $F6 = F_a(z = 1800) = 858,631$;
- $F7 = F_a(z = 2100) = 875,337$;
- $F8 = F_a(z = 2400) = 445,035$.

Os carregamentos obtidos poderiam ser divididos entre os pórticos da estrutura empregando uma análise de rigidez. Visto que, o foco desta pesquisa não é avaliar a distribuição de esforços provocados por ações de vento em estruturas, por simplificação, foi assumido que o carregamento total é aplicado no pórtico analisado.