

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

HOMOGENEIZAÇÃO DE UM MATERIAL COMPOSTO FORMADO POR UMA
MATRIZ POLIMÉRICA COM UMA SEGUNDA FASE PARTICULADA

por

Gilson Francisco Paz Soares

Tese para obtenção do Título de Doutor em Engenharia

Porto Alegre, novembro de 2010.

HOMOGENEIZAÇÃO DE UM MATERIAL COMPOSTO FORMADO POR UMA
MATRIZ POLIMÉRICA COM UMA SEGUNDA FASE PARTICULADA

por

Gilson Francisco Paz Soares
Mestre em Modelagem Matemática

Tese submetida ao Corpo Docente do Programa de Pós-Graduação em Engenharia Mecânica, PROMEC, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Doutor em Engenharia

Área de Concentração: Mecânica dos Sólidos

Orientador: Prof. Dr. Ignacio Iturrioz

Co-orientador: Prof. Dr. Adrian Cisilino

Aprovada por:

Prof. Dr. Herbert Martins Gomes - PROMEC/UFRGS

Prof. Dr. Inácio Benvegnu Morsch - PPGEC/UFRGS

Prof. Dr. Jun Sérgio Ono Fonseca- PROMEC/UFRGS

Prof. Dr. Wang Chong - UNIPAMPA

Prof. Dr. Horácio A Vielmo
Coordenador do PROMEC

Porto Alegre, 16 de novembro de 2010.

Dedico esta tese à minha família que sempre esteve ao meu lado, e também aos meus amigos, colegas e professores.

AGRADECIMENTOS

Ao término do doutorado, faço um agradecimento muito especial às seguintes pessoas e instituições que foram fundamentais à conclusão deste trabalho, seja de forma direta ou indireta:

- Professor Ignacio Iturrioz, meu orientador;
- Professor Adrian Cisilino, meu coorientador;
- À minha esposa Darlene e à minha filha Isadora;
- Aos meus pais e irmãs;
- A todos os colegas, sem exceção, desse período de doutorado, em especial ao Alexandre Molter, Ruben Galeano, Leandro Corso, Leonardo Hoss, Pablo Didoné, Tales Lisboa, Felipe Stumpf, André Casagrande, Luis Kostascki, Vicente Puglia e ao Otávio Pinto;
- Aos professores do GMAp, Rogério José Marczak, Jun Sérgio Ono Fonseca e Herbert Martins Gomes;
- Aos professores membros da banca de defesa;
- Aos funcionários da universidade, das secretarias, departamentos, bibliotecas, restaurante universitário, portarias, limpeza, PROMEC;
- À CAPES, pelo financiamento através da bolsa de doutorado;
- À UFRGS, pela oportunidade de cursar o doutorado;
- À Universidade de Mar Del Plata, pela participação através do coorientador desta tese.

RESUMO

O estudo numérico do comportamento estrutural de materiais compostos apresenta-se, desde os últimos anos, como um campo muito fértil de pesquisas, o que justifica o crescimento exponencial de trabalhos científicos nesta área. Atualmente é possível desenhar as propriedades físicas do material adequando-os ao uso que se queira dar a eles. Especificamente quanto às propriedades mecânicas vinculadas a função estrutural que o material em estudo possa ter, estas podem, na atualidade, ser quantificadas, modificadas e otimizadas. No presente trabalho, explora-se um material composto formado por uma matriz polimérica e uma segunda fase particulada com distribuição aleatória. Para realizar esse estudo foi utilizado o método dos elementos discretos e o método dos elementos finitos. No primeiro caso é apresentado um estudo onde se realizam tentativas de determinar o elemento de volume representativo considerando o comportamento do material como não-linear. No segundo caso, aplicando o método dos elementos finitos, realiza-se um estudo considerando a matriz e a fase particulada com comportamento elástico linear, determinando o elemento de volume representativo e comparando os resultados, em termos de constantes elásticas homogeneizadas, com propostas teóricas fornecidas pela micromecânica clássica. Um estudo da convergência da malha e exercícios de otimização foram realizados sobre o composto simulado. Finalmente, realiza-se um estudo não-linear através do método dos elementos finitos, onde a matriz é elástica e a fase particulada hiperelástica, onde se determina o elemento de volume representativo e se faz aplicações para verificação da eficácia dos resultados.

Palavras-chave: Estudo numérico; propriedades mecânicas; material composto; método dos elementos finitos; elemento de volume representativo.

ABSTRACT

The numerical study of the composite material mechanical behavior has shown lately a fertile field of research, which justifies the exponential growing of scientific works in this area. Nowadays it is possible to design the material properties adapting them to the usage that we want to give them. Specifically, regarding to the mechanical properties, there are methods that allow us to modify them in a rational way to reach different objectives. In the present work, different aspects are analyzed of a composite material built with a polymeric matrix and a second particulate phase with random distribution. The Discrete Element Method (DEM) and the Finite Element Method (FEM) were used to carry out the present work. Firstly, an application of DEM, a study of different alternatives is shown to determine the volume representative element (RVE) considering the non-linear behavior of the studied material. Secondly, an application of FEM, a study considering the matrix and particulate phase, both, with linear elastic behavior. This application consists on computing the RVE and comparing these results with analytical proposals available in the Micromechanics classical bibliography. A mesh convergence study of the FEM models used and simple applications of optimization are also presented. Finally, another application of FEM is presented. In this case a non linear study is shown, where the matrix is considered linear elastic, and the particulate phase is hyperelastic. In this case the RVE was determined and some applications to verify the consistency of the results obtained are presented.

Keywords: Numerical study; mechanical properties; composite material; Finite Element Method; volume representative element.

ÍNDICE

1	INTRODUÇÃO	1
2	FUNDAMENTOS TEÓRICOS.....	5
2.1	Introdução à Mecânica dos Sólidos.....	5
2.1.1	Formulação para Elasticidade Infinitesimal.....	5
2.1.1.1	Tensor de deformações – Equações de equilíbrio.....	5
2.1.1.2	Tensor de tensões – Equações de equilíbrio.....	7
2.1.1.3	Relações constitutivas.....	8
2.1.1.4	Problema da mecânica dos sólidos linear elástica com deslocamentos e deformações infinitesimais.....	9
2.1.1.5	Tensões principais.....	9
2.1.2	Problemas não-lineares na mecânica dos sólidos.....	11
2.1.2.1	Relação geral entre deformações e deslocamentos.....	11
2.1.2.2	Definição de uma medida de deformação.....	12
2.1.2.3	Medida de tensão.....	18
2.1.2.4	As componentes das tensões.....	19
2.1.2.5	Relações constitutivas para materiais elásticos.....	22
2.2	Aspectos Básicos da Mecânica da Fratura.....	25
2.2.1	Critério Energético de Griffith.....	28
2.2.2	Teoria de Griffith Modificada.....	31
2.2.3	Fator de Intensidade de Tensões.....	32
2.3	Materiais Compósitos e Tratamento Computacional de Materiais Micro-heterogêneos.....	36
2.3.1	Materiais Compostos.....	36
2.3.1.1	Elemento de Volume Representativo (EVR).....	37
2.3.1.2	Homogeneização.....	37
2.3.1.3	Solução de Voigt e Reuss.....	38
2.3.1.4	Solução de Eshelby.....	39
2.3.1.5	Definição do tensor de concentração.....	40
2.3.2	Tratamento Computacional de Materiais Micro-heterogêneos.....	43
2.3.2.1	Materiais compostos periódicos.....	45
2.3.2.2	Tratamento de materiais compósitos sem periodicidade.....	47
2.4	Fundamentos de Algoritmos Genéticos.....	51
2.4.1	Otimização com Algoritmos Genéticos.....	51
2.4.2	Estrutura dos Algoritmos Genéticos.....	52
2.4.3	Abordagem Multi-objetivo.....	54
2.5	Modelagem Numérica.....	54
2.5.1	Método dos Elementos Finitos (MEF).....	55
2.5.2	Método dos Elementos Discretos (MED).....	55
2.5.2.1	Características do MED Utilizado neste Trabalho.....	56
2.6	Resolução de Problemas Não-lineares através do MEF.....	59
2.6.1	Análise explícita.....	60

3	REVISÃO BIBLIOGRÁFICA.....	63
3.1	Polímeros reforçados com partículas de borracha.....	63
3.2	Homogeneização linear elástica.....	64
3.3	Homogeneização elástica não-linear.....	67
3.4	Otimização com Algoritmos Genéticos.....	68
4	METODOLOGIA.....	70
4.1	Homogeneização com distribuição aleatória dos micro-constituintes.....	70
4.2	Simulação da distribuição aleatória dos nódulos na matriz polimérica.....	71
4.3	Estudo de convergência da malha.....	71
4.4	Otimização de modelos com Algoritmos Genéticos.....	72
5	ESTUDOS REALIZADOS COM O MÉTODO DOS ELEMENTOS FINITOS.....	74
5.1	Homogeneização numérica do composto com distribuição aleatória de heterogeneidades esféricas.....	75
5.1.1	Caracterização do modelo.....	75
5.1.2	Estudo de convergência.....	76
5.1.2.1	Matriz e heterogeneidades com malhas iguais.....	77
5.1.2.2	Matriz e heterogeneidades com malhas diferentes.....	79
5.1.2.3	Escolha da malha.....	82
5.1.3	Determinação do elemento de volume representativo (EVR).....	83
5.1.4	Resultados e discussões da etapa.....	86
5.1.4.1	Convergência dos parâmetros K e G.....	86
5.1.4.2	Influência da fração de volume de heterogeneidades na determinação do EVR.....	87
5.1.4.3	Comparação dos resultados obtidos do MEF com modelos analíticos.....	89
5.1.4.4	Estudo paramétrico da influência da distância mínima entre as heterogeneidades nos resultados de K e G de amostras com 10 nódulos (EVR).....	92
5.2	Homogeneização numérica elástica não-linear de um composto com distribuição aleatória de heterogeneidades esféricas hiperelásticas dentro de uma matriz de PMMA.....	94
5.2.1	Caracterização do modelo.....	94
5.2.2	Caracterização do material elástico não-linear através de um estado de tensão-deformação característico e do conceito de “material map”.....	97
5.2.3	Malha.....	100
5.2.4	Tempo de análise e velocidade de deformação.....	100
5.2.5	Determinação do elemento de volume representativo.....	104
5.2.6	Análise e discussão dos resultados da etapa.....	121
5.2.7	Aplicações com o EVR determinado.....	124
5.2.7.1	Aplicação 1: Teste de tração uniaxial.....	124
5.2.7.2	Aplicação 2: Teste de corte puro.....	128
6	ESTUDOS REALIZADOS COM ALGORITMOS GENÉTICOS.....	132

6.1	Caracterização do modelo	132
6.2	Implementação do algoritmo genético.....	132
6.3	Resultados e discussões da etapa.....	133
6.3.1	Caso A.....	133
6.3.2	Caso B.....	135
6.3.3	Caso C.....	138
7	ESTUDOS REALIZADOS COM O MÉTODO DOS ELEMENTOS DISCRETOS.....	140
7.1	Caracterização do modelo.....	140
7.2	Determinação do elemento de volume representativo (EVR).....	141
7.3	Resultados e discussões da etapa.....	143
7.3.1	Convergência dos parâmetros ENEL/V e ENGD/V.....	143
8	CONCLUSÕES.....	146
8.1	Conclusões da etapa de implementações com MEF.....	146
8.1.1	Caso elástico linear.....	146
8.1.2	Caso elástico não- linear.....	147
8.2	Conclusões da etapa de implementações com AG.....	148
8.3	Conclusões da etapa de implementações com MED.....	149
8.4	Sugestões para trabalhos futuros.....	150
	REFERÊNCIAS BIBLIOGRÁFICAS.....	152
	APÊNDICE A.....	162
	APÊNDICE B.....	172
	APÊNDICE C.....	216

LISTA DE FIGURAS

Figura 2.1	Configurações indeformada e deformada do corpo tridimensional contínuo.....	6
Figura 2.2	Cubo infinitesimal extraído do corpo principal com as tensões atuantes.....	7
Figura 2.3	Deslocamento relativo na configuração indeformada e deformada	13
Figura 2.4	Barra longa sofrendo uma extensão na direção axial	18
Figura 2.5	Corpo contínuo de domínio B, sobre o qual se define uma tensão atuante	18
Figura 2.6	Paralelepípedo elementar que mostra a decomposição da tensão \vec{T}	20
Figura 2.7	Esquema que facilita entender a definição dos tensores de tensões.....	22
Figura 2.8	Característica de um corpo elástico.....	22
Figura 2.9	Representação das direções principais de deformação.....	25
Figura 2.10	Caso considerado por Inglis para sua dedução [Kanninen e Popelar, 1985].....	26
Figura 2.11	Experiência realizada por Griffith com as fibras de vidro [Kanninen e Popelar, 1985].....	29
Figura 2.12	Placa de dimensões infinitas utilizada pelo Griffith.....	30
Figura 2.13	Modos de fratura considerados por Irwin.....	32
Figura 2.14	Sistema de coordenadas utilizado na dedução do fator de intensidade de tensão (K_I).....	33
Figura 2.15	Figura esquemática mostrando a equivalência entre o campo de tensões na ponta da trinca entre uma estrutura real e um corpo de prova [Anderson, 1995].....	34
Figura 2.16	Quadro ilustrativo da relação entre G e K_I	35
Figura 2.17	Representação esquemática das considerações realizadas por Voigt e Reuss para estabelecer os limites das propriedades equivalentes de compostos, utilizando a análise numa dimensão [Zohdi e Wriggers, 2005].....	38
Figura 2.18	Esboço de um material composto periódico.....	44

Figura 2.19	Exemplos de materiais compostos de micro-estrutura aleatória. A) ferro fundido nodular, b) matriz de alumínio com partículas de carbeto de silício, c) Liga alumínio-titânio d) Matriz de prata com partículas de ouro [Zohdi, 2002].....	45
Figura 2.20	Esboço de um material periódico micro-heterogêneo.....	46
Figura 2.21	Esquema ilustrativo do tamanho do EVR [Zohdi, 2002].....	48
Figura 2.22	Desenho esquemático para explicar a homogeneidade estatística.....	49
Figura 2.23	Seqüência de volumes de material numa análise de EVR [Zohdi, 2002].....	49
Figura 2.24	Verificação da independência das condições de contorno com o incremento do volume de controle encontrado por Terada et al., 2000.....	50
Figura 2.25	Convergência da resposta em termos do módulo de cisalhamento da amostra de material (G^* representado por μ^*), com o incremento do número de partículas [Zohdi, 2002].....	50
Figura 2.26	Modelo de barras gerado usando o Método dos Elementos Discretos e detalhe do módulo cúbico.....	57
Figura 2.27	Lei constitutiva bilinear atribuída a cada barra do modelo, (a) parâmetros envolvidos na lei constitutiva, (b) comportamento em tração e compressão [Rocha, 1989].....	58
Figura 3.1	Verificação da independência das condições de contorno aplicadas, no valor do módulo volumétrico (k^{app}) obtido, com o incremento do volume de controle (V) [Kanit et al., 2003].....	65
Figura 4.1	Domínio de material micro-heterogêneo não periódico subdividido em volumes de controle.....	70
Figura 4.2	Fluxograma da implementação do algoritmo genético.....	73
Figura 5.1	Exemplo de distribuição do composto (a); Exemplo de malha de elementos finitos para o composto (b).....	77
Figura 5.2	Gráfico de barras indicando as médias pertencentes à mesma distribuição de probabilidade.....	79
Figura 5.3	Efeito principal da malha da matriz.....	82
Figura 5.4	Resultados de $K[N/\mu m^2]$ para diferentes refinamentos (N =razão da	

	malha dos nódulos e M =razão da malha da matriz). As curvas abaixo são das malhas mais refinadas.....	83
Figura 5.5	Resultados de G [$N/\mu m^2$] para diferentes refinamentos (N =razão da malha dos nódulos e M =razão da malha da matriz). As curvas abaixo são das malhas mais refinadas.....	83
Figura 5.6	Esquema mostrando malha, condições de contorno e distribuição aleatória de 2 nódulos esféricos de uma amostra.....	85
Figura 5.7	Comportamento do módulo de compressibilidade K com o incremento do número de nódulos para um volume de nódulos de 10% em relação ao volume da amostra.....	87
Figura 5.8	Comportamento do módulo de elasticidade transversal G com o incremento do número de nódulos para um volume de nódulos de 10% em relação ao volume da amostra.....	88
Figura 5.9	Comportamento do EVR para diferentes frações de volume de heterogeneidades. Foram analisados os valores de K e G com erros admissíveis de 0,5%, 1% e 1,5% em relação à maior amostra (64 nódulos).....	88
Figura 5.10	Comportamento do EVR para diferentes frações de volume de heterogeneidades, com erro=1% e critério G	89
Figura 5.11	Comparação do módulo de compressibilidade K obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos.....	90
Figura 5.12	Detalhe sem os limites de Voigt e Reuss da comparação do módulo de compressibilidade K obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos.....	90
Figura 5.13	Comparação do módulo de elasticidade transversal G obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos.....	91
Figura 5.14	Detalhe sem os limites de Voigt e Reuss da comparação do módulo de elasticidade transversal G obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos.....	91
Figura 5.15	Corte em um modelo com 10 heterogeneidades e distância mínima entre elas de $0,005r$	92
Figura 5.16	Corte em um modelo com 10 heterogeneidades e distância mínima	

	entre elas de $0,25r$	93
Figura 5.17	Evolução do módulo K com o aumento da distância mínima entre as heterogeneidades, onde K' é o valor de K em $0,25r$	93
Figura 5.18	Evolução do módulo G com o aumento da distância mínima entre as heterogeneidades, onde G' é o valor de G em $0,25r$	94
Figura 5.19	Curvas experimental e ajustada, do modelo hiperelástico utilizado para as heterogeneidades.....	95
Figura 5.20	Algoritmo para criação do mapeamento discreto do material elástico não-linear proposto por Temizer e Zohdi, 2007.....	98
Figura 5.21	Mapeamento discreto de um material de Mooney-Rivlin em uma formulação bidimensional [Temizer e Zohdi, 2007].....	99
Figura 5.22	Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 2 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	101
Figura 5.23	Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 4 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	101
Figura 5.24	Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 8 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	102
Figura 5.25	Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 16 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	102
Figura 5.26	Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 32 heterogeneidades; alfa(i)' é o	

	valor de alfa(i) obtido com a menor velocidade de deformação usada.....	103
Figura 5.27	Estudo de convergência dos valores de alfa em relação à velocidade de deformação λ_E^3 para uma amostra de 64 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	103
Figura 5.28	Comparação dos valores de alfa na velocidade $\lambda_E^3 = 6.94e + 6$, entre os tamanhos de amostras estudados; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada.....	104
Figura 5.29	Esquema com os pontos do domínio λ_E^i mapeados para determinação dos α_i da lei constitutiva do material. Acima esquema ilustrativo e abaixo exemplo do mapeamento gerado em Matlab para o caso de α_0 de 32 heterogeneidades e $\lambda_E^1 = 0$, com os pontos mapeados.....	105
Figura 5.30	Mapeamento dos valores de α_{0ij} médios do composto com a fixação de $\lambda_E^1 = 0$, para todos os tamanhos de amostras.....	106
Figura 5.31	Convergência do índice I_0 em relação ao tamanho da amostra.....	108
Figura 5.32	Convergência do índice I_1 em relação ao tamanho da amostra.....	109
Figura 5.33	Convergência do índice I_2 em relação ao tamanho da amostra.....	109
Figura 5.34	Convergência dos índices I_l em relação ao tamanho da amostra.....	110
Figura 5.35	Detalhe da figura 5.34 com escala do índice.....	110
Figura 5.36	Pontos 1, 2, 5, 6, 9 e 10 da tabela 5.8 para $\lambda_E^3 = -0.1039$	111
Figura 5.37	Pontos 3, 4, 7, 8, 11 e 12 da tabela 5.8 para $\lambda_E^3 = 0.20805$	112
Figura 5.38	Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).....	112
Figura 5.39	Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 0.90$,	

	$\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos).....	113
Figura 5.40	Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos).....	113
Figura 5.41	Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).....	114
Figura 5.42	Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos).....	114
Figura 5.43	Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos).....	115
Figura 5.44	Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).....	115
Figura 5.45	Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos).....	116
Figura 5.46	Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos).....	116
Figura 5.47	Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa0'	

	é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).....	117
Figura 5.48	Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos).....	117
Figura 5.49	Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos).....	118
Figura 5.50	Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).....	118
Figura 5.51	Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos).....	119
Figura 5.52	Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos).....	119
Figura 5.53	Evolução do coeficiente de variação dos α_{ij} para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ (Ponto 2 da tabela 5.8) em relação ao tamanho da amostra.....	120
Figura 5.54	Evolução do coeficiente de variação dos α_{ij} para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ (Ponto 12 da tabela 5.8) em relação ao tamanho da amostra.....	120
Figura 5.55	Coefficiente de variação (%) de 4 amostras de 2 heterogeneidades vs $\Delta\lambda_F^{1-3}$, sendo para todos os casos $\lambda_F^1 = 0.90$ e $\lambda_F^2 = 1.145$	123
Figura 5.56	Condições de contorno aplicadas no teste de tração uniaxial. Deslocamento prescrito aplicado em uma direção com as outras duas	

	livres.....	125
Figura 5.57	Comparação entre as curvas tensão x deformação do teste de tração uniaxial do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades (Tabela 5.10).....	126
Figura 5.58	Comparação entre as curvas tensão x deformação da matriz de PMMA, das heterogeneidades elastoméricas e do ensaio de tração uniaxial do material composto resultante.....	127
Figura 5.59	Detalhe das inclinações iniciais das curvas de tensão x deformação da matriz de PMMA e do material composto resultante para o ensaio de tração uniaxial.....	127
Figura 5.60	Detalhe da inclinação inicial da curva de tensão x deformação das heterogeneidades elastoméricas.....	128
Figura 5.61	Detalhe da curva do material elastomérico hiperelástico utilizado, até o nível de deformação analisado.....	128
Figura 5.62	Condições de contorno aplicadas no teste de corte puro. Deslocamento prescrito aplicado em uma direção, restrição em uma direção e a terceira livre.....	129
Figura 5.63	Comparação entre as curvas tensão x deformação do teste de tração uniaxial do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades (Tabela 5.11).....	130
Figura 5.64	Comparação entre as curvas tensão x deformação da matriz de PMMA, das heterogeneidades elastoméricas e do ensaio de corte puro do material composto resultante.....	130
Figura 5.65	Detalhe das inclinações iniciais das curvas de tensão x deformação da matriz de PMMA e do material composto resultante para o ensaio de corte puro.....	129
Figura 6.1	Esquema mostrando malha e distribuição regular de 8 nódulos esféricos do modelo utilizado.....	133
Figura 6.2	Gráfico mostrando o erro do valor obtido em relação ao desejado e a variação do percentual das combinações realizadas em relação às	

	possíveis, para cada número de indivíduos da população inicial.....	137
Figura 6.3	Distribuições de tensões de Von Mises no modelo regular de 8 nódulos esféricos, a) propriedades do estudo do EVR de Soares et al., 2008; b) propriedades otimizadas para minimizar as tensões, em diferentes escalas.....	139
Figura 7.1	Corte do modelo cúbico com uma heterogeneidade esférica de borracha onde a região: em azul é um nódulo interno de PMMA (raio $r=10 \times 10^{-9} \text{m}$); em cinza é borracha ($r=25 \times 10^{-9} \text{m}$); em rosa é a interface matriz/borracha ($r=27.5 \times 10^{-9} \text{m}$) e o restante é a matriz cúbica (lado $L=100 \times 10^{-9} \text{m}$).....	141
Figura 7.2	Curva Tensão x Deformação indicando os pontos de análise, $\sigma_c = 60 \text{MPa}$	142
Figura 7.3	Resultados da densidade de energia elástica para amostras contendo número crescente de nódulos de borracha, regime linear elástico (P_1). As barras de erros indicam a dispersão nos resultados, e $(\text{ENEL}/V)'$ é o valor de ENEL/V médio para a amostra de 50 heterogeneidades.....	143
Figura 7.4	Resultados da densidade de energia de dano para amostras contendo número crescente de nódulos de borracha, para o regime não-linear (P_2). As barras de erros indicam a dispersão nos resultados, e $(\text{ENGD}/V)'$ é o valor de ENGD/V médio para a amostra de 50 heterogeneidades.....	144
Figura 7.5	Resultados da densidade de energia de dano para amostras contendo número crescente de nódulos de borracha, para o regime de tensão máxima (P_3). As barras de erros indicam a dispersão nos resultados, e $(\text{ENGD}/V)'$ é o valor de ENGD/V médio para a amostra de 50 heterogeneidades.....	145
Figura A1	Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados.....	163
Figura A2	Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados.....	164

Figura A3	Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados.....	165
Figura A4	Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados.....	166
Figura A5	Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados.....	167
Figura A6	Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados.....	168
Figura A7	Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados.....	169
Figura A8	Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados.....	170
Figura A9	Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados.....	171
Figura B1	Diagrama de blocos do algoritmo utilizado para calcular as propriedades efetivas equivalentes do composto elástico-linear (estudo descrito na seção 5.1).....	173
Figura C1	Diagrama de blocos do algoritmo utilizado para realizar a homogeneização do composto elástico não-linear (estudo descrito na seção 5.2).....	217

LISTA DE TABELAS

Tabela 5.1	Resultados do módulo de compressibilidade para diferentes razões tamanho da malha máximo/raio da heterogeneidade, considerando a mesma malha para a matriz e heterogeneidades.....	77
Tabela 5.2	Análise de Variância considerando o efeito de um fator (mesma malha para matriz e heterogeneidades).....	78
Tabela 5.3	Resultados de módulo de compressibilidade para diferentes razões tamanho da malha máximo/raio da heterogeneidade, considerando malhas diferentes para a matriz e heterogeneidades.....	80
Tabela 5.4	Resumo da Tabela 5.3.....	80
Tabela 5.5	Análise de Variância considerando o efeito de dois fatores e a interação entre os mesmos (malhas diferentes para matriz e heterogeneidades).....	81
Tabela 5.6	Variáveis envolvidas na análise explícita.....	96
Tabela 5.7	Tempos de análise explícita aplicados e suas respectivas velocidades de deformação.....	100
Tabela 5.8	Estados de deformação para análise da convergência dos valores de α_{ij} médios.....	111
Tabela 5.9	Coeficiente de variação (%) de 4 amostras de 2 heterogeneidades com estados de carga diferentes mostrando a instabilidade da matriz de deformação para deformações próximas.....	122
Tabela 5.10	Número de nós e elementos das malhas dos modelos estudados	124
Tabela 5.11	Valores de α_i e T_2^3 obtidos através da metodologia <i>material map</i> no domínio de deformações aplicado ao ensaio de tração uniaxial.....	126
Tabela 5.12	Valores de α_i e T_2^3 obtidos através da metodologia <i>material map</i> no domínio de deformações aplicado ao ensaio de corte puro.....	129
Tabela 6.1	Número de bits e limites aplicados às variáveis de estudo do Caso A.	134
Tabela 6.2	Lista de polímeros utilizada no estudo do caso B. Valores médios [NDSM, 2009].....	135

Tabela 6.3	Número de bits e limites aplicados às variáveis de estudo do Caso B.	136
Tabela 7.1	Propriedades físicas dos materiais do modelo em elementos discretos.....	140
Tabela 7.2	Tabela dos desvios indicando a dispersão dos resultados.....	144

1 INTRODUÇÃO

O grande avanço nos estudos da engenharia e da ciência dos materiais tem proporcionado o projeto de produtos cada vez mais eficazes e seguros. Exemplo disto é a crescente utilização de materiais compostos em aplicações de alta precisão na engenharia, o que gera a necessidade de um amplo estudo de sua caracterização mecânica. Esses materiais são formados por duas fases ou mais com propriedades mecânicas diferentes. Quando se mistura o material empregado como matriz com uma segunda ou mais fases é possível modificar as propriedades mecânicas do conjunto, que pode apresentar propriedades totalmente diferentes às dos materiais que fizeram a composição. Essa característica permite, utilizada de forma adequada, a fabricação de materiais sob medida, satisfazendo necessidades tecnológicas específicas, muitas vezes com melhor viabilidade econômica caso fossem utilizados materiais tradicionais. Entretanto, devido à natureza heterogênea desses materiais, faz-se necessário uma caracterização rigorosa do ponto de vista mecânico, estabelecendo-se assim alguns limites para sua aplicação.

A exploração desse tipo de material requer uma caracterização mecânica mais completa, que possibilite prever e entender mecanismos de formação de defeitos que possam surgir perante a ação de determinadas solicitações mecânicas. Compreender este universo contribui com a elaboração de projetos de produtos cada vez mais eficazes e seguros, além de proporcionar a elaboração de estratégias de exploração e manutenção mais eficazes.

O estudo das condições de propagação instável de defeitos macroscópicos, cuja iniciação e propagação dependem de processos de fratura que ocorrem em escalas estruturais inferiores (mesoescala), é desenvolvido à luz dos conceitos da Mecânica da Fratura. O processo de investigação experimental destes processos ao nível da mesoescala requer a aplicação de técnicas experimentais que não são simples, o que limita sua disponibilidade a grandes centros de pesquisa.

Diante disto, o modelamento numérico computacional torna-se um importante aliado na pesquisa do comportamento de materiais micro-heterogêneos, pois se apresenta como uma alternativa mais acessível. A simulação de estruturas dos mais diversos tipos de materiais submetidos às mais diversas condições, em conjunto com dados obtidos experimentalmente, fornece ao pesquisador informações cruciais para o entendimento mais amplo do comportamento dos materiais compostos.

Alguns métodos apresentam, por suas características específicas, um grande potencial para a análise numérica do comportamento geral das estruturas micro-heterogêneas. Entre eles, têm grande aplicação o Método dos Elementos Discretos (MED) e o Método dos Elementos Finito (MEF). O primeiro considera o meio sólido a ser modelado como um conjunto de massas com distribuição espacial uniforme, interconectadas entre si por um sistema de barras de rigidez equivalente ao contínuo que se quer modelar. Se em alguma região do modelo as tensões ou deformações ultrapassarem os limites impostos pela relação constitutiva da barra, aquelas que estiverem dentro desta região rompem-se, originando de maneira natural um defeito local e sua posterior propagação. O segundo utiliza como parâmetro as variáveis nodais de um número finito de pontos escolhidos previamente, chamados de nós. O domínio de integrações é subdividido em várias regiões, ou elementos finitos, interconectadas entre si através de um número discreto de pontos nodais. Para cada elemento se estabelece um comportamento local aproximado, de tal forma que as incógnitas do problema em qualquer ponto do elemento podem ser definidas em função das mesmas incógnitas nos pontos nodais do elemento. Após, minimizando o funcional do problema, obtido das somas das contribuições de cada elemento, chega-se a um sistema total de equações em que a solução permite conhecer os valores das incógnitas nos pontos nodais.

Outro aspecto a ser considerado é o da otimização estrutural. O conceito de material composto pode permitir ao engenheiro “criar” um novo material em que é possível, por meio da combinação de componentes, aplicar uma técnica de otimização computacional para minimizar características e efeitos indesejáveis e valorizar outras desejáveis. Vários aspectos do comportamento do material podem ser manipulados no projeto de um composto, a saber: resistência estática e à fadiga; rigidez; resistência à corrosão; resistência à abrasão; redução de peso; capacidade de trabalho a alta e baixa temperatura; isolamento ou condutividade térmica, elétrica ou acústica; dureza; ductilidade e aparência estética. Para a otimização de problemas que envolvem microestruturas existem algumas técnicas, entre elas, a dos Algoritmos Genéticos, que são conjuntos de procedimentos e técnicas baseados nos princípios da genética populacional, onde sistemas de indivíduos interagindo com um ambiente evoluem, através de reprodução, seleção e mutação. Quando se trabalha com parâmetros micro estruturais, estes formam uma "cadeia genética". Posteriormente, conceitos evolutivos são aplicados a uma população destas cadeias.

Objetivos do trabalho

O objetivo principal deste trabalho consiste em desenvolver estratégias para o estudo do comportamento micro-mecânico das propriedades de materiais compostos particulados utilizando diferentes ferramentas numéricas, como o método dos elementos discretos e o método dos elementos finitos.

Para tanto, os seguintes objetivos específicos fazem parte do processo:

Objetivos específicos

a) Usar as técnicas de homogeneização para determinar o elemento de volume representativo (EVR) de um material composto bifásico (estudo em três dimensões), considerando o comportamento linear elástico do material, com a utilização do método dos elementos finitos (MEF);

b) Usar as técnicas de homogeneização para determinar o elemento de volume representativo (EVR) de um material composto bifásico (estudo em três dimensões), considerando o comportamento não-linear do material, através do método dos elementos discretos (MED);

c) Realizar a otimização da estrutura do EVR do caso linear elástico empregando o método dos algoritmos genéticos (AG).

d) Usar as técnicas de homogeneização para determinar o elemento de volume representativo (EVR) de um material composto bifásico (estudo em três dimensões), considerando o comportamento elástico não-linear do material, com a utilização do método dos elementos finitos (MEF);

O presente trabalho está estruturado da seguinte forma: Após a presente introdução onde é descrita de maneira breve a motivação do trabalho e os objetivos propostos, no capítulo dois são apresentados os fundamentos teóricos considerando-se aspectos relacionados com temas abordados no trabalho. No capítulo três é realizada uma revisão bibliográfica a fim de verificar o estado da arte quanto aos aspectos relacionados com o estudo realizado. Logo após, no capítulo quatro, são apresentados os métodos numéricos utilizados (MEF e MED), bem como a metodologia utilizada para realizar o estudo de convergência da malha e da otimização. No capítulo cinco são mostradas as implementações realizadas no MEF, enquanto que no seis e no sete, as implementações com AG e MED, respectivamente. No Capítulo oito são apresentadas as conclusões obtidas neste trabalho e sugestões para continuação futura do mesmo. Por fim, têm-se as referências bibliográficas, o apêndice A que apresenta as curvas do mapeamento discreto do material realizado na homogeneização não-linear, o apêndice B com

a descrição do algoritmo utilizado para fazer o estudo de elementos finitos do caso linear elástico e o apêndice C com a descrição do algoritmo utilizado para fazer o estudo de elementos finitos do caso elástico não-linear.

2 FUNDAMENTOS TEÓRICOS

O estudo realizado envolve conceitos de mecânica dos sólidos, mecânica da fratura na macro e mesoescala, modelagem computacional de materiais heterogêneos (compostos), otimização através dos algoritmos genéticos e modelagem numérica, em particular, os métodos dos elementos finitos e discretos. Neste capítulo, serão apresentadas as bases teóricas pertinentes aos conceitos envolvidos no estudo, começando por um apanhado geral da mecânica dos sólidos e seguindo pela mecânica da fratura com seus parâmetros macroscópicos e suas características. Logo após, serão mostradas as particularidades do tratamento computacional de materiais micro heterogêneos, em que alguns aspectos da micro mecânica também serão abordados. Em seguida serão abordados conceitos básicos de algoritmos genéticos para otimização de problemas de engenharia. Por fim será dada uma abordagem geral sobre os métodos numéricos empregados neste trabalho.

2.1 Introdução à Mecânica dos Sólidos

2.1.1 Formulação para Elasticidade Infinitesimal

Seja um corpo ou estrutura tridimensional sujeito a cargas em sua superfície, cargas de corpo como, por exemplo, o peso próprio e fixado em alguns pontos de seu contorno, como se pode observar na Figura 2.1. Na figura pode-se verificar as configurações indeformada (antes da aplicação das cargas) e deformada (após a aplicação das cargas). O vetor u representa o deslocamento de um ponto genérico \mathbf{P} de sua posição inicial (na configuração indeformada) até sua nova posição na configuração deformada. Considerando aqui a hipótese do meio contínuo, na qual se desconsidera a microestrutura do material, o vetor deslocamento é tratado como uma função contínua da posição inicial, isto é, para cada ponto \mathbf{x} do corpo existe um vetor $\mathbf{u}(\mathbf{x})$.

2.1.1.1 Tensor de deformações – Equações de equilíbrio

A partir dos deslocamentos é possível calcular as deformações (ε) em qualquer ponto do corpo. No caso geral, considerando a configuração indeformada como referência, as relações deformação-deslocamento podem se expressas pela equação:

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i} \cdot \frac{\partial u_k}{\partial x_j} \right) \quad (2.1)$$

A equação acima constitui as relações de deformação-deslocamento do meio contínuo para grandes deformações ou deformações finitas.

Se as componentes de deformação são suficientemente pequenas de forma que o produto de suas derivadas primeiras resulte num valor muito pequeno, o terceiro termo (não-linear) da equação (2.1) pode ser desprezado e as relações de deformação-deslocamento tornam-se lineares, da forma:

$$\varepsilon_{ij} = \frac{1}{2} \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.2)$$

A equação acima é o tensor de deformações de Cauchy e representa as relações de deformação-deslocamento infinitesimais, na configuração deformada ou indeformada, visto que nesta ordem de aproximação isto é indiferente.

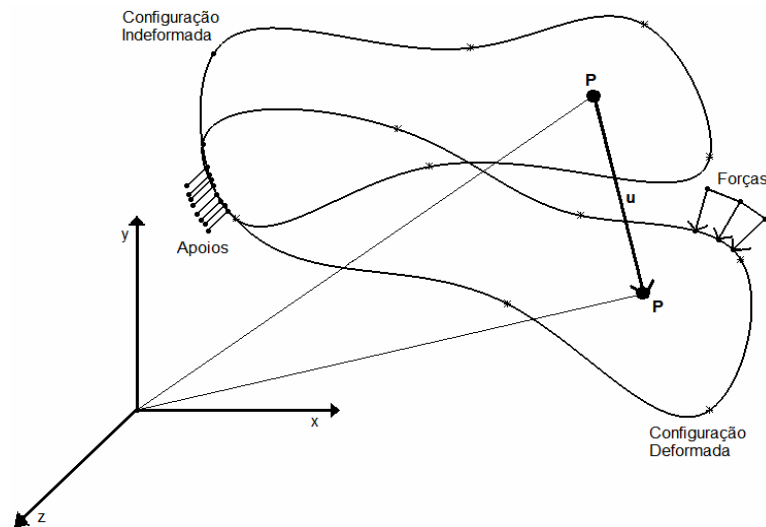


Figura 2.1 – Configurações indeformada e deformada do corpo tridimensional contínuo

Como apenas as três componentes do deslocamento u determinam as seis componentes do tensor de deformação, existe uma redundância de deformações em relação aos deslocamentos, ou seja, um tensor de deformação pode não representar um campo de deslocamento válido. Para garantir a validade das deformações, há um conjunto de equações que deve ser satisfeito por um campo de deformações para que este represente um campo de

deslocamentos válido. Este conjunto chamado de equações de compatibilidade, e representado por:

$$\epsilon_{qsp} \epsilon_{klm} \frac{\partial^2 \epsilon_{pm}}{\partial x_l \partial x_s} = 0 \quad (2.3)$$

onde ϵ é o operador de permutação:

$$\epsilon_{ijk} = \begin{cases} 0 & \text{se } ijk \text{ tiver elementos repetidos} \\ 1 & \text{se } ijk \text{ for uma permutação cíclica} \\ -1 & \text{se } ijk \text{ for uma permutação anticíclica} \end{cases} \quad (2.4)$$

2.1.1.2 Tensor de tensões - Equações de equilíbrio

Para o estudo do estado de tensões no interior do corpo deformado, o procedimento adotado é isolar um elemento infinitesimal de volume dv nas vizinhanças de um ponto P de um meio contínuo, com as faces paralelas aos planos coordenados, sobre as quais são definidas as tensões normais e tangenciais (Figura 2.2). Com o propósito de expressar o vetor força em qualquer direção, a tensão é representada por um tensor. O tensor de tensões Cauchy expressa os esforços internos na partícula dv do corpo sólido, em todas as direções. A força interna distribuída de um ponto sobre seu vizinho é determinada pela direção do vizinho n através da fórmula de Cauchy:

$$t_i = \sigma_{ij} n_j \quad (2.5)$$

onde t é a força distribuída em N/m^2 e σ é o tensor de tensão.

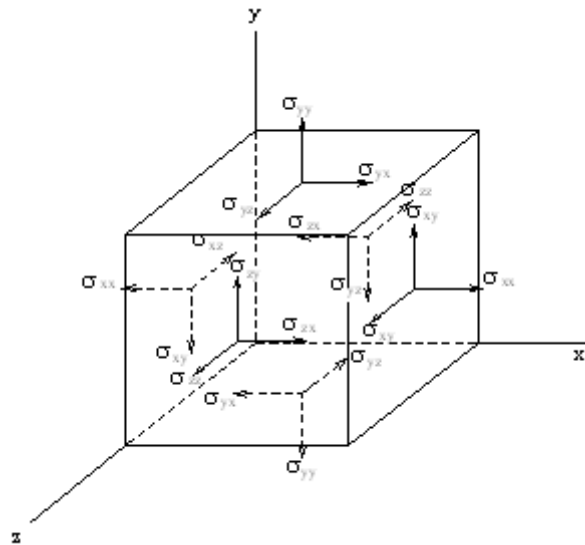


Figura 2.2 – Cubo infinitesimal extraído do corpo principal com as tensões atuantes

As cargas atuantes no corpo podem provocar uma variação de tensão de uma face para outra do cubo infinitesimal. Diante disso, é possível deduzir as equações de equilíbrio para esse cubo. A Equação (2.6) a seguir representa as três equações de equilíbrio dos momentos, que garantem a simetria do tensor de tensões e as três equações de equilíbrio das forças:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho \cdot b_i = 0 \quad (2.6)$$

onde $\frac{\partial \sigma_{ij}}{\partial x_j}$ é a derivada de uma tensão σ_{ij} na direção x_j em m, ρ é a densidade do volume dv e b_i é uma força de corpo na direção i . Trabalhando-se na hipótese de pequenas deformações e deslocamentos, não é necessário especificar sobre que configuração (deformada ou indeformada) se considera o cubo elementar para calcular as tensões.

2.1.1.3 Relações constitutivas

Até o momento foi visto como se dá a interação entre as partes de um material através das forças e como ocorrem as relações entre deformações e deslocamentos. Agora será apresentado como se desenvolve as relações entre tensões e deformações em um corpo. Tais relações são expressas pelas equações constitutivas, que dependem do tipo do material do qual é constituído o corpo sólido. No caso de sólidos elásticos lineares homogêneos, a relação obedece à Lei de Hooke, isto é, o tensor de tensão é linearmente proporcional ao tensor deformação:

$$\sigma_{ij} = C_{ijkl} \cdot \epsilon_{kl} \quad (2.7)$$

onde C_{ijkl} é um tensor de quarta ordem formado por constantes elásticas, que genericamente possui 81 elementos.

A partir de condições energéticas é possível determinar que o este tensor é simétrico e, no caso de se estudar um meio isotrópico, reduz-se o número de constantes independentes para apenas duas [Sokolnikoff, 1956]:

$$C_{ijkl} = K \delta_{ij} \delta_{kl} + 2G \left(\delta_{ik} \delta_{jl} - \frac{1}{3} \delta_{ij} \delta_{kl} \right) \quad (2.8)$$

onde K é o módulo de incompressibilidade do material, G é o módulo de elasticidade transversal e δ é o delta de Kronecker, definido como:

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (2.9)$$

Essas grandezas relacionam-se com o módulo de elasticidade longitudinal de Young (E) e coeficiente de Poisson (ν) através das expressões:

$$K = \frac{E}{3(1-2\nu)} \quad (2.10)$$

$$G = \frac{E}{2(1+\nu)} \quad (2.11)$$

Também se faz necessária a determinação das condições de contorno, afim de que se possa determinar a solução do problema. Há dois tipos de condições de contorno, e podem ser visualizadas na Figura 2.1: as mecânicas, representadas pelas forças atuantes na estrutura e as condições de contorno cinemáticas, representadas pelos apoios (pontos do corpo com movimentos restringidos).

2.1.1.4 Problema da mecânica dos sólidos linear elástica com deslocamentos e deformações infinitesimais.

Por fim, através das equações de equilíbrio (2.6), das equações constitutivas (2.7), das relações cinemáticas (2.1), das equações de compatibilidade (2.3) e das condições de contorno, determina-se o vetor deslocamento e, conseqüentemente, as deformações e tensões em todos os pontos do corpo sólido da figura 2.1.

Este sistema está formado por um conjunto de 15 equações (3 de equilíbrio, 6 relações cinemáticas, 6 equações constitutivas) que, junto com as equações de compatibilidade e as condições de contorno, permitem achar o campo de deslocamentos dentro do domínio de um sólido.

No entanto, muitas vezes o problema analisado, mesmo no caso linear, não possibilita uma resolução de forma analítica. Diante disso, faz-se necessária a utilização de algum método numérico, como o método dos elementos finitos, elementos discretos ou elementos de contorno, por exemplo.

2.1.1.5 Tensões principais

Outro conceito importante é o das tensões principais, porque forma a base da teoria de falha dos materiais. Pode-se definir um plano principal em cada ponto do corpo sólido, que ocorre sempre que o vetor tensão alinha-se com a normal n deste plano:

$$t_i = \sigma n_i = \sigma \delta_{ij} n_j \quad (2.12)$$

onde σ é a tensão normal atuando nesse plano.

Isto significa que não há tensão cortante agindo neste plano e a direção n , neste caso, é chamada direção principal. Introduzindo a equação (2.12) na equação (2.5):

$$(\sigma_{ij} - \sigma \delta_{ij}) n_j = 0 \quad (2.13)$$

Como $n_i n_i = 1$, para evitar a solução trivial:

$$|\sigma_{ij} - \sigma \delta_{ij}| = 0 \quad (2.14)$$

Mostrando a Equação (2.14) em forma expandida, tem-se:

$$\begin{vmatrix} \sigma_{11} - \sigma & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} - \sigma & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} - \sigma \end{vmatrix} = 0 \quad (2.15)$$

Que é uma equação cúbica em σ , podendo ser escrita na forma:

$$\sigma^3 - I_1 \sigma^2 + I_2 \sigma - I_3 = 0 \quad (2.16)$$

onde I_1, I_2 e I_3 são escalares independentes do sistema de coordenadas em que as componentes de tensões são expressas. São chamados de invariantes e expressos como:

$$I_1 = \sigma_{ii} \quad (2.17)$$

$$I_2 = \frac{1}{2} (\sigma_{ii} \sigma_{jj} - \sigma_{ij} \sigma_{ji}) \quad (2.18)$$

$$I_3 = \frac{1}{6} \epsilon_{ijk} \epsilon_{pqr} \sigma_{ip} \sigma_{jq} \sigma_{kr} \quad (2.19)$$

Devido à simetria do tensor de tensões, existem três raízes reais (σ_1, σ_2 e σ_3), chamadas de tensões principais da equação (2.14). Uma direção principal esta associada com cada tensão principal satisfazendo a equação (2.13) e $n_i n_i = 1$. As três direções principais associadas com os planos principais são ortogonais entre si. As tensões principais correspondem à máxima, intermediária e mínima tensões normais em um ponto. Além disso, a máxima tensão cortante nesse ponto é igual à metade da diferença entre as tensões principais máxima e mínima e atua em um plano a 45 graus da direção dessas tensões. Na realidade o

cálculo de invariantes de um tensor é uma propriedade matemática deste tipo de estrutura, como é apresentado em forma genérica mais adiante nas expressões (2.43) até (2.45).

2.1.2 Problemas não-lineares na mecânica dos sólidos

Na seção imediatamente anterior foi apresentado um resumo da formulação empregada na solução de problemas na mecânica dos sólidos. Considerando o sólido como um material elástico linear e homogêneo, e os deslocamentos e deformações aplicadas infinitesimais, isto se traduz num sistema de equações lineares, no qual foram realizadas algumas simplificações nas equações para tornar o problema linear, o que realmente pode ser empregado em diversos problemas dessa natureza. Porém, há casos em que as deformações ocorridas são consideradas grandes. Também há situações em que as relações constitutivas são não lineares. Nestes casos, é necessário utilizar os conceitos de não-linearidade na mecânica dos sólidos.

Se a matriz de rigidez ou o vetor de cargas é dependente dos deslocamentos, então um problema estrutural é considerado não-linear. Essas não-linearidades podem ser divididas em dois grandes grupos, a saber: não-linearidade do material e não-linearidade geométrica. A primeira representa alteração das propriedades do material quando deformado. Já a não-linearidade geométrica está associada a mudanças significativas da configuração indeformada para a configuração deformada da estrutura. Em outras palavras, é quando ocorrem grandes deformações, que é parte do objeto deste estudo. Muito material bibliográfico já foi desenvolvido sobre estes tópicos, entre eles podemos citar Truesdell, 1966 e também Malvern, 1969.

Na continuação, faz-se necessário a apresentação sucinta de alguns tópicos deste tema, para o desenvolvimento do presente trabalho.

2.1.2.1 Relação geral entre deformações e deslocamentos

É necessário definir a posição deformada e indeformada do corpo. Referindo-se à figura 2.1 pode-se dizer que

$$X = Xe_1 + Ye_2 + Ze_3 \quad (2.20)$$

e

$$x = xe_1 + ye_2 + ze_3 \quad (2.21)$$

representam os vetores posição da configuração indeformada e deformada, respectivamente, do corpo em estudo. Desta forma o vetor deslocamento de um ponto genérico P do corpo após a deformação do mesmo está representada pelo vetor \mathbf{u} , sendo \mathbf{u} :

$$\mathbf{u} = ue_1 + ve_2 + we_3 = (X - x)e_1 + (Y - y)e_2 + (Z - z)e_3 = \mathbf{X} - \mathbf{x} \quad (2.22)$$

Nas expressões (2.20) até (2.22), \mathbf{e}_1 , \mathbf{e}_2 e \mathbf{e}_3 representam os versores do sistema de coordenadas cartesiano adotado, e u , v , w , X , Y , Z , x , y , e z são as componentes dos vetores \mathbf{u} , \mathbf{X} e \mathbf{x} respectivamente.

2.1.2.2 Definição de uma medida de deformação

Para introduzir o conceito de deformação, parte-se de uma evidência experimental:

“Se são aplicadas forças externas sobre um meio contínuo, aparecem tensões e o corpo experimenta uma transformação de seus pontos materiais, ou seja, aparece um campo de deslocamentos (um valor de $\mathbf{u}(x,y,z)$, ou $\mathbf{u}(X,Y,Z)$ para cada partícula do corpo).

Porém, a inversa não é sempre certa: pode aparecer um campo de deslocamentos sem que tenham surgido tensões interiores. Por exemplo, um movimento de corpo rígido.

Isto implica que: não é possível relacionar em forma direta as tensões com os deslocamentos. É necessária uma grandeza intermediária, a deformação.

Para definir uma métrica da deformação seria desejável que:

- a) As deformações sejam nulas se o corpo se mexe como rígido;
- b) Que os valores da deformação possam organizar-se numa matriz simétrica de três por três para se poder relacioná-las com as tensões;
- c) Que representem extensões e distorções, que são as duas formas principais de deformação.

Para continuar avançando é necessário definir a matriz F como segue

$$F = \frac{\partial x}{\partial X} \quad (2.23)$$

Verifica-se que

$$\det F = \frac{d_{-}vol}{d_{-}VOL} \quad (2.24)$$

Este quociente mede a variação de volume entre uma porção diferencial do corpo deformado e indeformado. Em um movimento de corpo rígido verifica-se que $F=I$.

O objetivo da medida da deformação é avaliar a parte da transformação que não represente movimento de corpo rígido. Ou seja, medir o deslocamento relativo de dois pontos do corpo contínuo, por exemplo, P e Q na configuração indeformada, e P' e Q' na configuração deformada, como se indica na figura 2.3.

Desta forma, se verifica que o quadrado da distância entre os dois pontos, P e Q , será, na configuração indeformada, dada pela seguinte expressão:

$$dS^2 = dX^2 + dY^2 + dZ^2 = dx^T (F^T F)^{-1} dx = dx^T B dx \quad (2.25)$$

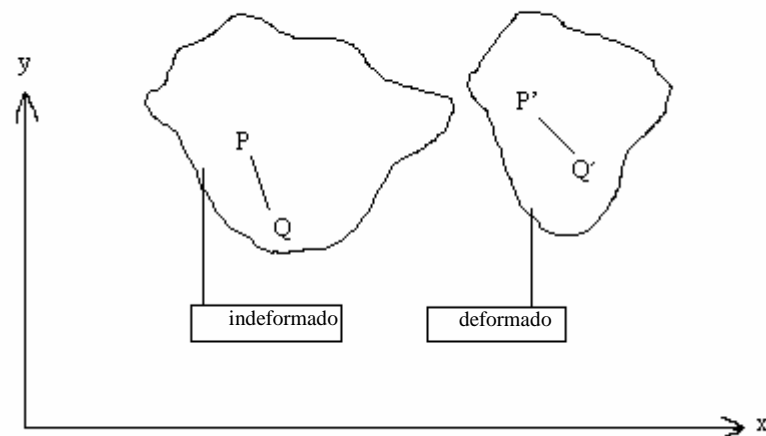


Figura 2.3 – Deslocamento relativo na configuração indeformada e deformada

e, na configuração deformada, por:

$$ds^2 = dx^2 + dy^2 + dz^2 = dX^T (F^T F) dX = dX^T G dX \quad (2.26)$$

onde o sub-índice T indica transposição e o sub-índice -1, inversa. Desta forma, já é possível propor duas formas de medir a deformação:

Com o tensor $\mathbf{B}^{-1} = (\mathbf{F}^T \mathbf{F})^{-1}$, que relaciona o comprimento do segmento PQ na configuração indeformada dS com o comprimento na configuração deformada ds . Este tensor de deformação é chamado de tensor de deformação de *Cauchy Green à esquerda*.

E com tensor $\mathbf{G} = (\mathbf{F}^T \mathbf{F})$, que relaciona o comprimento do segmento $P'Q'$ na configuração deformada ds com o comprimento na configuração indeformada dS . Este tensor de deformação é chamado de tensor de deformação de *Cauchy Green à direita*.

Cabe salientar que tanto \mathbf{B} como \mathbf{G} são simétricos, mas não cumprem a condição de ser nulos no caso de deslocamento de corpo rígido. Neste caso se verifica que $\mathbf{B} = \mathbf{G} = \mathbf{I}$, onde \mathbf{I} representa a matriz identidade.

Para definir outras métricas que verifiquem ser nulas quando tem-se deslocamento de corpo rígido se escreve:

$$ds^2 - dS^2 = dx^T dx - dX^T dX = (dX^T (F^T F) dX - dX^T dX) = dX^T (F^T F - I) dX = dX^T (2E_{GL}) dX \quad (2.27)$$

onde o tensor E_{GL} é chamado de tensor de *Green Lagrange*. Ele representa a variação do comprimento de um segmento com relação a seu comprimento inicial.

Verifica-se também que

$$\mathbf{G} = \mathbf{I} + 2\mathbf{E}_{GL} \quad (2.28)$$

Procedendo de forma análoga, também se pode escrever:

$$ds^2 - dS^2 = dx^T dx - dX^T dX = (dx^T dx - dx^T (F^{-T} F^{-1}) dx) = dx^T (I - F^{-T} F^{-1}) dx = dx^T (2E_A) dx \quad (2.29)$$

onde E_A é outra métrica de deformação, chamada de *tensor de Almansi*.

Neste caso se verifica que

$$B = I + 2E_A \quad (2.30)$$

Poderia ser de utilidade escrever os tensores em termos do vetor de deslocamentos \mathbf{u} em vez de trabalhar em termos dos vetores posição. Neste caso tem-se que a matriz gradiente de deformação \mathbf{F} pode ser expressa da forma:

$$F = \frac{\partial x}{\partial X} = \frac{\partial(X + u)}{\partial X} = I + \frac{\partial u}{\partial X} \quad (2.31)$$

Então utilizando (2.25) e (2.31), o tensor \mathbf{B} fica:

$$B = \left(I + \frac{\partial u}{\partial x}\right) \left(I + \frac{\partial u}{\partial x}\right)^T \quad (2.32)$$

E utilizando (2.26) e (2.31), o tensor \mathbf{G} fica:

$$G = \left(I + \frac{\partial u}{\partial X}\right)^T \left(I + \frac{\partial u}{\partial X}\right) \quad (2.33)$$

E os tensores E_A e E_{GL} em termos do vetor deslocamento ficam:

$$E_A = \frac{1}{2}(B - I) = \frac{1}{2} \left(\left(I + \frac{\partial u}{\partial x}\right) \left(I + \frac{\partial u}{\partial x}\right)^T - I \right) \quad (2.34)$$

$$E_{GL} = \frac{1}{2}(G - I) = \frac{1}{2} \left(\left(I + \frac{\partial u}{\partial X}\right)^T \left(I + \frac{\partial u}{\partial X}\right) - I \right) \quad (2.35)$$

Pode-se usar a notação indicial para representar (2.32), (2.33), (2.34) e (2.35). Esta notação é uma alternativa de representação empregada no desenvolvimento da mecânica do contínuo, onde os índices i, j, k, l podem variar de 1 a 3, índices repetidos indicam soma e o δ_{ij} é um operador chamado de delta de Kroneker, que tem valor 1 se $i=j$ e 0 se $i \neq j$. As regras básicas desta notação podem ser encontradas em Ogden, 1997. Assim, a representação indicial fica:

$$F_{ij} = \frac{\partial x_i}{\partial X_j} \quad (2.36)$$

$$B_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_m}{\partial x_j} \frac{\partial u_m}{\partial x_j} \quad (2.37)$$

$$G_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_m}{\partial X_j} \frac{\partial u_m}{\partial X_j} \quad (2.38)$$

$$E_{A_{ij}} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_m}{\partial x_j} \frac{\partial u_m}{\partial x_j} \right) \quad (2.39)$$

$$E_{GL_{ij}} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_m}{\partial X_j} \frac{\partial u_m}{\partial X_j} \right) \quad (2.40)$$

Desta forma têm-se definidas quatro medidas de deformação, as quais são utilizadas segundo conveniência em diferentes formulações da mecânica do contínuo. Cabe salientar algumas observações sobre elas:

a) No caso de deformações infinitesimais, termos de ordem superior podem ser desconsiderados. Como não se tem como diferenciar entre configuração deformada e indeformada neste caso, tem-se que:

$$B_{ij} = G_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} = 1 + 2\varepsilon_{ij} \quad (2.41)$$

$$E_{A_{ij}} = E_{GL_{ij}} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right) = \varepsilon_{ij} \quad (2.42)$$

onde ε_{ij} é o tensor de Cauchy utilizado, em geral, como medida de deformação no caso de problemas de deformações infinitesimais (equação (2.2)).

b) Todas as medidas de deformação vistas são tensores, o que implica que cumpram com uma série de propriedades matemáticas, entre elas:

b.1) É possível rotar seu sistema de referência até tornar o tensor diagonal. Neste caso, os valores da diagonal principal são os autovalores da matriz que representa o tensor. As direções a que está referido o tensor são definidas pelos autovetores do mesmo, chamadas

estas de direções principais. A este respeito, é possível provar que os tensores \mathbf{B} e \mathbf{E}_A por um lado, assim como \mathbf{G} e \mathbf{E}_{GL} pelo outro, têm as mesmas direções principais.

b.2) É possível definir os tensores pelos seus invariantes, que são escalares que caracterizam o tensor independentemente do sistema de referência ao qual estão expressados. Este conceito é fundamental, pois quando se definem relações constitutivas onde o material é isotrópico (mesmo comportamento em qualquer direção), é possível trabalhar com os invariantes das deformações.

Dado um tensor de segunda ordem genérico A_{ij} , definem-se seus invariantes como segue:

$$I_1 = A_{11} + A_{22} + A_{33} \quad (2.43)$$

$$I_2 = \begin{vmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{vmatrix} + \begin{vmatrix} A_{33} & A_{31} \\ A_{13} & A_{11} \end{vmatrix} + \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \quad (2.44)$$

$$I_3 = |\mathbf{A}| \quad (2.45)$$

c) No caso de uma barra longa submetida a uma extensão uniaxial, como a que se apresenta na figura 2.4, as diversas medidas de deformação, vistas no sentido da extensão, são:

$$F_{11} = \left(\frac{L}{L_0} \right) \quad (2.46)$$

$$B_{11} = \left(\frac{L_0}{L} \right)^2 \quad (2.47)$$

$$G_{11} = \left(\frac{L}{L_0} \right)^2 \quad (2.48)$$

$$E_{A11} = \frac{L - L_0}{L} \quad (2.49)$$

$$E_{GL11} = \frac{L - L_0}{L_0} \quad (2.50)$$

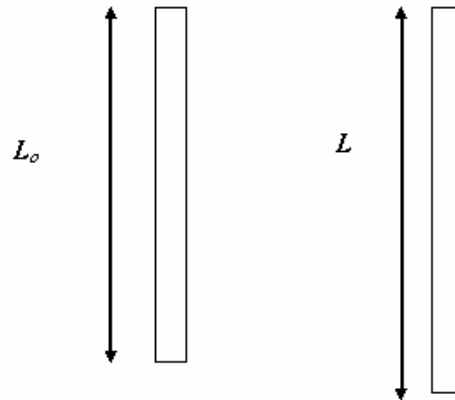


Figura 2.4 – Barra longa sofrendo uma extensão na direção axial

Para deformações infinitesimais, ocorre $\epsilon_{11} = E_{A11} = E_{GL11} = \frac{L-L_0}{L_0}$ e

$$B_{11} = G_{11} = 1 + 2 \frac{L-L_0}{L_0}.$$

2.1.2.3 Medida de tensão

Considera-se um meio material contínuo B ocupando um volume espacial V , como se indica na figura 2.5.

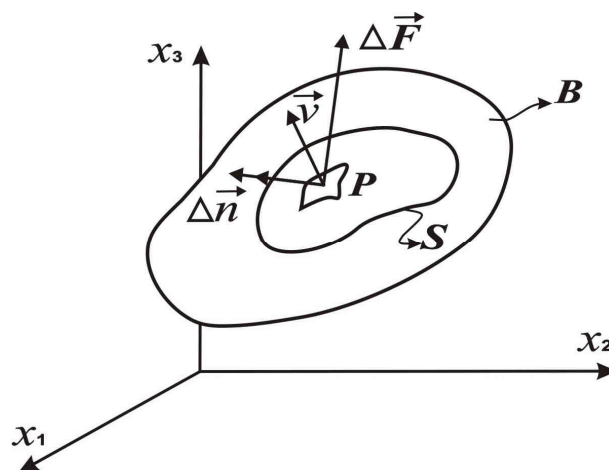


Figura 2.5 – Corpo contínuo de domínio B , sobre o qual se define uma tensão atuante

Imagine-se uma superfície fechada S em B , e deseja-se expressar a interação do meio material externo a S sobre o meio material interno a superfície S .

A interação entre sistemas materiais, do ponto de vista mecânico, pode ser manifestada através de forças de ação à distância, chamadas forças de volume, e de forças de superfície.

Admite-se que toda a interação entre as partes é devida somente a forças de superfície, Através da superfície de separação S , é então postular o princípio das tensões de Euler:

a) Existe o limite

$$\lim_{\Delta S \rightarrow 0} \frac{\Delta \vec{F}}{\Delta S} = \vec{T} \quad (2.51)$$

onde \vec{T} é denominado vetor tensão no ponto P sobre a superfície da normal externa $\vec{\nu}$.

b) Admite-se que

$$\lim_{\Delta S \rightarrow 0} \frac{\Delta \vec{M}}{\Delta S} = 0 \quad (2.52)$$

Os enunciados a e b acima são suficientes para a maioria das aplicações em engenharia.

Para aplicações muito específicas que ocorrem em micro-mecânica, entretanto, não é possível admitir a hipótese b [Kunetsova, 2002; Roderic, 1995]. É o caso de materiais micro polares.

2.1.2.4 As componentes das tensões

Considere-se nas vizinhanças de um ponto P de um meio contínuo, um paralelepípedo elementar com faces paralelas aos planos coordenados, como se indica na figura 2.6.

Fica convencionado que as faces de normal exterior na direção dos eixos positivos se denominam faces positivas e, no caso contrário, negativas. Sobre as fases positivas atuam os

vetores tensão $\frac{\nu_1}{T} \rightarrow$, $\frac{\nu_2}{T} \rightarrow$, $\frac{\nu_3}{T} \rightarrow$.

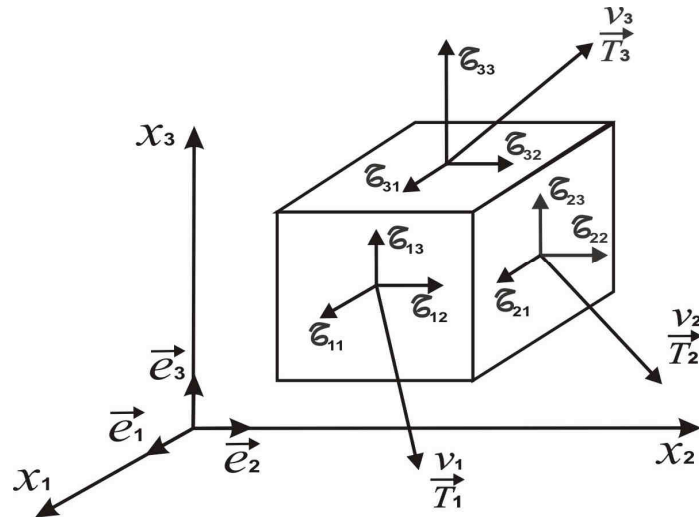


Figura 2.6 – Paralelepípedo elementar que mostra a decomposição da tensão $\frac{v}{T}$

Decompondo os vetores nas suas componentes cartesianas obtém-se:

$$\frac{v_1}{T} \rightarrow = \sigma_{11} \vec{e}_1 + \sigma_{12} \vec{e}_2 + \sigma_{13} \vec{e}_3$$

$$\frac{v_2}{T} \rightarrow = \sigma_{21} \vec{e}_1 + \sigma_{22} \vec{e}_2 + \sigma_{23} \vec{e}_3 \quad (2.53)$$

$$\frac{v_3}{T} \rightarrow = \sigma_{31} \vec{e}_1 + \sigma_{32} \vec{e}_2 + \sigma_{33} \vec{e}_3$$

Ou, em forma matricial:

$$\frac{v}{T}_i = \sigma_{ij} e_j \quad (2.54)$$

As nove quantidades σ_{ij} denominam-se componentes do tensor de tensões no ponto considerado.

Faz-se necessário aqui algumas observações:

a) Dependendo da área sobre a qual a tensão é medida (configuração deformada ou indeformada) e sobre que configuração é medido o vetor força, pode-se definir as diferentes formas para o tensor tensão, que podem ser:

a.1) O tensor de tensões de Cauchy, em que tanto a força $\Delta\mathbf{F}$ como a área ΔA são medidos na configuração deformada. Este tensor é simétrico, fato este que resulta em uma grande vantagem para simplificar as relações constitutivas.

a.2) O tensor de tensões de Piola-Kirchhoff de primeira espécie: neste caso tanto o vetor força $\Delta\mathbf{F}$ e a área ΔA são medidos na configuração indeformada. O problema deste tensor reside no fato dele não ser simétrico.

a.3) O tensor tensões de Piola-Kirchhoff de segunda espécie: o vetor $\Delta\mathbf{F}$ é medido na configuração deformada mas o ΔA é medida na configuração indeformada. Este tensor tem a vantagem de ser simétrico.

b) É possível relacionar os diferentes tensores de tensões como se indica a seguir:

$$\mathbf{T}_1 = \mathbf{PK}^{1esp} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma} \quad (2.55)$$

$$\mathbf{T}_2 = \mathbf{PK}^{2esp} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T} \quad (2.56)$$

Onde:

\mathbf{T}_1 : é o tensor de Piola-Kirchhoff de primeira espécie;

\mathbf{T}_2 : é o tensor de Piola-Kirchhoff de segunda espécie;

$\boldsymbol{\sigma}$: é o tensor de tensões de Cauchy.

c) No caso de pequenas deformações, como a configuração deformada e a indeformada se confundem, a diferença entre os tensores de tensão desaparece.

d) Considerando a viga da figura 2.7, onde se indicam o incremento de força e área nas configurações indeformada (com sub-índice 0) e deformada (sem sub-índice), tem-se que:

$$\boldsymbol{\sigma} = \frac{\Delta\mathbf{F}}{\Delta A}$$

$$\mathbf{T}_1 = \mathbf{PK}^{1esp} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma} = \frac{\Delta F_0}{\Delta A_0}$$

$$\mathbf{T}_2 = \mathbf{PK}^{2esp} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T} = \frac{\Delta F}{\Delta A}$$

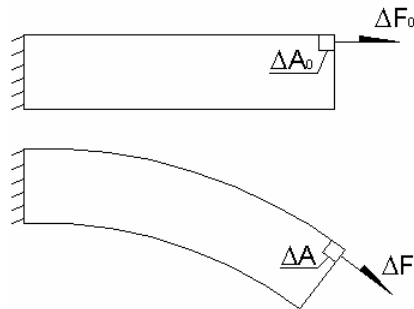


Figura 2.7 – Esquema que facilita entender a definição dos tensores de tensões

2.1.2.5 Relações constitutivas para materiais elásticos

O que caracteriza um material elástico, fisicamente, é que se aplicamos uma carga sobre ele, ele se deforma, ao passo que, se retiramos a carga, ele volta a sua posição original. E isto se faz seguindo sempre o mesmo caminho no domínio das tensões e deformações. Isto se indica em forma esquemática na figura 2.8.

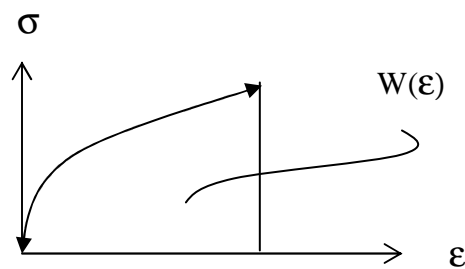


Figura 2.8 – Característica de um corpo elástico

Para que a propriedade antes indicada se cumpra é necessário verificar que as tensões derivem da densidade de energia de deformação $W(\epsilon)$, que é a energia de deformação acumulada no corpo por unidade de volume.

É possível escrever então, em termos do tensor de tensões de Cauchy e do tensor de deformações de Green-Lagrange à direita, que:

$$\sigma_{ij} = \frac{\partial(W(G_{ij}))}{\partial(G_{ij})} \quad (2.57)$$

Se o material estudado é isótropo, pode-se dizer que W é função só dos invariantes do tensor de deformações, independente do sistema de coordenadas ao qual o tensor de deformações está referido. Então seria possível escrever:

$$\sigma_{ij} = \frac{\partial(W((I_{1G} - 3), (I_{2G} - 3), (I_{3G} - 1)))}{\partial(G_{ij})} \quad (2.58)$$

Aplicando (2.43), (2.44) e (2.46), escreve-se os invariantes em termos das deformações principais do tensor de Green-Lagrange à direita, $\lambda_i^2 = (\frac{l_i}{L_i})^2$, da seguinte forma:

$$I_{G1} = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \quad (2.59)$$

$$I_{G2} = \lambda_1^2 \lambda_2^2 + \lambda_2^2 \lambda_3^2 + \lambda_3^2 \lambda_1^2 \quad (2.60)$$

$$I_{G3} = \lambda_1^2 \lambda_2^2 \lambda_3^2 \quad (2.61)$$

A equação (2.58) utiliza as expressões $(I_{1G}-3)$, $(I_{2G}-3)$, e $(I_{3G}-1)$ para fazer com que o termo entre parêntesis seja zero quando há deslocamento de corpo rígido. Cabe lembrar que neste caso $\lambda_i^2=1$.

Também neste ponto são necessárias algumas observações:

a) No caso de materiais elastoméricos, verifica-se que o material é praticamente incompressível. Neste caso a expressão (2.58) pode ser escrita da forma:

$$\sigma_{ij} = \frac{\partial(W((I_{1G} - 3), (I_{2G} - 3)))}{\partial(G_{ij})} \quad (2.62)$$

b) Para elastômeros, as leis constitutivas mais conhecidas podem ser detalhadas a seguir, segundo Ogden, 1997, como uma série de potências:

$$W(I_{B1}, I_{B2}) = \sum_{p,q=0}^{\infty} C_{pq} (I_{B1} - 3)^p (I_{B2} - 3)^q \quad (2.63)$$

Onde C_{pq} são constantes, que dependeram do material, e são obtidas através de ensaios sobre corpos de prova de geometrias simples.

Uma das leis mais simples propostas para materiais elastoméricos é a de Monney-Rivlin, considerando só dois parâmetros:

$$W(I_{G1}, I_{G2}) = C_1(I_{G1} - 3) + C_2(I_{G2} - 3) \quad (2.64a)$$

No caso de deformações infinitesimais, verifica-se que $E=6(C1+C2)$ e $G=2(C1+C2)$, sendo E e G os módulos de elasticidade longitudinal e transversal, respectivamente.

Outra lei conhecida é a de Yeoh, um modelo fenomenológico baseado no primeiro invariante de deformação, que tem a forma

$$W = \sum_{i=1}^N C_{i0} (I_{G1} - 3)^i \quad (2.64b)$$

onde N é o número de termos.

c) Dentro dos desenvolvimentos realizados neste campo no GMAp (Grupo de Mecânica Aplicada da UFRGS), pode-se citar as dissertações de Hoss, 2009 e Stumpf, 2009, as quais avaliam várias leis constitutivas hiperelásticas propondo um novo modelo hiperelástico e uma nova metodologia de ajuste.

d) Temizer e Zohdi, 2007, apresentam uma forma de descrever as tensões existentes em um material elástico isotrópico não-linear através do segundo tensor tensão de Piola-Kirchhoff, que se apresenta a seguir:

$$T_2 = \alpha_0 \cdot I + \alpha_1 \cdot E_L + \alpha_2 \cdot E_L^2 \quad (2.65)$$

onde T_2 é o segundo tensor de Piola Kirchhoff, E_L é o tensor deformação de Lagrange, I é a matriz identidade e α_i são funções dos invariantes de Lagrange I_{E1} , I_{E2} e I_{E3} . Estes invariantes são determinados univocamente através dos estiramentos principais λ_F^i do tensor

de deformação F , definidos já na equação (2.23), que representam as deformações nas três direções principais conforme figura 2.9. Assim:

$$I_{E_1} = \lambda_{E_1} + \lambda_{E_2} + \lambda_{E_3} \quad (2.66)$$

$$I_{E_2} = \lambda_{E_1} \cdot \lambda_{E_2} + \lambda_{E_2} \cdot \lambda_{E_3} + \lambda_{E_1} \cdot \lambda_{E_3} \quad (2.67)$$

$$I_{E_3} = \lambda_{E_1} \cdot \lambda_{E_2} \cdot \lambda_{E_3} \quad (2.68)$$

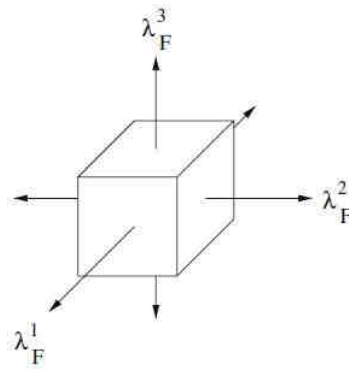


Figura 2.9 – Representação das direções principais de deformação

Onde os autovalores do tensor de deformações de Lagrange podem ser escritos em termos dos autovalores do tensor F como segue:

$$\lambda_E^1 = \frac{1}{2} \cdot \left((\lambda_F^1)^2 - 1 \right), \quad \lambda_E^2 = \frac{1}{2} \cdot \left((\lambda_F^2)^2 - 1 \right), \quad \lambda_E^3 = \frac{1}{2} \cdot \left((\lambda_F^3)^2 - 1 \right) \quad (2.69)$$

2.2 Aspectos básicos da Mecânica da Fratura

O estado de tensões de qualquer ponto de uma estrutura sob um sistema de cargas atuantes, está caracterizado por um tensor de tensões que contém informação da tensão em todas as direções. Através da utilização de alguma das teorias de resistência clássicas, pode-se obter, a partir do tensor de tensões, um único valor de comparação, chamado de tensão de cálculo (σ_{calc}). Utilizando a teoria de resistência de von Mises, por exemplo, a tensão de cálculo será igual à tensão equivalente de von Mises σ_{vm} . Na prática do projeto convencional,

o valor de tensão calculado (σ_{vm} , por exemplo) é comparado com uma tensão de referência segundo a relação estabelecida pela equação (2.70).

$$\sigma_{calc.}(F, Geometria) = \sigma_{material}(material, T^*, \dot{F}) \quad (2.70)$$

A tensão de referência diz respeito à resistência mecânica do material de modo que, para o projeto de componentes mecânicos considerando o término do comportamento elástico, se utiliza a tensão de escoamento obtida de um ensaio de tração uniaxial.

A tensão calculada é função da geometria e condições de contorno aplicadas (F), e a tensão de referência vai depender do material e das condições em que foi realizado o ensaio, temperatura (T^*) e velocidade da carga aplicada (\dot{F}). Neste caso se supõe a estrutura livre de defeitos e discontinuidades e não se considera o efeito dos mesmos.

Seja agora a placa quadrada com um orifício elíptico central, tracionada pelos extremos superior e inferior como mostrado na figura 2.10.

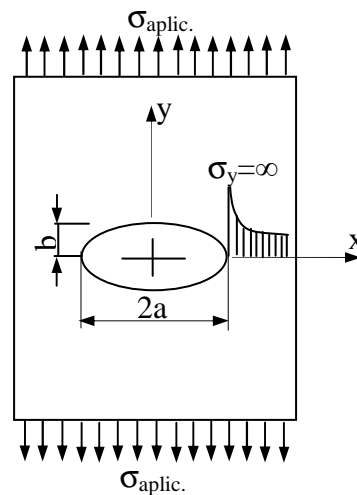


Figura 2.10 – Caso considerado por Inglis para sua dedução [Kanninen e Popelar, 1985]

A distribuição de tensões no componente é influenciada pela presença do orifício. Fazendo uso da teoria da elasticidade, Inglis, 1913 [apud Anderson, 1995], para este caso, propôs que a variação da tensão na direção da tração aplicada (σ_{aplic}) sob o eixo x é calculada pela expressão (2.71). Nela se pode observar que quando o semi-eixo menor da elipse tender a zero ($b \rightarrow 0$), a tensão no extremo do semi-eixo a , $\sigma_y(a)$ vai ser infinita, ou seja, vai resultar

numa singularidade. Pode-se estabelecer uma semelhança entre estas elipses de semi-eixo menor tendendo a zero com os defeitos e falhas que se apresentam no material em micro, meso, ou macro escala [Kanninen e Popelar, 1985; Knott, 1993].

$$\sigma(a) = \sigma \left(1 + 2 \left(\frac{a}{b} \right)^{\frac{1}{2}} \right) \quad (2.71)$$

Na medida em que vão sendo desenvolvidos os métodos de detecção das trincas, tem-se comprovado que elas estão sempre presentes em qualquer estrutura. Elas podem ser inerentes ao material ou ao processo de fabricação, ou então podem ter surgido durante o desempenho em serviço da estrutura.

A equação (2.71) sugere que a tensão calculada na ponta de uma trinca teórica ($a/b \rightarrow \infty$) será infinita independentemente do valor da carga remota aplicada. Isso faz com que nunca seja satisfeita a relação estabelecida pela equação (2.70), na qual se baseia o projeto convencional. No entanto, sabe-se que as estruturas reais, ainda na presença de defeitos, resistem com segurança a valores de cargas finitos. Isso indica a necessidade de procurar critérios de projeto aplicáveis a estruturas com defeitos macroscópicos.

Seguindo o mesmo princípio da equação (2.70), a verificação de projeto considerando os critérios da mecânica da fratura está representada pela equação (2.72). Nela, para garantir a integridade da estrutura trincada, o parâmetro calculado $X_{cal.}$ deve ser menor que $X_{mat.}$. Neste caso o parâmetro calculado $X_{cal.}$ é função do tamanho de trinca (a), geometria da estrutura e condições de contorno (F). Por outro lado, o parâmetro que caracteriza o material $X_{mat.}$ é função da temperatura (T), velocidade de aplicação da carga (\dot{F}), e espessura analisada (B) [Kanninen e Popelar, 1985].

$$X_{cal.}(F, Geometria, a) = X_{mat.}(B, T^*, \dot{F}) \quad (2.72)$$

Em ambas as equações, (2.70) e (2.72), a variável correspondente ao material não é função da geometria do problema, e pode ser obtida a partir de um ensaio num corpo de prova simples, para o caso de σ_{mat} , ou de um ensaio em corpos de prova compacto de tração ou de flexão em três pontos, para o caso de $X_{mat.}$, conforme os procedimentos descritos em normas. Como exemplo deste tipo de normas, pode-se citar a E399-90, 1997. Segundo as formulações utilizadas para a obtenção do parâmetro $X_{cal.}$, a mecânica da fratura pode ser dividida em linear

elástica ou não-linear. Como parâmetros da mecânica da fratura linear elástica podem-se mencionar:

a) A energia específica de fratura (G) ou taxa de libertação de energia. Foi primeiramente proposta por Griffith, 1920 e é considerado um parâmetro global por envolver o balanço energético da estrutura como um todo;

b) O fator de intensidade de tensão K [Irwin, 1957] é considerado um parâmetro local, pois considera somente a distribuição de tensões no entorno da ponta da trinca.

No caso da mecânica da fratura não-linear, podem ser mencionados a integral J [Rice, 1968], considerada como parâmetro global que envolve também o balanço energético da estrutura em análise, e o Crack Opening Displacement (COD), parâmetro introduzido por Wells, 1961.

A seguir são discutidos os fundamentos teóricos correspondentes aos parâmetros da mecânica da fratura linear elástica. Informações mais detalhadas sobre aos parâmetros da mecânica da fratura não linear podem ser encontradas em Anderson, 1995 e Kanninen e Popelar, 1985.

2.2.1 Critério Energético de Griffith

Os primeiros estudos dentro da mecânica da fratura foram dados por Griffith, 1920. Intrigado com a divergência existente entre a resistência teórica e a resistência real da maioria dos sólidos cristalinos, Griffith realizou uma série de ensaios com fibras de vidro de diferentes espessuras obtendo os resultados ilustrados na figura 2.11.

Observa-se que quanto menor a espessura utilizada, a resistência obtida aproxima-se da resistência teórica do vidro, enquanto que para espessuras maiores os valores de resistência apresentados aproximam-se com a resistência convencional do vidro.

Tais resultados levaram Griffith a concluir que o motivo da discrepância entre a resistência real e a teórica dos sólidos cristalinos era a presença de defeitos ou trincas nos mesmos. Quanto maior o tamanho, maior a probabilidade de encontrar defeitos, o que poderia ser interpretado a nível global como a probabilidade da existência de mais e maiores defeitos.

Com o propósito de fundamentar a hipótese anterior, Griffith propõe o balanço energético representado pela equação (2.73). Ela estabelece que a condição crítica para a propagação da trinca é que a variação da energia potencial experimentada pelo sistema seja igual à energia necessária para formar uma superfície (fratura ou trinca).

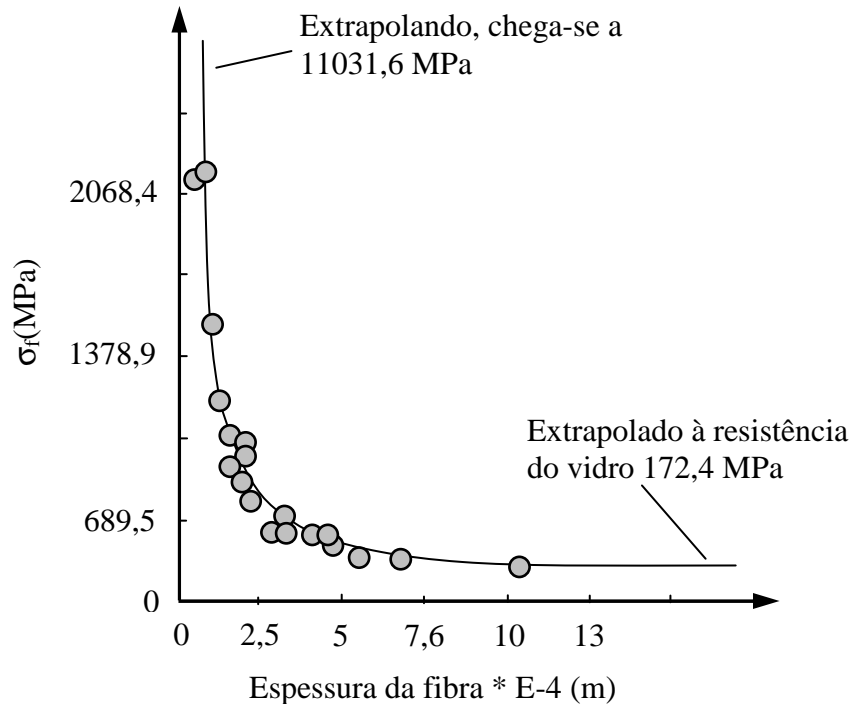


Figura 2.11 – Experiência realizada por Griffith com as fibras de vidro [Kanninen e Popelar, 1985]

Esta última, considerada por unidade de área gerada, é uma propriedade do material. Considera-se para a análise que as forças de inércia, produto da aplicação das cargas, sejam desprezíveis.

$$\frac{dW}{dA} - \frac{dU}{dA} = \gamma \quad (2.73)$$

Onde:

W : Potencial das forças externas aplicadas sobre o corpo;

U : Energia de deformação interna;

γ : Energia de superfície do material;

A : Área de superfície da trinca formada. Para uma placa de espessura B com uma trinca interna de comprimento $2a$ resulta $A = 4Ba$.

Para esta análise, Griffith se baseou no problema apresentado na figura 2.12, o qual consiste numa placa de espessura B e dimensões de lado muito maiores em relação a dimensão da trinca central. Nos extremos da placa é aplicada uma tensão uniforme e o sistema de coordenadas está localizado no centro da trinca. Pelas características do caso tomado como referência para a análise, o mesmo pode ser considerado em estado plano de tensões.

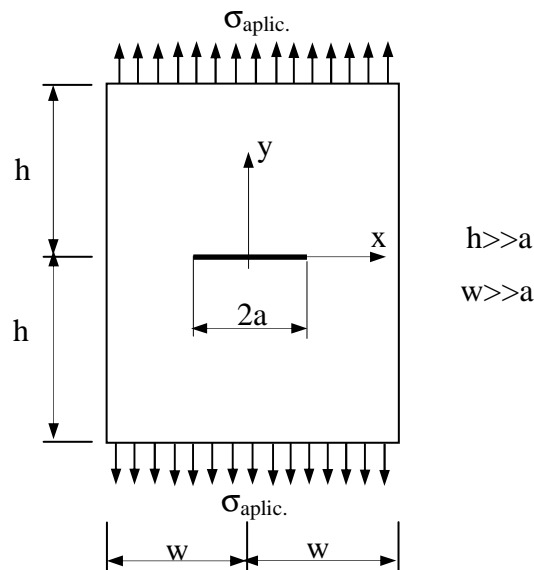


Figura 2.12 – Placa de dimensões infinitas utilizada pelo Griffith

Considerando o teorema de Clapeyron, o potencial do trabalho externo (W) pode ser considerado como duas vezes a energia de deformação elástica. Com isso a equação (2.73) pode ser escrita como mostra a equação (2.74) [Kanninen e Popelar, 1985].

$$\frac{dW}{dA} - \frac{dU}{dA} = \frac{2U}{dA} - \frac{dU}{dA} = \frac{dU}{dA} = \gamma \quad (2.74)$$

Por outro lado, utilizando a solução de Westergaard para o caso analisado, Griffith obtém que a abertura da face da trinca é calculada por:

$$v = \frac{2\sigma}{E} (a^2 - x^2)^{\frac{1}{2}} \quad (2.75)$$

Onde:

v : deslocamento na direção y da face da trinca;

σ : Tensão remota aplicada;

a : comprimento da trinca;

E : Módulo de elasticidade longitudinal.

Das considerações anteriores, é possível escrever:

$$W - U = 2U - U = 4B \int_0^a \frac{1}{2} \sigma \cdot v(x) dx = \frac{\pi a^2 \sigma^2 B}{E} \quad (2.76)$$

Aplicando a equação (2.74), e considerando $dA=4B \cdot da$, obtém-se:

$$\frac{d(W-U)}{dA} = \frac{a\pi\sigma^2}{2E} = \gamma \quad (2.77)$$

De onde se deduz:

$$\sigma_f = \sqrt{\frac{\gamma 2E}{\pi a}} \quad (2.78)$$

Onde:

σ_f = é a tensão aplicada que provoca a ruptura da placa (Figura 2.12).

A equação (2.78) justifica a hipótese de Griffith, de que quanto maior o tamanho da trinca a tensão de fratura é menor. Para o estado plano de deformações a solução para σ_f é obtida substituindo E por $E/(1-\nu^2)$.

2.2.2 Teoria de Griffith Modificada

O fato da teoria de Griffith ser proposta para materiais idealmente frágeis, no caso o vidro, a torna restrita quanto à aplicação. Orowan, 1950 [*apud* Kanninen e Popelar, 1985], propôs uma solução para esta limitação, em que admite que a energia liberada na propagação da fissura é consumida não só em formação de superfícies de fratura, mas também na deformação plástica associada ao processo de fratura.

Diante disso, a equação (2.78) é modificada resultando em:

$$\sigma_f = \sqrt{\frac{(2\gamma + \gamma_p)E}{\pi a}} = \sqrt{\frac{G_c E}{\pi a}} \quad (2.79)$$

Sendo:

γ_p = energia de deformação plástica associada ao processo de fratura.

A equação (2.74) também é modificada, assumindo a forma:

$$G_c = \frac{dW}{dA} - \frac{dU}{dA} \quad (2.80)$$

Onde:

$G_c = (2\gamma + \gamma_p)$ é a taxa de liberação de energia crítica, e recebeu o símbolo de G_c em homenagem a Griffith.

A equação (2.79) é conhecida como teoria de Griffith modificada.

Orowan e Irwin, 1957, estabeleceram que, desde que as teorias da elasticidade linear possam ser utilizadas, a resistência à fratura de um material está dada pelo valor crítico de G , (G_c), e o G pode ser calculado ou medido experimentalmente utilizando a relação mostrada pela equação (2.81) a seguir:

$$G = \frac{dW}{da} - \frac{dU}{da} = \frac{d(W-U)}{da} \quad (2.81)$$

2.2.3 Fator de Intensidade de Tensões

O fator de intensidade de tensão K é mais um parâmetro fractomecânico que determina as condições de propagação da trinca em função da intensidade das tensões na ponta da mesma, sendo considerado um parâmetro local. Foi introduzido por Irwin, 1957 e está baseado na solução elástica de Westergaard na vizinhança de uma fissura [Kanninen e Popelar, 1985; Moura Branco, 1986]. Irwin estabeleceu três modos básicos de fratura, (I, II e III), os quais são ilustrados na figura 2.13. De maneira geral o modo que mais se apresenta na prática e para o qual o K é menor é o modo I. Toda a formulação mostrada a seguir é deduzida considerando este modo de fratura.

A descrição da obtenção do fator de intensidade de tensão será realizada considerando um sistema de coordenadas polares com a origem na ponta da trinca como mostra a figura 2.14.

Irwin estabeleceu que, a distribuição da tensão na ponta da trinca para o modo I está determinada pela equação:

$$\sigma_{ij} = \frac{K_I}{\sqrt{2r}} \cdot f(\theta) + \sum_{n=1}^{\infty} \frac{\varphi_n}{\sqrt{2}} \cdot r^{\frac{(n-1)}{2}} \cdot f(\theta) \quad (2.82)$$

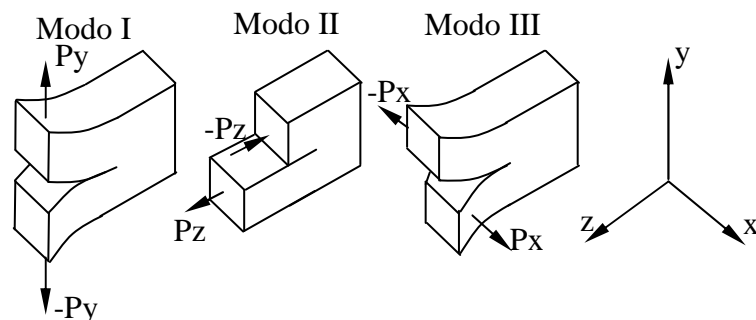


Figura 2.13 – Modos de fratura considerados por Irwin

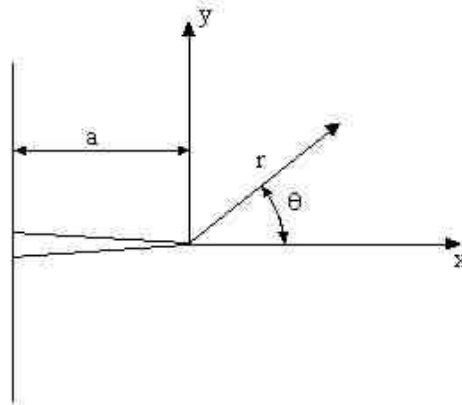


Figura 2.14 – Sistema de coordenadas utilizado na dedução do fator de intensidade de tensão (K_I)

Pode-se observar que, a partir do segundo termo, o raio está elevado ao quadrado, o que indica que para valores de r pequenos (próximos da trinca), o primeiro termo da expansão em série da equação (2.82) predomina sob os outros termos, e os mesmos podem ser desconsiderados. Já para valores de r distantes da ponta da trinca, os outros termos começam ter maior significância e a tensão começa aproximar-se da tensão remota aplicada. Nesta equação (2.82) observa-se que a função $f(\theta)$ depende do ângulo e K_I funciona como um fator de escala.

Considerando a afirmação anterior, nas proximidades da trinca a equação (2.82) pode ser escrita da forma:

$$\sigma_{ij} = \frac{K_I}{\sqrt{2r}} f(\theta) \quad (2.83)$$

Com relação à equação (2.83) são pertinentes algumas observações:

Para $\theta=0$, a função $f(\theta)=1$ e a equação (2.83) pode ser escrita como $\sigma_{ij} = \frac{K_I}{\sqrt{2r}}$.

Analisando a equação (2.83) pode-se concluir que a distribuição de tensões nas proximidades da ponta da trinca independe das condições de contorno e da geometria da estrutura analisada. Isso permite comparar estados de tensão nas proximidades de uma trinca em estruturas de geometria diferentes e sob condições de contorno também diferentes. Este aspecto é de grande importância, uma vez que o estudo de propagação de trincas em corpos de provas realizados em laboratório permite a obtenção do valor de K_I crítico (K_{IC}) que caracteriza o material (o X_{mat} da equação 2.72). E o mesmo pode ser utilizado como referência

para comparar com o K_{Icalc} (o X_{calc} . da equação 2.72) calculado para uma trinca presente numa estrutura real deste mesmo material.

Se $K_{Icalc} > K_{IC}$, então a trinca da estrutura real vai estar numa situação crítica. Na figura 2.15 se mostra de forma esquemática esta analogia.

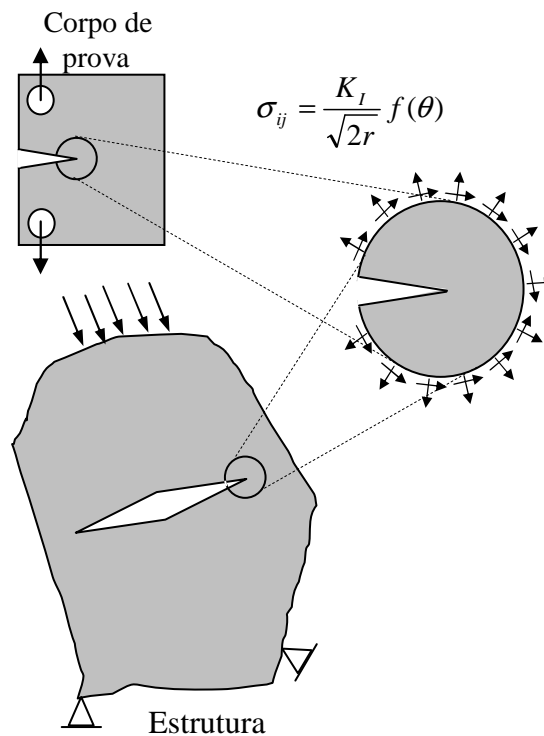


Figura 2.15 – Figura esquemática mostrando a equivalência entre o campo de tensões na ponta da trinca entre uma estrutura real e um corpo de prova [Anderson, 1995]

O fator de intensidade de tensões K_I pode ser determinado de várias formas, tanto numérica quanto analiticamente. Uma ótima revisão pode ser encontrada no trabalho de Aliabadi e Rooke, 1991. Em Soares, 2003, pode-se verificar o cálculo dos fatores de intensidade de tensão de uma trinca elíptica-hiperbólica homofocal através de método analítico. No entanto, na maior parte das vezes é possível escrever o K_I , como propôs Irwin, 1957, na relação apresentada pela equação:

$$K_I = \varphi \cdot \sigma \cdot \sqrt{\pi a} \quad (2.84)$$

Onde:

σ : Tensão existente na estrutura no local da trinca sem considerar a presença da mesma;

φ : Coeficiente dependente da geometria do problema analisado;

a : Comprimento da trinca analisada.

Uma relação entre o G e o K_I é possível de ser estabelecida. A mesma é muito importante, uma vez que através dela se evidencia a relação entre aspectos mecânicos e geométricos (K_I), com aspectos relacionados às propriedades do material (G_c) implícitos nos problemas de mecânica da fratura. A equação (2.85) representa a relação entre estas duas variáveis fractomecânicas quando se tem somente o modo I de fratura (conforme figura 2.13).

$$K_I = \sqrt{G \cdot E'} \quad (2.85)$$

Sendo:

$E' = E$ para estado plano de tensões (EPT);

$E' = E/(1-\nu^2)$ para estado plano de deformações (EPD).

A dedução da relação entre G e K_I pode ser encontrada em qualquer livro básico de mecânica da fratura, como por exemplo, Moura Branco, 1986. Apresenta-se de forma resumida no quadro da figura 2.16, a seguir:

Teoria de Griffith	Griffith modificada
$\sigma_f = \sqrt{\frac{\gamma 2E}{\pi a}}$	$\sigma_f = \sqrt{\frac{E(2\gamma + \gamma_p)}{\pi a}} = \sqrt{\frac{EG_c}{\pi a}} \implies \sigma_f^2 \pi a = EG_c \implies \sigma_f^2 = \frac{EG_c}{\pi a}$
Irwin	
$K_I = \varphi \cdot \sigma \cdot \sqrt{\pi a} \implies \sigma = \frac{K_I}{\varphi \sqrt{\pi a}} \implies \sigma^2 = \frac{K_I^2}{\varphi^2 \pi a}$	
Considerando que $\varphi=1$ e $\sigma=\sigma_f$, obtém-se a relação representada pela Eq. (2.16)	
$\frac{K_I^2}{\varphi^2 \pi a} = \frac{EG_c}{\pi a} \implies K_I^2 = EG_c$	

Figura 2.16 – Quadro ilustrativo da relação entre G e K_I

Como comentado anteriormente, a tensão na ponta da trinca tem um valor finito devido ao efeito da plastificação. Entretanto, surgem tensões acima da tensão de escoamento do material, o que pressupõe um comportamento local não linear que deixa de ser compatível com a elasticidade linear sobre a qual está baseada a formulação do fator de intensidade de tensões e o G_c . Se a região plastificada for suficientemente pequena em relação à região onde é válida a equação (2.83), então pode ser aplicada a mecânica linear elástica da fratura. Caso

contrário, devem ser utilizados outros parâmetros fractomecânicos, como a Integral J e o COD (*Crack Opening Displacement*), os quais não serão abordados aqui.

A relação entre os parâmetros apresentados da mecânica da fratura linear elástica pode ser utilizada para estabelecer critérios de falha de elementos estruturais.

2.3 Materiais Compostos e Tratamento Computacional de Materiais Micro-Heterogêneos.

A grande diversidade de combinações de propriedades físicas, mecânicas e tecnológicas que pode ser obtida com o emprego de materiais compostos os tornaram cada vez mais utilizados em diferentes áreas da engenharia. Em contrapartida, essa crescente aplicação vem exigindo uma rigorosa investigação, sobretudo no aspecto das propriedades mecânicas, estabelecendo assim limites na aplicação ou exploração desses materiais.

Historicamente, a caracterização da resposta mecânica de um material esteve sustentada na base fenomenológica, com a realização de numerosos testes experimentais, o que, por vezes, não permitem uma caracterização mecânica completa do mesmo. Sabe-se, por exemplo, que o comportamento macroscópico de um material está determinado por fenômenos que acontecem em escalas micro estruturais, nas quais se faz extremamente difícil a realização de uma avaliação experimental adequada.

É nesse contexto que a experimentação computacional assume uma relevante importância, uma vez que se apresenta como uma ferramenta muito útil, permitindo a análise mecânica de geometrias complexas, bem como a simulação de ensaios sob as mais diversas condições. Tais resultados numéricos em conjunto com experimentos macroscópicos complementares, contribuem para a caracterização completa da resposta mecânica do material.

Nesta seção será apresentada a definição e classificação dos materiais compostos e também aspectos relacionados ao tratamento numérico computacional dos mesmos.

2.3.1 Materiais Compostos

Sucintamente, materiais compostos são aqueles que apresentam heterogeneidades em escalas estruturais maiores que a escala atômica, mas podem ser tratados como homogêneos na macro-escala ou em alguma escala estrutural intermediária (mesoescala) [Milton, 2002]. Essa definição sugere que a maior parte dos materiais utilizados na engenharia podem ser

considerados como materiais compostos, pois a maioria deles apresentam heterogeneidades na mesoescala. A partir desse momento, o termo mesoescala será utilizado para referir-se a escalas estruturais abaixo da escala macroscópica.

Os materiais compostos são constituídos por duas ou mais fases. Sua combinação dá origem a um agregado de propriedades equivalentes diferente das propriedades dos constituintes. Essas propriedades, bem como a forma e a distribuição espacial dos constituintes podem ser selecionadas visando à obtenção de uma combinação ótima de propriedades mecânicas, físicas e tecnológicas, ou seja, um material sob medida para determinado emprego.

2.3.1.1 Elemento de Volume Representativo (EVR)

As propriedades efetivas ou homogeneizadas utilizadas na análise computacional de estruturas cujo material é de micro-estrutura heterogênea, são obtidas, relacionando os valores médios de variáveis de campo dentro de um elemento de volume representativo da micro-estrutura. Quando se tratam de micro-estruturas heterogêneas periódicas, o EVR está constituído pela menor célula unitária que a caracteriza. Caso o material analisado possua micro-estrutura heterogênea de distribuição aleatória, o EVR deve conter suficiente informação dos constituintes, visando garantir a homogeneidade estatística do mesmo [Ostoja-Starzewski, 2006].

2.3.1.2 Homogeneização

A homogeneização é uma ferramenta matemática que, aplicada à ciência dos materiais, consiste em determinar as propriedades mecânicas equivalentes ou efetivas de materiais micro heterogêneos. Por vezes estas propriedades equivalentes também são chamadas de propriedades homogeneizadas. No texto que segue serão utilizadas estas terminologias indistintamente.

Usualmente, a homogeneização é realizada relacionando-se os valores médios de variáveis de campo dentro de um EVR do material micro heterogêneo considerado. A mesma tem sido amplamente utilizada no tratamento computacional de materiais micro-heterogêneos, tanto em estudos relacionados com a criação de novos materiais, como na caracterização mecânica dos já existentes.

2.3.1.3 Solução de Voigt e Reuss

As primeiras tentativas na obtenção de propriedades mecânicas efetivas são atribuídas a Voigt e Reuss. O primeiro, considerou um campo uniforme de deformações dentro de uma amostra de material micro-heterogêneo, conforme se observa na figura 2.17. Assim a tensão média dentro do EVR poderia ser determinada pela equação (2.86). Mais detalhes podem ser obtidos adiante no item 2.3.1.5.

$$\langle \sigma \rangle = \frac{1}{V_{EVR}} \left(\sum_{i=1}^n \int_{V_n} \sigma_n dV \right) = \frac{V_{EVR_1}}{V_{EVR}} (E_1 : \varepsilon_1) + \dots + \frac{V_{EVR_n}}{V_{EVR}} (E_n : \varepsilon_n) \quad (2.86)$$

Onde:

$\langle \sigma \rangle$: é a tensão média dentro do EVR;

i : indica o número de fases em que o material é composto;

“:” : indica produto interno duplo;

V_{EVR} e V_{EVRn} : são o volume total do EVR e o volume correspondente a uma determinada fase dentro do EVR, respectivamente.

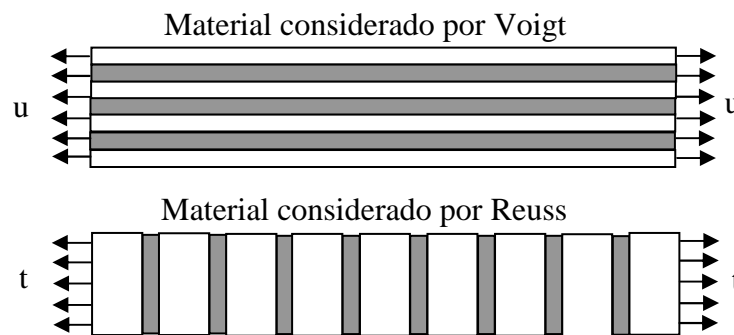


Figura 2.17 – Representação esquemática das considerações realizadas por Voigt e Reuss para estabelecer os limites das propriedades equivalentes de compostos, utilizando a análise numa dimensão [Zohdi e Wriggers, 2005]

Considerando que as deformações são uniformes, a equação (2.86) poderia ser escrita como mostra a equação (2.87), a qual indica que a propriedade equivalente (E^*), pode ser obtida a partir das propriedades das fases constituintes, multiplicada pela sua respectiva fração de volume, na amostra analisada.

$$\langle \boldsymbol{\sigma} \rangle = \frac{V_{EVRI}}{V_{RVE}} (\mathbf{E}_1 : \boldsymbol{\varepsilon}_1) + \dots + \frac{V_{EVRn}}{V_{RVE}} (\mathbf{E}_n : \boldsymbol{\varepsilon}_n) = \underbrace{\left(\frac{V_{EVRI}}{V_{RVE}} \mathbf{E}_1 + \dots + \frac{V_{EVRn}}{V_{RVE}} \mathbf{E}_n \right)}_{\mathbf{E}^*} : \boldsymbol{\varepsilon} \quad (2.87)$$

Reuss complementa o trabalho do Voigt e realiza uma análise similar considerando um campo de tensões uniformes dentro da amostra de material, como também mostra a figura 2.17. Desta forma, são obtidas as propriedades equivalentes dos micro-constituintes (\mathbf{E}^{*-1}) considerando a relação inversa, equação (2.88). Esta metodologia seguida por Voigt e Reuss, é conhecida como regra das misturas e, apesar de não ser a mais adequada, constitui uma maneira rápida de obter os valores extremos esperados das propriedades homogeneizadas ou efetivas para qualquer mistura de fases. Voigt, equação (2.87) representa o limite superior e Reuss, equação (2.88) o limite inferior das propriedades homogeneizadas [Zohdi, 2002].

$$\langle \boldsymbol{\varepsilon} \rangle = \frac{V_{EVRI}}{V_{RVE}} (\mathbf{E}_1^{-1} : \boldsymbol{\sigma}_1) + \dots + \frac{V_{EVRn}}{V_{RVE}} (\mathbf{E}_n^{-1} : \boldsymbol{\sigma}_n) = \underbrace{\left(\frac{V_{EVRI}}{V_{RVE}} \mathbf{E}_1^{-1} + \dots + \frac{V_{EVRn}}{V_{RVE}} \mathbf{E}_n^{-1} \right)}_{\mathbf{E}^{*-1}} : \boldsymbol{\sigma} \quad (2.88)$$

Outras aproximações para homogeneização de materiais micro-heterogêneos particulados têm sido propostas [Zohdi, 2002; Hollister e Kikuchi, 1997], dentre elas:

2.3.1.4 Solução de Eshelby

Uma das mais importantes soluções analíticas da micro-mecânica foi apresentada por Eshelby, 1957. Ele considerou um domínio com uma inclusão elipsoidal Ω_2 submetida a uma autodeformação constante e apresentou uma solução para o campo de deformações, da forma:

$$\boldsymbol{\varepsilon}_{ij} = S_{ijkl} \boldsymbol{\varepsilon}_{kl}^t = \text{constante} \quad \text{em } \Omega_2 \quad (2.89)$$

onde:

$\boldsymbol{\varepsilon}_{ij}$ é o campo de deformação dentro da inclusão;

$\boldsymbol{\varepsilon}_{kl}^t$ é a autodeformação sofrida pela inclusão;

S_{ijkl} é o tensor de Eshelby de 4ª ordem, que depende da geometria da inclusão e do material da matriz;

Ω_2 é o domínio com uma inclusão elipsoidal.

Cabe nesse momento apresentar a definição de inclusão: é uma região do material que possui as mesmas propriedades elásticas da matriz. No caso das propriedades não serem as mesmas, a região Ω_2 é chamada de heterogeneidade [Gross e Seelig, 2006]. Ao longo do texto, os termos *heterogeneidades* e *nódulos* serão usados indistintamente para indicar a imersão de um material diferente da matriz no composto.

Eshelby faz uma equivalência entre um material homogêneo com uma inclusão e um material com uma heterogeneidade permitindo definir as propriedades equivalentes de materiais compostos por uma matriz e um elipsóide de outro material.

2.3.1.5 Definição do tensor de concentração

Assim, é possível nas expressões (2.90) e (2.91), obter o tensor constitutivo equivalente de um material heterogêneo com duas fases, para deformação constante e tensão constante aplicada, respectivamente:

$$C^{*(a)} = C_M + c_I (C_I - C_M) : A_I \quad (2.90)$$

$$C^{*(b)} = [C_M^{-1} + c_I (C_I^{-1} - C_M^{-1}) : B_I]^{-1} \quad (2.91)$$

sendo:

C_M : o tensor constitutivo da matriz;

C_I : o tensor constitutivo da inclusão;

c_I : a fração de volume de inclusões em relação ao volume da amostra.

A_I e B_I : são os tensores de concentração da inclusão para os casos de aplicação de deformação constante e tensão constante, respectivamente.

O tensor de concentração de tensão permite achar o valor das respostas médias dentro da partícula (ϵ_I ou σ_I) em função do estado de carga remoto aplicado (ϵ_0, σ_0), ou seja:

$$\langle \epsilon_I \rangle = A_I : \langle \epsilon_0 \rangle \quad (2.92)$$

$$\langle \sigma_I \rangle = B_I : \langle \sigma_0 \rangle \quad (2.93)$$

O problema reside então em determinar esse tensor a ser aplicado nas equações (2.90) e (2.91) possibilitando a obtenção do tensor constitutivo equivalente C^* do material. A seguir será mostrada a definição do tensor de concentração para os modelos disponíveis na bibliografia:

Aproximações de Voigt e Reuss

Aplicando as condições de contorno das equações $u|_{\partial\Omega} = \varepsilon \cdot x \Rightarrow \langle \varepsilon \rangle_{\Omega} = \varepsilon$ e $t|_{\partial\Omega} = \zeta \cdot n \Rightarrow \langle \sigma \rangle_{\Omega} = \zeta$ aos modelos de Voigt, 1889 e Reuss, 1929, já descritos anteriormente no item 2.3.1.3, respectivamente, são gerados os tensores de concentração que valem

$$A_I = B_I = I \quad (2.94)$$

Modelos de baixas concentrações de inclusões (DILUTE)

Nestes modelos as soluções analíticas são válidas para $c_I \ll 1$, ou seja, a interação entre as partículas não é considerada. No caso de inclusões elipsoidais, tem-se:

$$A_I^{\infty} = [1 + S_M : C_M^{-1} : (C_I - C_M)]^{-1} \quad (2.95)$$

onde S_M é o tensor de Eshelby.

Há vários casos resolvidos analiticamente disponíveis na bibliografia, como se encontra em Nemat-Nasser e Hori, 1999, entre outros. O índice ∞ na expressão (2.95) indica que a matriz de concentração para a inclusão é válida considerando uma única inclusão numa matriz infinita.

Existem formas de considerar a influência da interação entre as partículas. Esta situação torna-se relevante quando a concentração de inclusões é importante. Alguns métodos clássicos para considerar esta interação são apresentados a seguir:

Método de Mori-Tanaka

Neste modelo, Mori-Tanaka, 1973 determinam que

$$A_I^{MT} = [c_I + c_2(A_I^\infty)^{-1}]^{-1} \quad (2.96)$$

onde c_2 é a fração de volume da matriz em relação ao volume da amostra.

Conceitualmente, este modelo considera que o que rodeia a inclusão é a matriz e não o meio homogeneizado.

Método Auto-Consistente

No presente método, a solução do problema é obtida a partir da solução do modelo DILUTE, através da substituição das propriedades da matriz pelas propriedades efetivas (C^*) representadas pelas equações (2.90) e (2.91). Assim, a tensão média na inclusão e o tensor de influência são expressos, respectivamente, por:

$$\langle \varepsilon \rangle_I = A_I^{SK} : \langle \varepsilon \rangle \quad (2.97)$$

$$A_I^{SK} = A_I^\infty (C_M = C^*) = [1 + S^* : C^{*-1} : (C_I - C^*)]^{-1} \quad (2.98)$$

Para o caso isotrópico com inclusões esféricas, a solução é encontrada pela resolução do sistema de equações:

$$\frac{c_2}{K^{*SK} - K_I} + \frac{c_I}{K^{*SK} - K_M} - \frac{3}{3K^{*SK} + 4G^{*SK}} = 0 \quad (2.99)$$

$$\frac{c_2}{G^{*SK} - G_I} + \frac{c_I}{G^{*SK} - G_M} - \frac{6(K^{*SK} + 2G^{*SK})}{5G^{*K}(3K^{*SK} + 4G^{*SK})} = 0$$

onde:

K_I e G_I são as constantes elásticas da inclusão;

K_M e G_M são as constantes elásticas da matriz.

Esquema Diferencial

Este método propõe proceder da mesma forma que no método auto-consistente, porém em múltiplos passos diferenciais. Ou seja, ao contrário do método auto-consistente, em que a totalidade da fração de volume da segunda fase era imersa na matriz em uma só etapa, o esquema diferencial agrega a fase particulada em passos infinitesimais.

O esquema diferencial conduz a uma equação diferencial ordinária não-linear em que o tensor elástico efetivo é uma função da fração volumétrica de inclusões, como se pode observar abaixo:

$$\frac{dC^*(c_I)}{dc_I} = \frac{1}{1-c_I} (C_I - C^*(c_I)) : A_I^\infty \quad (2.100)$$

Para o caso isotrópico, é possível obter-se a partir da equação (2.100) o seguinte sistema acoplado de equações diferenciais para o módulo de compressibilidade (K) e o módulo de elasticidade transversal (G):

$$\frac{dK^{*DS}}{dc_I} = \frac{1}{1-c_I} (K_I - K^{*DS}) \frac{3K^{*DS} + 4G^{*DS}}{3K_I + 4G^{*DS}} \quad (2.101)$$

$$\frac{dG^{*DS}}{dc_I} = \frac{1}{1-c_I} (G_I - G^{*DS}) \frac{5G^{*DS} (3K^{*DS} + 4G^{*DS})}{G^{*DS} (9K^{*DS} + 8G^{*DS}) + 6G_I (K^{*DS} + 2G^{*DS})}$$

Detalhes sobre as deduções apresentadas nesta seção podem ser encontradas em Gross e Seelig, 2006.

2.3.2 Tratamento Computacional de Materiais Micro-heterogêneos

Para simular numericamente em escala real uma estrutura de material micro-heterogêneo incorporando todos os detalhes da microestrutura, seria necessário um tamanho de elemento tão pequeno quanto a menor das micro-heterogeneidades que constituem o material. Isto poderia demandar um esforço computacional demasiadamente grande.

Para solucionar este problema, são utilizadas propriedades efetivas ou homogeneizadas, que consideram o efeito das propriedades dos micro-constituintes, permitindo a análise computacional de estruturas de materiais micro-heterogêneos em escala real, sem considerar os detalhes da micro-estrutura. Como visto anteriormente, nos itens 2.3.1.1 e 2.3.1.2, essas propriedades efetivas são obtidas através de uma homogeneização da micro-estrutura relacionando-se os valores médios de variáveis de campo dentro de um EVR do material micro-heterogêneo. Assim, podem-se utilizar discretizações que requeiram um trabalho computacional dentro das possibilidades disponíveis [Zohdi, 2002].

Uma referência clássica sobre homogeneização é o livro de Nemat-Nasser 1999, o qual trata com rigor os procedimentos e a teoria relacionados com a homogeneização dentro do campo da elasticidade, apresentando inclusive grande quantidade de soluções analíticas para muitos casos de interesse prático.

Existem duas técnicas de homogeneização básicas: a) técnicas empregadas partindo da hipótese que o material composto tem algum tipo de periodicidade em sua organização, conforme modelo da figura 2.18. Nesse caso, é possível empregar métodos que consideram a célula básica do composto, os quais permitem representar a estrutura do mesmo. A teoria das expansões assintóticas é uma metodologia consistente para tratar esse tipo de problema [Hollister e Kikuchi, 1997; Sanchez-Palencia e Sanchez Hubert, 1992]; b) Por outro lado, quando consideramos a distribuição aleatória, figura 2.19, relaxando a hipótese de um padrão ordenado entre as fases, faz-se necessário uma potência de cômputo maior. Como ilustração dessa forma de trabalho pode-se citar Zohdi e Wriggers, 2005.

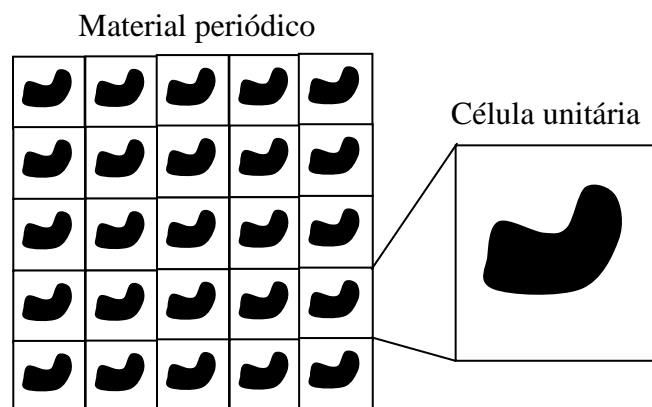


Figura 2.18 – Esboço de um material composto periódico

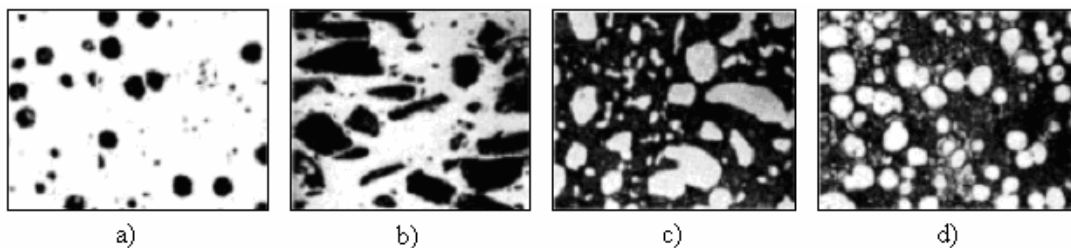


Figura 2.19 – Exemplos de materiais compostos de micro-estrutura aleatória. a) ferro fundido nodular, b) matriz de alumínio com partículas de carbeto de silício, c) Liga alumínio-titânio d) Matriz de prata com partículas de ouro [Zohdi, 2002]

As técnicas mencionadas, tanto considerando periodicidade na ordem das fases do compósito, ou não, são conhecidas. Mas, quando os fenômenos que se desejam homogeneizar são não-lineares, o problema se complica enormemente e, nesse caso, muito se tem, ainda, por fazer. Como uma revisão do estado da arte das técnicas desenvolvidas até o presente pode-se citar Kanouté et al. 2009.

Trabalhando com os conceitos de micro mecânica abordados, considerando ou não a periodicidade, junto com técnicas de otimização conhecidas, é possível desenvolver ferramentas numéricas que permitem auxiliar na construção da microestrutura de materiais compostos. Em Zohdi e Wriggers, 2005, apresenta-se um exemplo disso: nesse trabalho, se emprega os algoritmos genéticos para determinar a forma mais adequada para as partículas de um composto particulado, a fim de minimizar os gradientes de tensões internas e, indiretamente, aumentar a vida à fadiga do material.

Estas duas abordagens de homogeneização são descritas de forma mais detalhada na continuação:

2.3.2.1. Materiais compostos periódicos

A homogeneização em materiais compostos periódicos se baseia na teoria das expansões assintóticas e foi introduzida na década de 70. Dentre alguns dos seus precursores estão Sanchez-Palencia e Babuska. As principais características desse tipo de homogeneização são: i) considera a existência de periodicidade na micro-estrutura do material, o que, a priori, é uma maneira de impor a condição de homogeneidade estatística do mesmo [Terada et al., 2000]; ii) considera que o deslocamento total de um material micro-heterogêneo pode ser representado por uma expansão assintótica que compreende o

deslocamento em todas as escalas consideradas, (equação (2.102)) [Hollister e Kikuchi, 1997].

$$u_i^\eta(x^0, x^1, x^2, \dots) = u_i^0(x^0, x^1, x^2, \dots) + \eta^1 \tilde{u}_i^1(x^0, x^1, x^2, \dots) + \eta^2 \tilde{u}_i^2(x^0, x^1, x^2, \dots) + \dots \quad (2.102)$$

Onde:

u_i^η : Deslocamento total na menor escala micro-estrutural;

u_i^0 : Deslocamento na escala macroscópica;

\tilde{u}_i^1 e \tilde{u}_i^2 : Perturbação no deslocamento total devido às escalas micro-estruturais 1 e 2;

η : Relação entre a η -ésima escala micro-estrutural analisada, e a escala imediatamente anterior (fator de escala);

x^0, x^1, x^2, \dots : Coordenadas nas escalas macroscópica e demais escalas micro-estruturais analisadas, respectivamente.

De maneira geral, o material micro-heterogêneo periódico é modelado utilizando-se como estrutura básica uma célula unitária de morfologia simples, a qual funciona como elemento de volume representativo do material (Figura 2.20). Esta característica reduz de maneira significativa o tempo e esforço computacional nas análises.

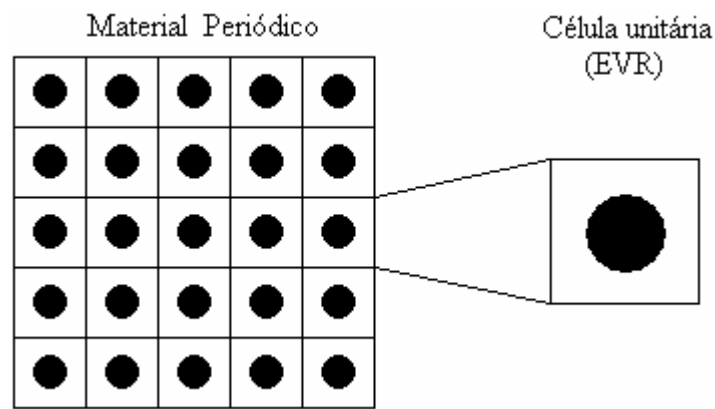


Figura 2.20 – Esboço de um material periódico micro-heterogêneo

Apesar destas vantagens, a homogeneização periódica apresenta o inconveniente de que, na prática, a maioria dos materiais micro-heterogêneos possuem distribuição aleatória dos seus constituintes. Por outro lado, a distribuição das variáveis de campo (tensões, deformações, energia, etc.) dentro de um EVR que considera somente uma célula unitária, difere da distribuição das variáveis de campo dentro de outro que considera a forma e distribuição real dos micro-constituintes, ainda que ambos possuam a mesma fração de

volume dos constituintes [Terada et al, 2000]. Por esta razão alguns pesquisadores consideram que o método é aplicável somente sob comportamento linear elástico [Kusnetsova, 2001].

Contudo, a hipótese de periodicidade tem sido aplicada à homogeneização de materiais com comportamento viscoplástico e elastoplástico [Van der Sluis et al, 1999; Kaminski et al., 2001]. Para tanto, o EVR está geralmente constituído por um conjunto de células unitárias e não mais por uma única célula unitária. Aspectos relacionados com análise computacional de micro-mecânica de fratura dúctil [Armero et al, 2000; Kim et al., 2004] também têm sido estudados utilizando a hipótese de periodicidade, considerando-se apenas uma única célula de morfologia simples.

2.3.2.2 Tratamento de materiais compostos sem periodicidade

Quando se trata de materiais compostos onde não há periodicidade na distribuição da segunda fase, o seu tratamento está baseado na definição de dois conceitos básicos, já descritos anteriormente, a saber: elemento de volume representativo e homogeneização.

Os critérios de determinação do EVR mais utilizados na obtenção de propriedades equivalentes de micro-estruturas heterogêneas de distribuição aleatória são os propostos por Hill e Hashin [Hollister e Kikuchi, 1997].

Hill define, em 1963, o elemento de volume representativo como uma amostra de material que deve: i) conter todos os constituintes que caracterizam a micro-estrutura do material heterogêneo em estudo; ii) conter um número suficiente desses constituintes para que as propriedades médias correspondentes ao EVR independam das condições de contorno aplicadas, desde que tais condições sejam macroscopicamente uniformes, ou seja, os valores oscilam em torno de um valor médio com um desvio padrão pequeno, e o mesmo passa a ser insignificante a pouca distância da superfície [Aidun et al, 1999].

O critério proposto por Hill, em 1963, para a determinação do EVR é chamado de condição energética de Hill, e está representada pela equação (2.103):

$$\langle \sigma : \varepsilon \rangle_{L_2} = \langle \sigma \rangle_{L_2} : \langle \varepsilon \rangle_{L_2} \quad (2.103)$$

A relação representada pela equação (2.103) apóia-se na figura 2.21 e tem implícitas as duas considerações anteriores sobre EVR, uma vez que a mesma só se satisfaz para

tamanhos de volume de controle grandes e pequenas flutuações nos campos de tensão e deformação no domínio analisado.

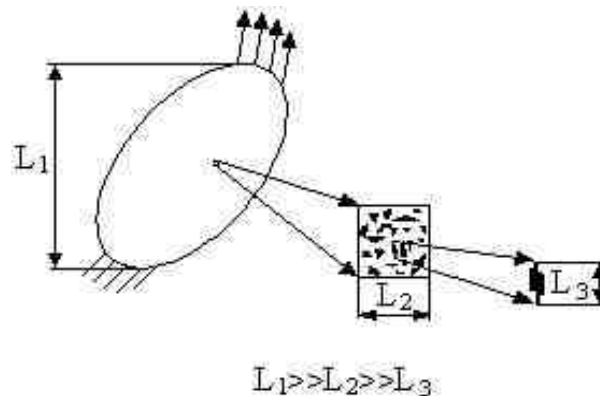


Figura 2.21 – Esquema ilustrativo do tamanho do EVR [Zohdi, 2002]

Hashin, em 1983, estabelece que o EVR deve ser grande em comparação com a dimensão e espaçamento da microestrutura típica presente dentro do mesmo. Além disso, estabelece o conceito de homogeneidade estatística de materiais heterogêneos e a necessidade de sua existência para a determinação do elemento de volume representativo. Define-se micro estrutura estatisticamente homogênea, como aquela em que suas características micro estruturais são as mesmas em qualquer elemento de volume analisado, independentemente da sua localização. Apoiando-se no esquema da figura 2.22, se existir homogeneidade estatística, as características da microestrutura dos volumes V_1 , V_2 e V_3 são iguais [Hollister e Kikuchi, 1997].

As duas definições anteriores sugerem que o EVR deve possuir representatividade suficiente dos micro constituintes do material analisado. Portanto, deve ser grande em comparação com o tamanho característico de partícula presente na micro estrutura, mas ao mesmo tempo deve ser pequeno em comparação com as dimensões das estruturas mecânicas no nível macroscópico, pois dentro do mesmo serão considerados os detalhes da micro estrutura ao ser realizada a análise computacional. Em resumo, deve-se satisfazer à relação $L_1 \gg L_2 \gg L_3$ (Figura 2.21).

Apesar de existirem controvérsias quanto à utilização de um ou outro critério [Aidum et al., 1999], ambos têm sido utilizados com sucesso [Terada et al., 2000; Kanit, 2003], verificando-se que o tamanho do EVR está influenciado por fatores tais como, grau de dispersão dos micro constituintes, fração de volume dos mesmos, contraste nas suas propriedades mecânicas, propriedade que se quer homogeneizar, entre outros.

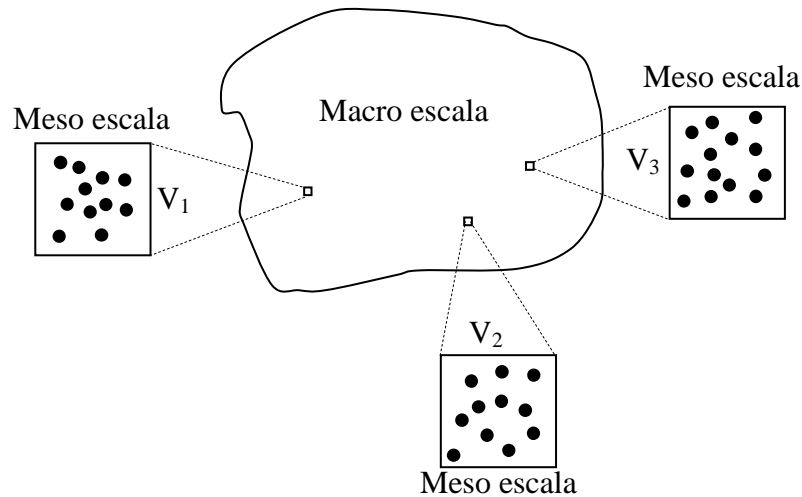


Figura 2.22 – Desenho esquemático para explicar a homogeneidade estatística.

Considerando o parâmetro adimensional δ_c que resulta da relação entre a dimensão do volume analisado L_2 e a dimensão característica do micro constituinte L_3 , equação (2.104), Ostoja-Starzewski, 2006, sugere que o δ_c do EVR pode se encontrar entre 10 e 100, dependendo das características do problema estudado.

$$\delta_c = \frac{L_2}{L_3} \quad (2.104)$$

Mesmo que sejam considerados, dentro do EVR, os detalhes da micro estrutura, o trabalho computacional envolvido é consideravelmente menor que se fosse realizada a mesma análise na estrutura em escala real.

A determinação do tamanho do EVR é realizada resolvendo problemas de valores de contorno, de amostras de material micro heterogêneo nas quais se verifica a resposta do valor médio de variáveis de campo. Usualmente vai se incrementando a quantidade de micro constituintes nas amostras analisadas, mantendo constantes suas frações de volume (Figura 2.23).

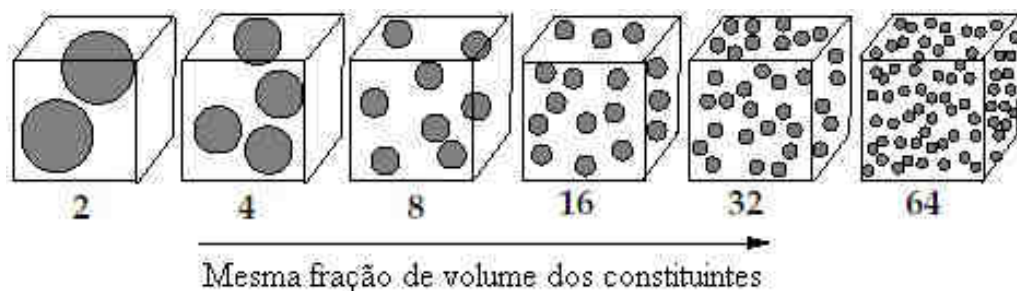


Figura 2.23 – Sequência de volumes de material numa análise de EVR [Zohdi, 2002]

Segundo as definições anteriores, será considerada uma amostra representativa de material aquela em que as propriedades homogeneizadas independam das condições de contorno (Figura 2.24) ou então, aquela para a qual se verifique homogeneidade estatística (Figura 2.25).

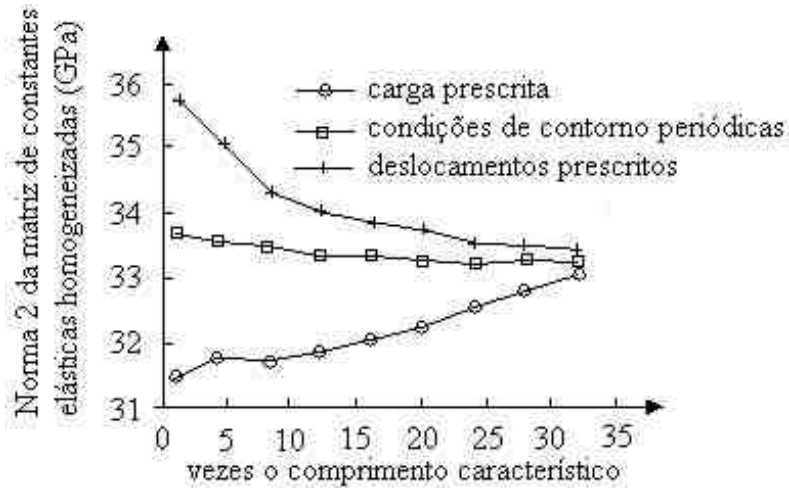


Figura 2.24 – Verificação da independência das condições de contorno com o incremento do volume de controle encontrado por Terada et al., 2000

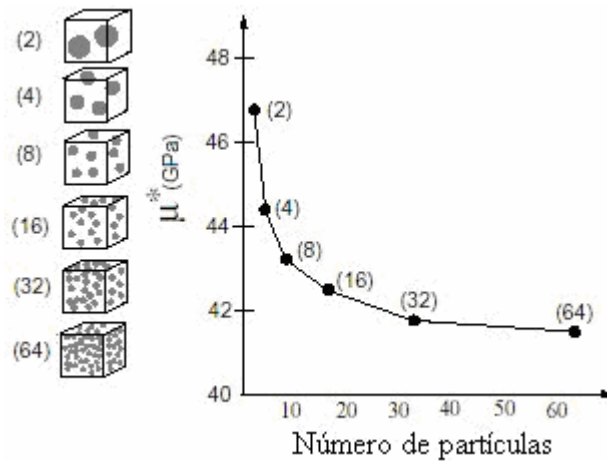


Figura 2.25 – Convergência da resposta em termos do módulo de cisalhamento da amostra de material (G^* representado por μ^*), com o incremento do número de partículas [Zohdi, 2002]

2.4 Fundamentos de Algoritmos Genéticos

2.4.1 Otimização com Algoritmos Genéticos

Muitos métodos de otimização que utilizam cálculo de gradientes, buscando solução na vizinhança de um ponto inicial, encontram problemas de eficiência ou robustez na aplicação em diferentes problemas de otimização em engenharia. Se o problema apresenta vários ótimos locais, o resultado obtido vai depender da escolha do ponto inicial, e o ótimo global pode não ser encontrado. Além disso, quando a função objetivo (função que se quer otimizar) e suas restrições apresentam máximos ou mínimos, a procura da solução através de cálculo de gradientes torna-se difícil e instável.

Os Algoritmos Genéticos (AGs) constituem um método estocástico de solução de problemas de otimização, criado por Holland, 1975 e desenvolvido por diversos pesquisadores, com destaque para Goldberg, 1989. Estes algoritmos, baseados nos conceitos da teoria da evolução natural de Darwin realizam uma busca de soluções globais, combinando sobrevivência do melhor indivíduo (solução) com troca aleatória de informações para formar um algoritmo de busca estruturado. A cada geração, um novo conjunto de indivíduos artificiais (um tipo de estrutura de dados) é criado a partir de características (dados) do conjunto antigo.

Os algoritmos genéticos, a partir da definição de um problema a ser estudado e de uma função objetivo, tentam explicar de maneira abstrata o processo adaptativo de sistemas naturais. Para tal, criam programas de otimização que usam os conceitos de seleção natural para resolver problemas de Engenharia [Goldberg,1989].

A utilização de métodos de otimização estocásticos, entre eles os Algoritmos Genéticos, é, em muitos casos, conveniente, uma vez que, de forma geral, os problemas de engenharia são complexos, não-lineares, de difícil representação e descritos por funções que nem sempre são diferenciáveis. Estes métodos estocásticos utilizam um conjunto de ações que buscam a solução ótima de maneira aleatória, não necessitando de muitas informações sobre o comportamento do problema (cálculo de gradientes), apenas das informações contidas na função objetivo.

Os algoritmos genéticos procuram a solução ótima partindo de uma população definida por diferentes pontos no domínio da função estudada. Concluída a construção da população inicial obtém-se um conjunto de pontos solução do problema entre as quais se

escolhe a melhor ou a elite criando um subconjunto de soluções. Uma vantagem dos AGs é que este algoritmo não fica restrito a análise e procura de ótimos locais procurando soluções em regiões maiores, a nível global. Os AGs, desta forma, são algoritmos adequados para localizar rapidamente regiões promissoras dentro do espaço de busca. No entanto, são lentos para refinar as soluções encontradas.

2.4.2 Estrutura dos Algoritmos Genéticos

Partindo de uma população de indivíduos, cada um com um valor de aptidão associado, os AGs evoluem através de operações genéticas como cruzamentos e mutações, para uma nova geração de indivíduos. Para essa evolução são utilizados os princípios Darwinianos de reprodução e sobrevivência dos mais aptos, sendo que esta aptidão é calculada a partir do valor da função objetivo que se quer otimizar.

Um indivíduo, ou uma solução do problema, é representado pelo cromossomo. Este cromossomo é uma estrutura de dados (vetor ou cadeia de bits) que representa um conjunto de parâmetros (variáveis de projeto) da função objetivo. Cada parâmetro é codificado em um gene do cromossomo (posição do vetor ou parte da cadeia de bits), sendo o conjunto de informações contidas no cromossomo um genótipo.

O AG parte de uma população inicial aleatória que vai gerar indivíduos para a próxima geração, consistindo num processo que chega ao fim (converge) quando for atingido o número máximo de gerações ou quando for satisfeito outro critério de parada.

A seguir são apresentadas de forma sucinta as principais etapas de um AG:

i) Geração da População Inicial: Aqui ocorre, normalmente, uma geração aleatória da população inicial. Cabe observar que, se uma população pequena for gerada inicialmente, é provável que algumas regiões do espaço de busca não sejam representadas.

ii) Avaliação da População: A população é avaliada pela função objetivo. As restrições de projetos geralmente utilizadas em um problema de otimização podem ser trabalhadas de duas formas: aplicando penalidades à função de aptidão e definindo o quanto a solução viola a restrição, ou então fazendo de cada restrição uma função objetivo diferente e partindo para uma abordagem multi-objetivo.

iii) Mapeamento da Função Objetivo: Nesta etapa é realizado o cálculo da aptidão ou grandeza que caracteriza a variável do problema a ser otimizado. Deve-se ter o cuidado para que os valores da função objetivo sejam bons descritores dos valores da aptidão. Isto se deve ao fato de que a função objetivo pode fornecer valores muito próximos para um conjunto de indivíduos, tornando a seleção aleatória demais ou alguns valores podem ser muito elevados em relação aos demais do grupo, causando convergência prematura devido a grande influência na seleção causada por estes indivíduos bons na geração presente, mas que podem não ser necessariamente os melhores no espaço global de busca. Desta forma, os valores de aptidão precisam ser mapeados (ou escalonados) de maneira que todos os indivíduos possam ser selecionados para as operações genéticas.

iv) Seleção: Nesta etapa ocorre a escolha de indivíduos do grupo atual que servirão como pais para os indivíduos da próxima geração. Os meios mais comuns para realizar esta seleção são: o *Método da Roleta* e a *Seleção por Torneio*, que podem ser vistos com mais detalhes em Goldberg, 1989.

v) Cruzamento (Crossover): Consiste na recombinação do material genético (variáveis do vetor ou parte da cadeia de bits) entre dois indivíduos, resultando em um novo indivíduo para a próxima geração, obedecendo a uma probabilidade de cruzamento.

vi) Mutação: É um operador genético que modifica os genes de um indivíduo que vai compor a próxima geração, de acordo com uma probabilidade de mutação. O objetivo desta fase é melhorar a diversidade dos cromossomos de uma população. Porém, observa-se que ocorre destruição de parte da informação genética contida. Portanto, deve ser utilizada uma baixa probabilidade de mutação (valores altos tornam a busca, embora abrangente no espaço, aleatória).

vii) Elitismo: Aqui ocorre a cópia do melhor cromossomo de uma população para a geração seguinte, sem passar pelo processo seletivo, garantindo a convergência.

viii) Critérios de avaliação: A otimização através de AGs é um método paralelo de avaliação da função objetivo de um indivíduo. Cada etapa de avaliação é independente das demais. Devido à esta característica, o emprego de programas comerciais fechados de elementos finitos podem ser utilizados com AGs facilmente.

2.4.3 Abordagem Multi-objetivo

A solução ótima que se busca, muitas vezes vem de várias características que se quer maximizar ou minimizar, e não de uma somente. Assim, a função objetivo passa a representar uma combinação de várias características na abordagem multi-objetivo.

Grandezas diferentes, como, por exemplo, desempenho e custo, podem ser avaliados de forma mais adequada sob a ótica multi-objetivo. Uma solução de menor custo e com menor desempenho pode ser comparada com outra de maior custo e melhor desempenho na abordagem multi-objetivo. Desconhecendo-se o comportamento de várias grandezas que determinam o comportamento de um sistema e que vão constituir a função objetivo do problema, pode avaliar o problema obtendo um grande número de soluções não-dominadas para que com isto se caracterize a região de soluções eficientes (também denominada de região ótima de Pareto).

A abordagem de um problema de otimização multi-objetivo pode ser realizada de dois modos: a) estabelecendo pesos (ou prioridades) para cada objetivo da função; b) encontrando um conjunto de soluções ótimas (não dominadas), que se aproximem da região de Pareto, para posterior escolha do projetista.

2.5 Modelagem Numérica

Os fenômenos da natureza podem de uma forma geral, ser representados com o emprego das leis da física, expressos através de equações algébricas, diferenciais e/ou integrais. Neste contexto, é comum a utilização de aproximações numéricas para se obter as variáveis envolvidas nos eventos e, assim, representar tais situações. São exemplos dessas modelagens: a determinação da temperatura num certo ponto de uma caldeira, o valor da deflexão no centro de uma laje carregada ou a deformação ao longo do tempo apresentada por um corpo sujeito a uma ação impulsiva, entre outros.

Existem alguns métodos de análise exatos que facilitam a obtenção das equações que governam determinados problemas de forma relativamente fácil. Entretanto, há casos em que a definição das condições de contorno é complexa, resultando assim, em tarefas difíceis ou impossíveis até os dias atuais. Desta forma, os métodos aproximados de análise constituem-se caminhos alternativos para encontrar uma solução. Entre tais métodos numéricos, podem ser citados, entre outros, o Método dos Elementos Finitos (*MEF*), o Método dos Elementos

Discretos (*MED*), o Método dos Elementos de Contorno (*MEC*) e métodos híbridos, que combinados aproveitam as vantagens dos outros antes mencionados.

Na presente seção, será comentada de forma breve alguns conceitos básicos sobre o *MEF* e o *MED*.

2.5.1 Método dos Elementos Finitos (MEF)

O Método dos Elementos Finitos é hoje um dos mais utilizados para obtenção de soluções em problemas de engenharia. Este consiste em discretizar o domínio em estudo numa série de domínios elementares, cuidando-se para manter a compatibilidade nos deslocamentos entre os mesmos. Há uma grande variedade de programas comerciais que utilizam este método, entre eles o Ansys ,2009, o LS-Dyna, 2009 e o Abaqus/Simulia, 2009.

De uma forma geral, estes sistemas permitem realizar análise de diversos problemas de valores de contorno utilizando o *MEF*, envolvendo análises do tipo: a) Estática, b) Modal: determinação de modos e frequências, c) Superposição modal: avaliação de problemas de vibrações forçadas, d) Integração direta da equação do movimento resultante da discretização espacial utilizando um esquema de integração implícito.

Tais programas permitem também considerar as não linearidades físicas e geométricas para as análises.

2.5.2 Método dos Elementos Discretos (MED)

Outro método muito utilizado para simulações numéricas em engenharia é o método dos elementos discretos (*MED*), que foi introduzido na década de setenta. Na literatura, Cundall aparece como um dos seus principais precursores [Cundall, 1971; Cundall, 1979]. No *MED* considera-se o meio a ser modelado como um sistema de partículas discretas, geralmente de forma esférica e massa conhecida. De maneira geral, é considerada uma interação viscoelástica entre as partículas.

O método dos elementos discretos tem sido aplicado com sucesso à simulação computacional de processos que envolvem comportamentos predominantemente descontínuos, como é o caso do comportamento de sólidos não-coesivos, onde se destacam os trabalhos de estudo do transporte de sedimentos num fluido [Goth e Sakai, 1997; Lin et al., 2000], e simulação computacional de filtros com camadas granulares [Chou et al., 2000]. O *MED* também tem sido utilizado no estudo de processos de degradação e fratura em sólidos

coesivos, entre os quais podem ser citados: simulação de explosões em rochas [Ladislav e Dutko, 2003; Procházka, 2004], fratura de materiais compostos como o concreto [Sawamoto et al., 1998; Camborde et al., 2000; Brara et al., 2001; Wittel et al., 2003], entre outras aplicações.

Uma das maiores desvantagens do método está no fato do grande trabalho computacional requerido, pois, como cada partícula é considerada separadamente, existe a execução de múltiplos processos em paralelo. Neste sentido têm-se desenvolvido alguns trabalhos que otimizam os recursos de processamento computacional, fazendo uso de técnicas avançadas de programação em paralelo [Schafer et al., 2002] ou a combinação com o método dos elementos finitos [Gethin et al., 2001; Munjiza et al., 2004], entre outros.

Já no contexto da versão do MED utilizada neste trabalho ele já foi utilizado por [Dalguer et al., 2003] para estudar de fratura dinâmica em rochas, produto de terremotos, impacto em concreto armado [Riera e Iturrioz, 1998], fenômenos de fratura por impacto em polímeros [Barrios et al, 2002], na simulação do comportamento micro mecânico de um ferro fundido nodular em Batista, 2007, entre outros exemplos.

2.5.2.1 Características do MED Utilizado Neste Trabalho

A idéia para a representação do contínuo segundo a formulação em elementos discretos utilizada neste trabalho, pode ser atribuída a Hrennikoff, 1942 e Absi, 1971. Esses pesquisadores utilizaram estruturas de barras de rigidez equivalente a estruturas contínuas com o objetivo de simplificar o modelo em análise.

Este método está baseado na representação espacial do contínuo utilizando elementos de barra, interconectados entre si, que possuem rigidez equivalente ao contínuo que se deseja representar. As barras estão dispostas no espaço segundo um arranjo em treliça formando a unidade básica de formato cúbico (módulo cúbico) que aparece no detalhe da figura 2.26, com barras normais de comprimento L e barras diagonais de comprimento $\sqrt{3}L/2$. Para esta distribuição espacial das barras (formato cúbico), Hayashi, 1982 determinou a rigidez das mesmas para poder representar um contínuo isotrópico equivalente. As equações 2.105 e 2.106 representam as rigidezes obtidas para as barras normais e diagonais respectivamente:

$$EA_n = \frac{L^2 (9+8\delta)}{18+24\delta} \times E \quad (2.105)$$

$$EA_d = \frac{2\delta L^2 (9+8\delta)}{27+36\delta} \times E \quad (2.106)$$

Onde:

EA_n e EA_d : rigidez das barras normais e diagonais, respectivamente;

E : módulo de Young;

L : comprimento das barras normais;

δ : coeficiente que relaciona as propriedades unidirecionais na direção normal e diagonal.

Para esta distribuição (Figura 2.26), e utilizando as constantes elásticas convencionais, Hayashi, 1982 determinou que:

$$\delta = \frac{9\nu}{(4-8\nu)} \quad (2.107)$$

As massas estão concentradas nos nós de interconexão das barras e as mesmas vão depender do volume de influência do nó. Cada nó possui três graus de liberdade (deslocamento em três direções).

As equações de movimento resultantes da discretização espacial são desacopladas (a matriz de massa é diagonal e o amortecimento é proporcional à massa) e são integradas no tempo através de um esquema de integração explícita, o Método das Diferenças Finitas Centrais [Bathe, 1996]. Cada equação desacoplada de movimento é do tipo apresentado em (2.108).

$$m \frac{d^2 u_i}{dt^2} + c \frac{du_i}{dt} = F \quad (2.108)$$

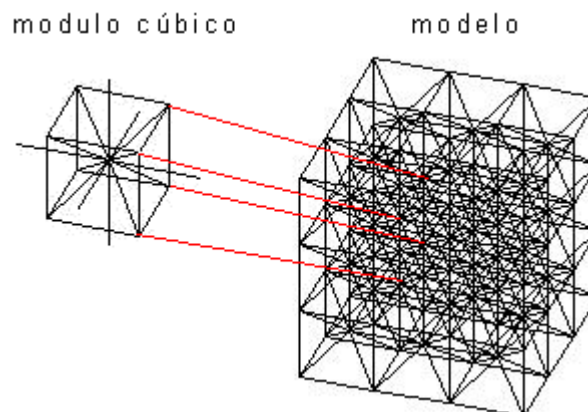


Figura 2.26 – Modelo de barras gerado usando o Método dos Elementos Discretos e detalhe do módulo cúbico

Sendo:

m : massa nodal;

c : coeficiente de amortecimento proporcional à massa;

u_i : deslocamentos nodais correspondentes às três direções ortogonais de referência;

F : componentes das forças internas (aplicadas através das barras coincidentes em um nó) e externas.

A relação constitutiva correspondente a cada barra foi proposta por Rocha, 1989 e responde ao diagrama bilinear inspirado na proposta por Hilleborg, 1978, mostrado na figura 2.27. A mesma está caracterizada por dois parâmetros fundamentais: a deformação crítica da barra (ε_p) e a energia específica de fratura (G_f). A relação entre ε_p e G_f foi estabelecida como critério de falha sobre tração e sua dedução está baseada na mecânica linear elástica da fratura. Considerando a tensão crítica como sendo ε_p , e as expressões que vinculam o fator de intensidade de tensões com esta tensão crítica, e sua relação com a energia específica de fratura G_f , Rocha, 1989 apresenta a relação (2.109) para determinar ε_p .

$$\varepsilon_p = Rfc \cdot \sqrt{\frac{G_f}{E(1-\nu^2)}} \quad (2.109)$$

Sendo Rfc o fator de falha, que contém informação sobre o tamanho do defeito (2.110) e suas características geométricas Y , sendo possível escrever:

$$Rfc = \frac{1}{Y\sqrt{d}} \quad (2.110)$$

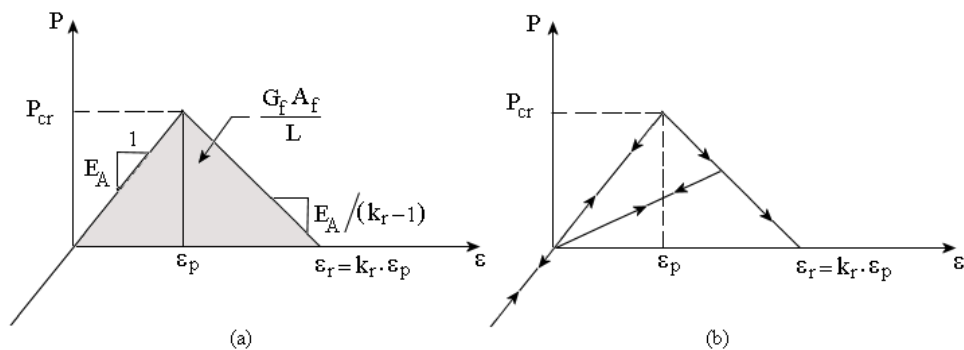


Figura 2.27 – Lei constitutiva bilinear atribuída a cada barra do modelo (a) parâmetros envolvidos na lei constitutiva, (b) comportamento em tração e compressão [Rocha, 1989]

A relação entre G_f e a área abaixo da curva da lei constitutiva da figura 2.27 está dada pela relação (2.111) que se apresenta na continuação.

$$\int_0^{\varepsilon_r} F(\varepsilon) d\varepsilon = \frac{G_f A_f}{L} = \frac{K_r \varepsilon_p^2 EA_f}{2} \quad (2.111)$$

Onde A_f é a área de fratura da barra determinada por Rocha, 1989 em $0.1385xL^2$.

Para que a relação constitutiva seja estável, a condição de $K_r \geq 1$ tem que ser garantida, sendo que K_r é função das propriedades mecânicas do material e da discretização adotada (L). Desta forma, é considerado no parâmetro K_r , o efeito de escala, característica que está presente em fenômenos onde ocorre fratura.

Como pode ser observado na figura 2.27, é considerado que as barras apresentam um comportamento elástico em compressão independente da carga aplicada, portanto a ruptura sempre acontecerá por tração. A consideração de distribuição aleatória de alguma das propriedades é muito simples de ser implementada, e esta característica já foi utilizada na modelagem de concreto e de solos [Rocha, 1989; Rios, 2004].

A representação do sólido através de um sistema de treliças visando a obtenção de tensores constitutivos pré-determinados também tem sido obtida através da homogeneização inversa [Sigmund, 1994].

2.6 Resolução de Problemas Não-Lineares através do MEF

Para se resolver problemas não-lineares, como os apresentados na seção 2.1.2, utilizando o MEF, é necessária uma formulação do problema em taxas. Para isso, ao iniciar cada novo passo, deve-se determinar a configuração deformada aproximada do corpo em estudo. Mas para se saber se essa configuração é realmente a configuração que o corpo irá assumir após a aplicação do carregamento, é preciso que o corpo esteja em equilíbrio. A forma de resolver equações de equilíbrio não-lineares é dividindo-as em várias etapas lineares, também chamadas de *passos*, através de processos de linearização das equações do problema. Dessa forma, tem-se a solução de um problema não-linear através de vários problemas lineares resolvidos cada um dentro de um *passo*. Assim, pode-se dizer que os problemas não-lineares são resolvidos substituindo-se um conjunto de equações diferenciais que caracterizam o equilíbrio por equações integrais através da aplicação do Princípio dos Trabalhos Virtuais (PTV) [Cook, Malkus e Plesha, 1974], assim como em problemas lineares.

Às vezes, algoritmos de ajuste não-lineares são aplicados para poder aumentar o incremento de tempo sem perder precisão na integração. Como exemplo pode-se citar o Método de Newton, o Método do Comprimento de Arco, Método de Controle da Energia, entre outros. Mais detalhes sobre este tema pode ser obtido em Bathe, 1996.

Dependendo de como é realizada a aproximação da função $u(t)$ (deslocamentos em função do tempo) dentro do intervalo, ter-se-á diferentes métodos de integração. Existe uma primeira classificação dos métodos em implícitos e explícitos. A seguir, será tratado sobre o método explícito de integração, utilizado neste trabalho.

2.6.1 Análise explícita

Um dos métodos mais utilizados dentro dos esquemas explícitos é o Método das Diferenças Centrais. As acelerações e velocidades são calculadas no tempo t , com as seguintes expressões

$$\{\ddot{u}_t\} = \frac{1}{\Delta t^2} (\{u_{t+\Delta t}\} - 2\{u_t\} + \{u_{t-\Delta t}\}) \quad (2.112)$$

$$\{\dot{u}_t\} = \frac{1}{2\Delta t} (-\{u_{t-\Delta t}\} + \{u_{t+\Delta t}\}) \quad (2.113)$$

que são substituídas na equação do movimento

$$[M]\{\ddot{u}_t\} + [C]\{\dot{u}_t\} + [K]\{u_t\} = \{f_t\} \quad (2.114)$$

onde:

$[M]$: é a matriz de massa;

$[K]$: é a matriz de rigidez;

$[C]$: é a matriz de amortecimento.

Trabalhando algebricamente a equação (2.114), tem-se

$$\left(\frac{1}{\Delta t^2} [M] + \frac{1}{2\Delta t} [C] \right) \{u_{t+\Delta t}\} = \{f_t\} - \left([K] - \frac{2}{\Delta t^2} [M] \right) \{u_t\} - \left(\frac{1}{\Delta t^2} [M] - \frac{1}{2\Delta t} [C] \right) \{u_{t-\Delta t}\} \quad (2.115)$$

Desta forma, os deslocamentos calculados para o tempo $(t+\Delta t)$ são obtidos em função dos valores conhecidos no tempo $(t-\Delta t)$. É importante observar que a solução é montada utilizando a equação de equilíbrio no tempo t e, por este motivo, o procedimento de integração é chamado de explícito [Bathe, 1996]. Ainda como observação, nota-se na equação (2.115) que para tal esquema de integração a fatorização da matriz de rigidez na solução passo a passo não é requerida. Isto permite resolver problemas nos quais as matrizes de rigidez estejam mal condicionadas (estrutura hipostática).

Este método é condicionalmente estável, pois converge para intervalos de integração menores que um determinado valor crítico. Este limite de estabilidade, no caso de não haver amortecimento, é dado em termos da maior frequência natural circular do sistema, como

$$\Delta t \leq \frac{2}{\omega_{\max}} \quad (2.116)$$

Este procedimento é utilizado geralmente quando o problema a ser resolvido é transiente (impactos, explosões, etc.), e quando as não-linearidades no comportamento do material são esperadas em regiões localizadas. Neste caso seria necessário um intervalo de tempo de integração suficientemente pequeno para descrever o fenômeno estudado.

Uma aproximação para o limite de estabilidade, frequentemente escrito como o menor tempo de trânsito da onda elástica dilatacional, C_d , através de qualquer dos elementos da malha, tem a forma

$$\Delta t \approx \frac{L_{\min}}{C_d} \quad (2.117)$$

Algumas particularidades do uso da integração explícita no contexto do ABAQUS/SIMULIA

O incremento de tempo estável escolhido por Abaqus/Explicit [Abaqus/Simulia, 2009] será um fator entre $\frac{1}{\sqrt{2}}$ e 1 para um modelo bidimensional e entre $\frac{1}{\sqrt{3}}$ e 1 para modelo tridimensional, inferior a estimativa dada pela equação (2.117).

Se o modelo contém apenas um tipo de material, o incremento de tempo inicial é diretamente proporcional ao tamanho do menor elemento da malha. Se a malha contém elementos de tamanho uniforme, mas contém descrições de vários materiais, o elemento com a maior velocidade de onda vai determinar o incremento de tempo inicial [Abaqus/Simulia, 2009].

Em problemas não-lineares, a frequência mais alta do modelo irá mudar continuamente, o que conseqüentemente modifica o limite de estabilidade. O Abaqus/Explicit [Abaqus/Simulia, 2009] tem duas estratégias para controle do tempo de incrementação: o incremento de tempo automático (*automatic time incrementation*) e o incremento de tempo fixo (*fixed time incrementation*). O primeiro é o esquema padrão de incrementação do tempo no Abaqus/Explicit e é totalmente automático, não exigindo nenhuma intervenção do usuário. No segundo, o tamanho do incremento de tempo fixo pode ser especificado pelo usuário. Nesse último caso, o Abaqus/Explicit não verifica se a resposta calculada é estável durante a etapa. O usuário deve garantir que uma resposta válida foi obtida através de cuidadosa verificação da história da energia e outras variáveis de resposta. A análise com incremento de tempo fixo pode ser útil quando uma representação mais precisa da resposta de um problema é necessária. Neste caso, um tamanho de incremento de tempo menor do que o automático pode ser usado.

Quanto ao amortecimento, o Abaqus/Explicit possui várias formas de introdução desse fator na análise, com o objetivo, ou de limitar o número de oscilações ou de adicionar amortecimento físico ao sistema. Dentre estes métodos, está a adição do *bulk viscosity* (*Linear bulk viscosity* e *Quadratic bulk viscosity*), estimador associado às deformações volumétricas, que tem por finalidade ajustar o modelo para efeitos dinâmicos de grandes velocidades de carga.

Outro aspecto a ser considerado em análise explícita é a curva de amplitude do carregamento aplicado ao modelo. Tais curvas permitem especificar a velocidade de aplicação da excitação. O Abaqus/Explicit possui várias formas para as curvas de amplitude, como a *Time span* (*Step time* ou *Total time*), a *Tabular*, a *Periodic*, a *Modulated*, entre outras.

Finalizando esta seção, salienta-se que o método de integração explícita é mais adequado, em relação ao implícito, quando grandes não-linearidades são esperadas na equação de movimento [HIBBIT et al., 1995]. Assim, realizou-se este estudo na forma explícita visando que o mesmo possa ser aprofundado para casos em que se preveja a perda de aderência entre matriz e nódulos, a aplicação de taxas de deformação ou análise de propriedades visco elásticas, por exemplo.

3 REVISÃO BIBLIOGRÁFICA

A utilização de métodos numéricos para o estudo de fenômenos micro-estruturais vem sendo largamente desenvolvida durante as últimas três décadas. Apesar de todo o ganho computacional que se obteve ao longo desse período, com o aumento da capacidade de processamento computacional, a análise de materiais ainda é um problema em aberto [Zohdi, 2002]. O que se pode dizer é que, mesmo que modernos métodos numéricos tenham assumido ultimamente um papel significativo na análise e modelagem de materiais heterogêneos, ainda há muito que fazer, tanto no campo não-linear, quanto no âmbito dos materiais linearmente elásticos, com deformações infinitesimais.

3.1 Polímeros reforçados com partículas de borracha

Neste contexto, a tecnologia dos materiais poliméricos compostos constitui uma área de grande interesse científico e comercial. Devido a sua versatilidade e a facilidade de processamento, os termoplásticos modificados são um dos produtos de crescimento mais rápido na indústria dos materiais [Fasce et al., 2004]. Neste sentido, o polipropileno (PP) e o polimetilmetacrilato (PMMA) são materiais muito difundidos no meio regional [ABIPLAST, 2009; CAIP, 2009] que apresentam propriedades que os tornam elegíveis para um grande número de aplicações. Mas estas aplicações estão limitadas pela reduzida tenacidade e elevada sensibilidade a entalhes dos mesmos. Um método usualmente utilizado para aumentar a tenacidade e resistência ao impacto de polímeros de matriz vítrea é seu reforço com partículas de borracha [Argon e Cohen, 2003]. Estas partículas podem ser só de borracha, ou ter uma estrutura interna de acrílico e uma cobertura de borracha (chamadas tipo “core shell”). Os mecanismos de aumento da tenacidade neste tipo de misturas poliméricas têm dado lugar a um importante número de estudos experimentais em distintos polímeros [Garg e Mai, 1998; Heim et al., 1993]. Estes estudos têm demonstrado que o escoamento em escala local pode ser o principal mecanismo de deformação. As partículas de borracha cavitam e essa cavitação pode ser a precursora de outros mecanismos de dano como a formação de “crazes” e “shear yielding” [Okamoto et al., 1993; Lovell et al., 1994].

Como boas referências teóricas sobre esse assunto, particularmente no campo dos polimetilmetacrilatos (PMMA), pode-se citar Hooley et al., 1981 e Laurienzo et al., 1996.

Alguns outros trabalhos que têm investigado o comportamento de materiais

poliméricos reforçados com uma segunda fase particulada convêm ser mencionados: Bittencourt e Cisilino, 2005, que estudam as propriedades da interface matriz-heterogeneidade e da fratura em um material composto por matriz polimérica de PMMA e heterogeneidades de borracha; Fasce, 2002, que investiga o comportamento mecânico de um polipropileno modificado com uma poliolefina elastomérica; Pettarin, 2002, que faz análises de informações de ensaios de impacto em materiais de matriz polimérica.

No Grupo de Mecânica Aplicada (GMAp) da Universidade Federal do Rio Grande do Sul (UFRGS), há alguns desenvolvimentos nessa área com aplicação do método dos elementos discretos (MED), entre eles: Batista et al., 2003, que faz uma homogeneização das propriedades mecânicas de um material microporoso; Batista et al., 2004, sobre dano em materiais microporosos; Tech et al., 2004, onde usam o método para estudar fratura estática e dinâmica; Galeano et al., 2005, onde se determina o EVR de um material composto particulado; Galeano et al., 2007, que apresenta um estudo do mecanismo de fratura por crescimento e coalescência de vazios em um material composto. Também cabe citar o trabalho de Batista, 2007, no qual faz uma aplicação do método dos elementos discretos na investigação do dano em compostos de matriz metálica.

3.2 Homogeneização linear elástica

Particularmente no campo da homogeneização, já conceituada anteriormente, esta tem sido amplamente utilizada na modelagem computacional para determinação das propriedades de materiais micro-heterogêneos.

No caso da homogeneização linear elástica, Terada et al., 2000, utilizando o critério de independência das condições de contorno proposto por Hill, estuda o tamanho do EVR de um compósito de boro-silício como função da resposta de uma constante elástica obtida a partir dos valores médios de tensão e deformação. Os volumes de controle foram gerados pelo método de processamento de imagens a partir da estrutura real. Pode-se observar nos resultados apresentados no gráfico da figura 2.24, que relativa independência das condições de contorno aplicadas é alcançada, para um tamanho de amostra aproximadamente 33 vezes maior do que o tamanho médio do micro-constituente.

De maneira similar ao procedimento seguido por Terada et al., 2000, Kanit et al., 2003, realiza um estudo de homogeneização das propriedades de um material micro-heterogêneo gerado de maneira artificial pelo método de tecelagem de Voronoi [Nygårds, 2003]. Os resultados obtidos da propriedade homogeneizada (módulo volumétrico, k) em

função do tamanho do volume de controle são apresentados na figura 3.1. Nesse caso, observa-se certa independência nas condições de contorno somente para um volume de controle aproximadamente oito mil vezes maior do que o volume da célula de Voronoi considerada. Apesar de, neste caso, terem sido comparados os volumes e não a dimensão da partícula em relação à dimensão do lado do volume de controle utilizado, como tinha sido realizado por Terada *et al.*, é interessante observar na figura 3.1 que tanto o critério de independência nas condições de contorno quanto o critério de homogeneidade estatística pode ser utilizado na determinação do EVR.

Cabe salientar que nos resultados apresentados por Kanit et al., 2003 (Figura 3.1) existe incoerência com a teoria, uma vez que os limites de Voigt e Reuss são ultrapassados pelos resultados obtidos na determinação do EVR, em muitos dos tamanhos de amostras analisadas. Isto se mostra incoerente, pois os limites de Voigt e Reuss são os valores extremos dentro dos quais poderiam ser encontradas as propriedades de qualquer mistura de fases. Pode ser que, na análise realizada por Kanit et al., 2003, em alguns dos casos não foram consideradas as propriedades mecânicas de um dos constituintes.

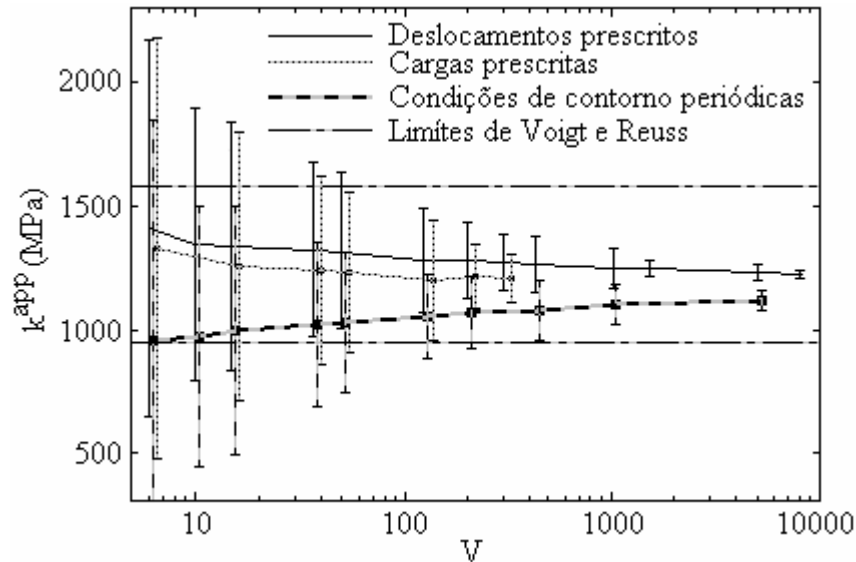


Figura 3.1 – Verificação da independência, das condições de contorno aplicadas, no valor do módulo volumétrico (k_c^{app}) obtido, com o incremento do volume de controle (V) [Kanit et al., 2003]

Apesar de não poderem ser comparados os tamanhos de EVRs obtidos nas figuras 2.24 e 3.1, por não ter sido utilizada a mesma referência dimensional, pode-se comentar que

Terada analisou um material constituído somente por duas fases, sendo que uma delas funcionava como matriz. No caso de Kanit, o material considerado estava constituído por múltiplas fases, e nenhuma funcionava como matriz. Estes fatos certamente levariam à obtenção de EVRs de tamanhos diferentes com a necessidade de maiores EVRs no caso analisado por Kanit por existir maior número de micro-constituintes.

Estas observações sugerem que, o tamanho do EVR depende das características do material analisado. Alguns dos estudos realizados neste sentido comprovaram que o tamanho do EVR vai depender de fatores tais como: grau de dispersão dos micro-constituintes e fração de volume dos mesmos [S. Graham, 2003; Kanit, 2003], contraste nas suas propriedades mecânicas e propriedade que se deseja homogeneizar [Kanit, 2003; Cailletaud, 2003], entre outros.

Teoricamente, a independência absoluta nas condições de contorno ou a homogeneidade estatística absoluta em um material micro-heterogêneo somente se torna possível para amostras de tamanho infinito [Aidun, 1999], o qual não teria sentido computacionalmente. Por esse motivo opta-se por trabalhar com EVRs em que é alcançada a independência nas condições de contorno ou a homogeneidade estatística das propriedades homogeneizadas dentro de tolerâncias pré-estabelecidas [Aidun, 1999; Kanit, 2003]. No caso de comportamento não-linear acentuado, estes critérios são ainda mais necessários [Kouznetsova, 2001].

Mais recentemente, Kari et al., 2007 aplicam o método dos elementos finitos para realizar uma homogeneização numérica, no qual determinam um elemento de volume representativo e avaliam as propriedades efetivas do material com condições de contorno periódicas. Foram homogeneizadas as constantes elásticas K , ν e G do composto, com distribuição aleatória dos constituintes esféricos, e os resultados foram comparados com aproximações analíticas da literatura, como as apresentadas no capítulo 2 deste trabalho, e os resultados foram bons. Foi investigada ainda a influência do tamanho das partículas esféricas e do tamanho do EVR nas propriedades efetivas do material.

Outras análises no mesmo campo de conhecimento vêm sendo feitas, como por exemplo, as de Trias et al., 2006, que fazem uma homogeneização estatística para estabelecer o tamanho do EVR em um polímero reforçado com fibras de carbono, as de Ren e Zheng, 2004, que analisam o efeito do tamanho, forma e distribuição dos grãos no tamanho do EVR de uma estrutura policristalina e as de Biwa et al., 2001, que avaliam as propriedades elásticas de partículas de borracha dispersas aleatoriamente em uma matriz de polimetilmetacrilato (PMMA).

3.3 Homogeneização elástica não-linear

Já no regime elástico não-linear, a homogeneização é um campo muito menos desenvolvido que no linear [Temizer e Zohdi, 2007]. Aproximações analíticas do problema envolvem construção de limites e conceitos de energia de deformação [Castañeda, 1989; Müller, 1987; Ogden, 1978; Talbot e Willis, 1985; Willis, 1994].

Temizer e Zohdi, 2007, apresentam procedimentos para homogeneização numérica elástica não-linear, em elementos finitos, de materiais heterogêneos baseados na determinação estatística de um EVR. Eles estendem os procedimentos da análise linear para o regime não-linear introduzindo um estado de tensão-deformação como parâmetro de caracterização e também o conceito de *material map*, que nada mais é do que um mapeamento discreto do material elástico não-linear que caracteriza o seu comportamento. O material é admitido como não-linear elástico, isotrópico e homogêneo, e as tensões existentes são apresentadas através do segundo tensor tensão de Piola-Kirchhoff, como já apresentado nas equações (2.65) a (2.69) e ilustrado pela figura (2.9). Com isso, um conjunto de α_i pode ser determinado para uma faixa particular de λ_F^i triaxiais, criando assim o mapeamento discreto do material. Temizer e Zohdi, 2007, mostram algumas implementações realizadas em modelos bidimensionais, onde as relações constitutivas assumem a forma

$$T_2 = \alpha_0 \cdot I + \alpha_1 \cdot E_L \quad (3.1)$$

que deriva da expressão (2.65), porém apresentando somente dois coeficientes, α_0 e α_1 .

Um material hiperelástico foi analisado, onde o segundo tensor tensão de Piola-Kirchhoff assume a forma

$$T_2 = \frac{\partial W}{\partial E} \quad (3.2)$$

sendo $W = \hat{W}(I_{E1}, I_{E2})$ a função energia de deformação. Para o modelo de Monney-Rivlin adotado, a função energia de deformação foi:

$$W = \frac{\mu}{2} (I_{C1} I_{C2}^{-1/2} - 2) + \frac{\kappa}{2} (I_{C2}^{1/2} - 1)^2 \quad (3.3)$$

Sendo:

C: o tensor deformação de Cauchy-Green a direita;

G e κ : os módulos de elasticidade transversal e de compressibilidade volumétrica, respectivamente.

Por fim, os autores apresentam uma forma de realizar a homogeneização de um material isotrópico, baseado na determinação de um EVR em que a equação constitutiva macroscópica tem a forma:

$$T_2^* = \alpha_0^* \cdot I + \alpha_1^* \cdot E^* + \alpha_2^* (E^*)^2 \quad (3.4)$$

onde $\alpha_i^* = \hat{\alpha}_i^*(I_{E^*}, II_{E^*}, III_{E^*})$ e E^* nesse contexto representam deformações.

Uma implementação similar a apresentada por Temizer e Zohdi, 2007, porém num caso tridimensional, será utilizada em uma das implementações a serem realizadas neste trabalho.

3.4 Otimização com Algoritmos Genéticos

A elaboração de estratégias para desenvolver uma ferramenta capaz de produzir variedades de materiais com propriedades “na medida” faz parte de um processo de otimização estrutural, no qual os Algoritmos Genéticos (AG) inserem-se como uma metodologia largamente empregada. Tal processo, no contexto dos materiais micro-heterogêneos, pode atuar sobre parâmetros como número de partículas, sua forma, distribuição, tamanho e fração volumétrica, orientações preferenciais, combinação de rigidez entre a matriz e o reforço, topologia dos reforços, etc, com a finalidade de satisfazer uma função objetivo, colocada em termos das propriedades macroscópicas do material. Em suma, a filosofia básica na construção destes materiais é selecionar combinações de micro-constituintes para produzir propriedades macroscópicas ou efetivas desejadas.

Uma utilização de AG no projeto de materiais micro-heterogêneos é proposta no trabalho de Zohdi, 2003, em que a função objetivo para o problema de otimização é colocada como a diferença entre as propriedades mecânicas efetivas desejadas (E_D^*, ν_D^*) e aquelas resultantes da microestrutura otimizada (E^*, ν^*). Como é assumido um comportamento isotrópico linear do material, o mesmo é completamente representado por duas constantes elásticas independentes, e a expressão para a função objetivo é escrita como:

$$\Pi = \left| \omega_E \frac{E^* - E_D^*}{E_D^*} \right| + \left| \omega_\nu \frac{\nu^* - \nu_D^*}{\nu_D^*} \right| \quad (3.5)$$

onde o sub-índice D denota a constante elástica “desejada”. Os coeficientes ω_E e ω_ν são fatores de peso que são estabelecidos pelo usuário em função da importância da correspondente propriedade que se quer projetar. E e ν são, respectivamente, o módulo de elasticidade longitudinal e o coeficiente de Poisson.

No âmbito do GMAp, na dissertação de Buroni, 2006, apresenta-se a implementação de algoritmos genéticos para o projeto de materiais micro-heterogêneos com distribuição aleatória, combinado com o Método de Busca Direta e o Método de Elementos de Contorno. É aplicado também ao estudo de aproximação da função objetivo visando a aceleração do algoritmo.

4 METODOLOGIA

Nesta seção, é feita uma breve descrição sobre algumas das metodologias empregadas nas implementações realizadas.

4.1 Homogeneização com distribuição aleatória dos micro-constituintes

Uma técnica de homogeneização comumente utilizada para obter propriedades efetivas de materiais compostos é aquela que considera a distribuição aleatória dos micro-constituintes dispersos na matriz. A mesma considera um EVR que reproduz tanto quanto possível a microestrutura do material. Para a obtenção do EVR, são resolvidos problemas de valores de contorno em volumes de controle do material micro-heterogêneo, incrementando gradualmente seu tamanho para incorporar na análise maior informação da micro-estrutura (Figura 4.1).

Dentro dos volumes de controle são verificadas as variáveis de campo e a partir da relação entre elas são obtidas as propriedades equivalentes E^* ou E^{*-1} , dependendo das condições de contorno utilizadas.

De maneira geral, a obtenção das componentes do tensor E^* ou E^{*-1} pode ser realizada, como propõe Zohdi, 2002, através da resolução de problemas de valores de contorno, onde são aplicadas condições de contorno de deslocamentos ou tensões, considerando seis casos de carregamento linearmente independentes como mostra a equação (4.1).

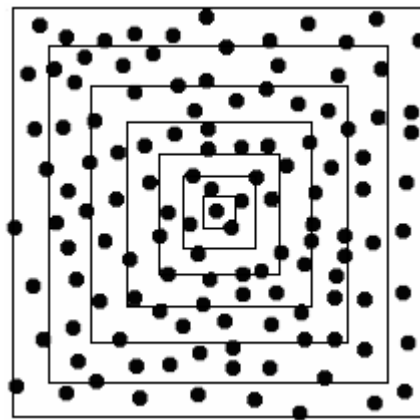


Figura 4.1 – Domínio de material micro-heterogêneo não periódico subdividido em volumes de controle

$$\varepsilon \text{ ou } p = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix}, \begin{bmatrix} 0 & \beta & 0 \\ \beta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \beta \\ 0 & \beta & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \beta \\ 0 & 0 & 0 \\ \beta & 0 & 0 \end{bmatrix} \quad (4.1)$$

Onde:

ε : Deslocamentos uniformes aplicados;

P : Tensões aplicadas;

β : variável que representa a carga aplicada (ou deslocamentos ou tensões).

Para cada caso de carregamento linearmente independente, verifica-se a influência nas outras direções, do qual se originam 6 equações. Considerando-se os seis casos de carregamento completa-se um total de 36 equações.

Para simplificar as análises podem ser aplicados alguns artifícios. Por exemplo, se for admitido comportamento isotrópico, somente é realizado um único caso de carregamento que permita a obtenção de um tensor de tensões de componentes hidrostáticas e desviadoras não nulas. Por outro lado, se o problema for analisado em duas dimensões, seriam necessários somente três casos de carregamento independentes.

Esta metodologia de trabalho é facilmente implementada no método dos elementos finitos e no método dos elementos discretos, utilizados nos estudos apresentados nos capítulos 5, 6 e 7.

4.2 Simulação da distribuição aleatória dos nódulos na matriz polimérica

A distribuição dos nódulos utilizada nos modelos também foi obtida através de uma rotina em Matlab, desenvolvida no INTEMA, Universidad de Mar del Plata, sob orientação do professor Adrian Cisilino, coorientador desta tese, que gera de forma automática distribuições aleatórias de nódulos, respeitando as condições de vizinhança características das distribuições de nódulos reais [Bermont e Cisilino, 2000].

4.3 Estudo de convergência da malha

A simulação do comportamento de materiais compostos através do método dos elementos finitos pode gerar resultados que se aproximam da solução exata do problema à medida que a malha é refinada [Temizer e Zohdi, 2007].

Por isso, malhas grosseiras podem gerar resultados que apresentam erros

significativos. Julgou-se necessário então, no presente caso, analisar a influência do refino da malha sobre as propriedades elásticas do composto, mais especificamente sobre o módulo de compressibilidade (K). Uma vez definida a malha ótima, procedeu-se a determinação do EVR.

Com esse objetivo, foi aplicado um estudo de Análise de Variância (*Anova*), para se verificar: a) a influência de uma só variável sobre o K do composto, utilizando-se a mesma malha para a matriz e o composto; b) a influência de dois fatores, e da interação entre eles, sobre o valor do K , com a definição de duas malhas distintas, uma para a matriz e outra para as heterogeneidades, com o objetivo de reduzir o tempo computacional (menor quantidade de elementos). Mais detalhes sobre Análise de Variância podem ser encontrados em Montgomery, 1991.

4.4 Otimização de modelos com Algoritmos Genéticos

Foram realizados estudos de otimização envolvendo as constantes elásticas K (módulo de compressibilidade) e G (módulo de elasticidade transversal), da matriz, das heterogeneidades e do composto, através da técnica dos Algoritmos Genéticos (AG), em elemento de volume representativo (EVR) de matriz polimérica termoplástica com heterogeneidades esféricas de elastômeros, cuja caracterização completa está no capítulo 6. A idéia global das etapas do algoritmo utilizado neste trabalho é:

- a) Foram codificadas as características de interesse em termos de cromossomos;
- b) Para $t=0$, avaliou-se uma população inicial $B_0 = \{b_{10}, b_{20}, \dots\}$ com uma certa diversidade;
- c) Repetiu-se, enquanto o critério de parada não fosse satisfeito:
 - seleção de indivíduos de B_t por probabilidade de sobrevivência do melhor adaptado, ou seja, quem possui maior função objetivo associada (Roulett Wheel);
 - geração de filhos do cruzamento destes indivíduos;
 - eventualmente se fez a mutação de alguns destes indivíduos;
 - avaliou-se a nova geração em função da geração antiga e dos filhos quanto à sua adaptação B_{t+1} .

Nesta implementação de algoritmos genéticos também foi incorporado o elitismo, isto é, os indivíduos de maior desempenho foram automaticamente selecionados. Evitou-se assim que os operadores genéticos pudessem realizar modificações neste indivíduo.

O critério de parada para o algoritmo foi o estado estacionário atingido pela diversidade. O fluxograma da figura 4.3 dá a idéia geral da implementação do algoritmo genético para esse estudo.

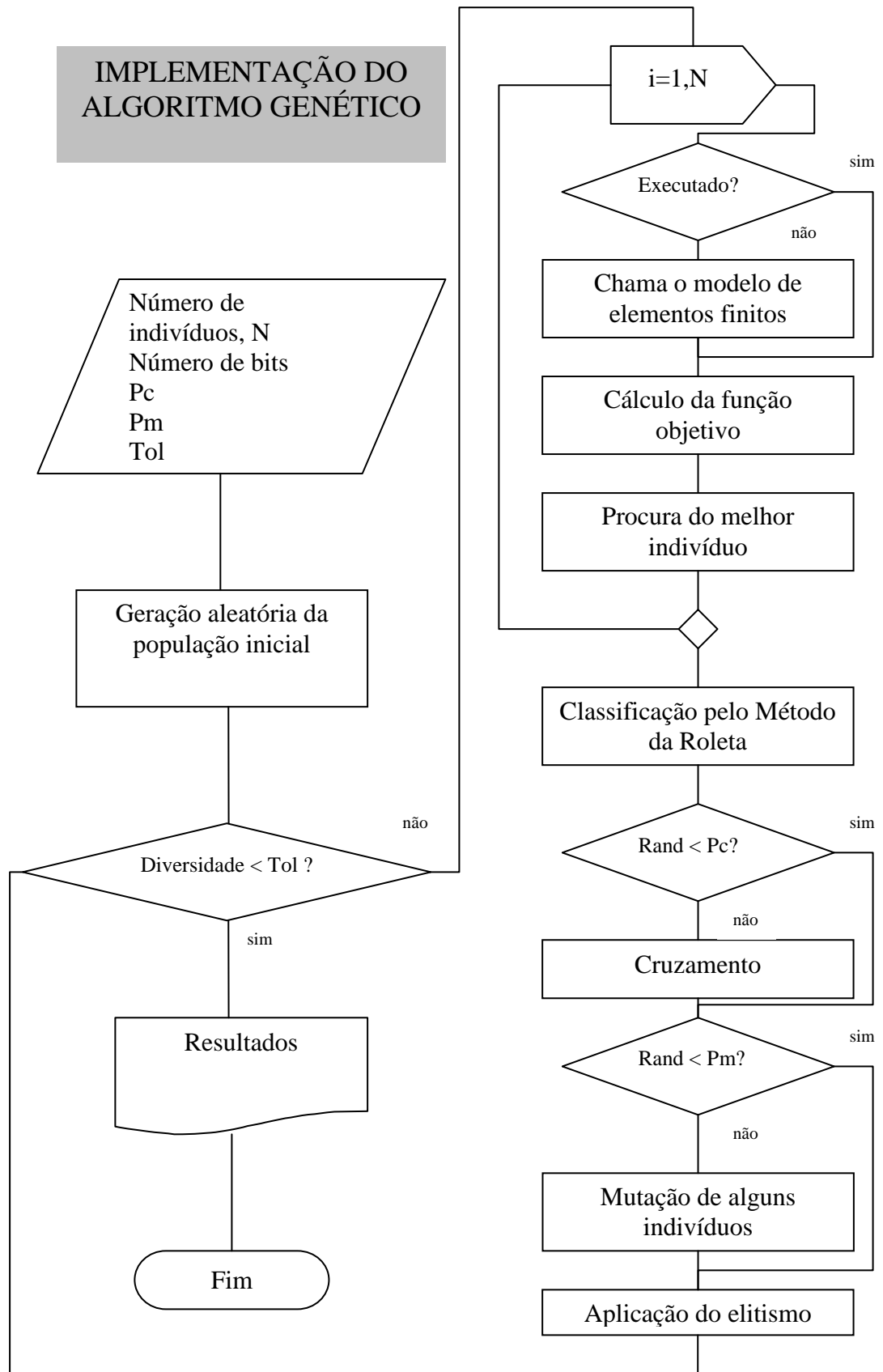


Figura 4.2 – Fluxograma da implementação do algoritmo genético

5 ESTUDOS REALIZADOS COM O MÉTODO DOS ELEMENTOS FINITOS

Neste capítulo visa-se:

a) Mostrar os resultados do modelamento numérico computacional que teve por objetivo aplicar as técnicas de homogeneização apresentadas em Suquet, 1985 e Zohdi, 2002 para determinar o elemento de volume representativo (EVR) de um material bifásico particulado onde a matriz é constituída de PMMA e a segunda fase representada por heterogeneidades esféricas similares à borracha com uma distribuição aleatória. O material composto resultante foi considerado elástico linear e também isotrópico, uma vez que, pelo fato da distribuição das partículas esféricas se darem de forma aleatória, considera-se que, na média, o material homogeneizado não apresente nenhuma tendência na distribuição das heterogeneidades, podendo assim ser considerado como isotrópico. Os parâmetros de análise foram as convergências dos valores das constantes elásticas K (módulo de compressibilidade) e G (módulo de elasticidade transversal). O estudo foi realizado utilizando-se o Método dos Elementos Finitos (MEF). Foram utilizadas cinco configurações aleatórias para caracterizar o composto com o valor médio e desvio da propriedade elástica estudada. Como os estudos de convergência foram realizados no sistema elástico linear, os resultados foram comparados com limites obtidos de desenvolvimentos teóricos clássicos da micro-mecânica, apresentados no item 2.3.1.4, fundamentações teóricas, que podem ser encontrados também em Zohdi, 2002, Gross e Seelig, 2006, Böhm, 1998, entre outros. Na seção 5.1 serão apresentados a caracterização do modelo, os procedimentos para determinação da discretização e do EVR, os modelos analíticos utilizados para comparações e os resultados obtidos com estes modelos lineares elásticos. No final da tese, no apêndice B, é apresentada a descrição do algoritmo utilizado para fazer este estudo de elementos finitos do caso linear elástico.

b) Aplicar as técnicas de homogeneização apresentadas em Temizer e Zohdi, 2007, para determinar o elemento de volume representativo (EVR) do material descrito no item a acima, porém agora com a matriz considerada elástica linear com a fase particulada, hiperelástica. Um diferencial em relação ao estudo realizado por Temizer e Zohdi, 2007, apresentado no presente trabalho, é a modelagem de um caso tridimensional. Os parâmetros de análise neste caso foram as convergências dos valores das constantes α_0 , α_1 e α_2 da equação (2.65). O estudo também foi realizado utilizando o Método dos Elementos Finitos

(MEF), com programação em linguagem Matlab (Matlab, 2009) e empregando o software comercial Abaqus/Simulia (Abaqus/Simulia, 2009). Foram utilizadas quatro configurações aleatórias para caracterizar estatisticamente o composto com o valor médio e desvio dos parâmetros estudados. Cabe salientar aqui que nos modelos de 32 e 64 heterogeneidades foram analisadas, respectivamente, duas e uma amostras. Justifica-se esta simplificação pelo reduzido coeficiente de variação observado no estudo a partir dos modelos com 16 heterogeneidades, como se poderá observar nos gráficos na seção 5.2.5 mais adiante. Ao final, no apêndice C, é apresentada a descrição do algoritmo utilizado para fazer este estudo de elementos finitos do caso elástico não-linear. Após a determinação do EVR, foram desenvolvidos os seguintes modelos como forma de aplicação e verificação da eficácia do estudo realizado:

Testes de tração uniaxial e corte puro

Para verificar o cálculo do EVR, realizou-se, para casos de solicitações simples (tração uniaxial e corte puro), o cálculo da curva tensão deformação obtida modelando no Abaqus/Simulia um elemento de volume representativo (número determinado de heterogeneidades inseridas na matriz). Compararam-se esses resultados com os obtidos aplicando a expressão 2.65 com os valores de alfas extraídos dos mapas médios obtidos para o EVR.

Na seqüência são apresentadas as implementações supracitadas nos pontos *a* e *b*.

5.1 Homogeneização numérica linear elástica do composto com distribuição aleatória de heterogeneidades esféricas

5.1.1 Caracterização do modelo

A micro-estrutura estudada é um material composto formado por uma matriz cúbica elástica de PMMA com heterogeneidades esféricas similares a um elastômero. O cubo possui módulo de elasticidade de Young $E = 3240\text{MPa}$ e coeficiente de Poisson $\nu = 0,25$ enquanto as esferas $\nu = 0,40$ e $E = 40\text{MPa}$. Como uma primeira aproximação, as heterogeneidades foram consideradas de material elástico linear. O raio das mesmas foi mantido constante em $r = 22,8 \times 10^{-6}\text{m}$ (este valor será justificado adiante na seção 5.2.1 onde se apresenta e se

justifica uma transformação de unidades) e o tamanho da matriz cúbica variável à medida que nela eram adicionados nódulos, de forma a manter constante a fração volumétrica de heterogeneidades em relação ao volume da matriz. A distribuição das posições das esferas na matriz foi dada de forma aleatória fixando a distância mínima entre as partículas em $0,04r$ e a distância mínima delas às bordas externas da matriz em $0,01r$ onde r é o raio do nódulo. Mais detalhes sobre a geração aleatória da distribuição das partículas podem ser encontrados em Ortiz et al., 2001. Nesta fase, pequenas adaptações foram realizadas no programa, em Matlab, que realiza esta geração aleatória para que se adaptasse à finalidade proposta nesse trabalho. O sistema foi modelado com uso do software comercial de elementos finitos Abaqus/Simulia (Abaqus/Simulia, 2009) considerando o material elástico linear. A malha utilizada foi formada por elementos finitos tetraédricos de 4 nós e foi considerada uma união perfeita na interface partícula/matriz sem permitir a perda de aderência entre elas. Anterior ao procedimento de determinação do EVR foi realizado um estudo de convergência para determinar a discretização a ser empregada nos estudos a serem realizados. Nesse estudo, mostrado no item seguinte, faz-se referência aos módulos K e G , cujos procedimentos para seus cálculos são apresentados adiante na seção 5.1.3 e no final da tese no apêndice B.

5.1.2 Estudo de convergência

Para realizar o estudo de convergência da malha, utilizou-se o modelo anteriormente descrito, com duas heterogeneidades esféricas iguais, representando 10% do volume da amostra. Foi utilizado um modelo elástico linear e se considerou distribuições aleatórias das heterogeneidades dentro da matriz cúbica para poder assumir que o composto fosse isotrópico. A Figura 5.1 mostra uma das distribuições possíveis para o composto analisado.

Primeiramente, foi utilizada a mesma malha para a matriz e para as heterogeneidades. Assim, foi estudada a influência de uma só variável sobre o módulo de compressibilidade K do composto. Posteriormente, procurando um menor tempo computacional (menor quantidade de elementos), definiram-se duas malhas distintas: uma para a matriz e outra para as heterogeneidades. Nesse caso, foram analisadas as influências de dois fatores, e da interação entre eles, sobre o valor de K .

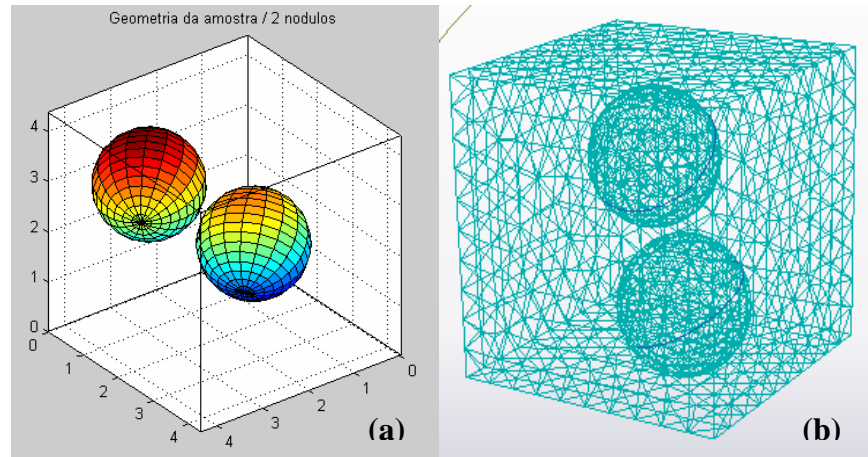


Figura 5.1 – Exemplo de distribuição do composto (a); Exemplo de malha de elementos finitos para o composto (b)

5.1.2.1 Matriz e heterogeneidades com malhas iguais

Foram definidos nove tamanhos distintos de malha, analisando-se 5 distribuições aleatórias para cada um. Na tabela 5.1 estão dispostos os resultados do módulo de compressibilidade.

Tabela 5.1 – Resultados do módulo de compressibilidade para diferentes razões tamanho da malha máximo/raio da heterogeneidade, considerando a mesma malha para a matriz e heterogeneidades

	Tamanho da malha máximo/raio da heterogeneidade								
	0,1	0,15	0,2	0,3	0,4	0,6	0,8	1,2	1,6
K [MPa]	1764,40	1771,15	1772,63	1804,97	1798,66	1858,75	1849,92	1879,03	1877,72
	1779,50	1770,29	1787,13	1785,24	1796,17	1864,87	1862,11	1856,83	1869,50
	1767,38	1792,65	1775,35	1802,67	1802,48	1857,78	1852,26	1858,80	1861,79
	1777,68	1785,79	1785,47	1792,32	1819,42	1862,64	1866,80	1869,74	1873,90
	1784,74	1769,46	1792,03	1784,16	1820,22	1862,55	1857,92	1863,90	1864,82
Médias	1774,74	1777,87	1782,52	1793,87	1807,39	1861,32	1857,80	1865,66	1869,55

Fazendo a análise de variância para verificar a influência da malha sobre o valor de K do composto (*One-Way Analysis of Variance*) obtiveram-se os resultados mostrados na tabela 5.2, onde

SQ: é a soma quadrática;

Gl: são os graus de liberdade;

MQ: é a média quadrática;

F: teste F;

valor -P: probabilidade ligada ao teste F;

F crítico: valor de probabilidade ligado à sensibilidade.

Comparando o valor de F calculado (116,91) com o valor de F crítico (2,21), fica evidenciado que pelo menos uma das médias obtidas entre os diversos grupos de malha difere de alguma outra.

Tabela 5.2 – Análise de Variância considerando o efeito de um fator (mesma malha para matriz e heterogeneidades)

<i>Fonte da variação</i>	<i>SQ</i>	<i>Gl</i>	<i>MQ</i>	<i>F</i>	<i>valor-P</i>	<i>F crítico</i>
Entre grupos	68664,52	8	8583,07	116,91	$2,08 \times 10^{-23}$	2,21
Dentro dos grupos	2643,04	36	73,42			
Total	71307,56	44				

Comparação múltipla de médias

Para verificar se duas médias não pertencem à mesma curva de distribuição de probabilidade, a diferença entre elas deve ser maior do que o chamado Limite de Decisão L_d , que é definido com três desvios padrões dos dados experimentais:

$$L_d = 3\sigma \cong 3 \frac{\sqrt{MQR}}{\sqrt{n}} = 3 \frac{\sqrt{73,42}}{\sqrt{5}} = 11,50 \text{ MPa}$$

onde n é o número de repetições do experimento dentro de um determinado grupo.

Comparando todas as diferenças das médias com o limite de decisão, obteve-se o gráfico da figura 5.2. Na figura, as linhas sobre as barras indicam que as barras sob ela representam médias que pertencem à mesma distribuição de probabilidade.

Como comentado anteriormente, uma malha mais refinada leva a um resultado mais próximo da solução exata. Analisando o gráfico da figura 5.2, pode-se escolher a malha com a razão entre tamanho máximo da malha e raio da heterogeneidade igual a 0,2 como a ideal para os valores estabelecidos. Essa é a malha que tem menor número de elementos (menor

custo computacional) quando comparada com as malhas de razão 0,1 e 0,15, as quais fornecem resultados equivalentes.

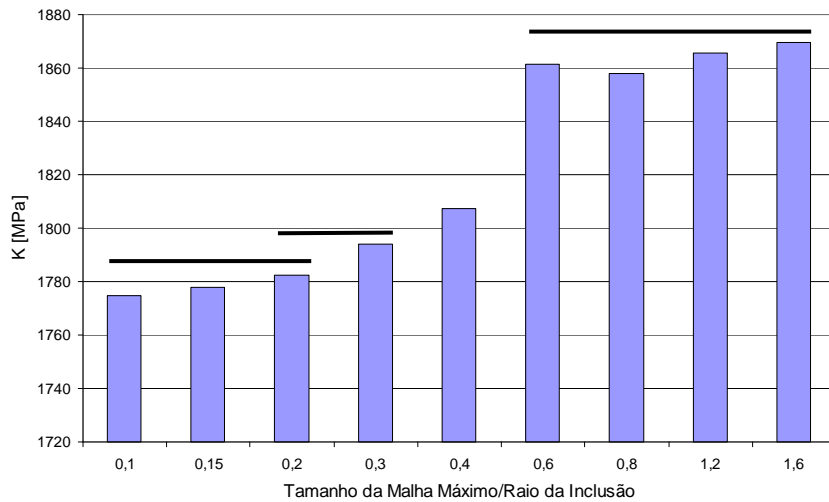


Figura 5.2 – Gráfico de barras indicando as médias pertencentes à mesma distribuição de probabilidade

5.1.2.2 Matriz e heterogeneidades com malhas diferentes

Nesse caso, tanto a malha da matriz como a das heterogeneidades podem variar de forma independente. Desta maneira, têm-se dois fatores que podem influenciar o resultado, cada um dos quais considerados com quatro níveis de variação (razão entre tamanho da malha máximo e raio da heterogeneidade igual a 0,2; 0,4; 0,8 ou 1,6).

Na tabela 5.3, estão dispostos os resultados do módulo de compressibilidade, com cinco réplicas de experimentos para cada combinação entre os dois fatores de influência, e na tabela 5.4, apresenta-se um resumo da tabela 5.3, onde são mostradas as médias dentro dos grupos (entre as réplicas) e as médias entre grupos. O nível de significância no estudo foi de 5%.

Fazendo a análise de variância para se verificar a influência de cada uma das malhas e da interação entre elas sobre o valor de K do composto (*Two-Way Analysis of Variance*), obtêm-se os resultados mostrados na tabela 5.5.

Tabela 5.3 – Resultados de módulo de compressibilidade para diferentes razões tamanho da malha máximo/raio da heterogeneidade, considerando malhas diferentes para a matriz e heterogeneidades

		Razão da Malha da Matriz			
		0,2	0,4	0,8	1,6
Razão da Malha das Heterogeneidades	0,2	1772,63	1800,82	1870,29	1877,29
		1787,13	1798,70	1858,87	1869,19
		1775,35	1804,09	1851,66	1864,47
		1785,47	1812,78	1854,38	1865,52
		1792,03	1799,86	1857,74	1840,30
	0,4	1770,98	1816,81	1862,13	1867,92
		1769,20	1798,73	1869,24	1874,74
		1775,10	1814,30	1859,75	1867,72
		1795,54	1806,22	1865,91	1875,65
		1795,15	1797,47	1869,18	1876,91
	0,8	1773,94	1796,67	1853,91	1866,01
		1784,87	1808,87	1854,36	1865,77
		1772,40	1803,50	1858,55	1865,36
		1798,27	1806,79	1870,02	1877,14
		1769,84	1807,10	1871,07	1877,38
	1,6	1795,46	1822,84	1849,92	1857,23
		1788,57	1795,30	1862,13	1872,60
		1769,18	1794,62	1852,26	1858,88
		1792,69	1811,46	1866,80	1873,66
		1776,70	1802,06	1857,92	1865,76

Tabela 5.4 – Resumo da Tabela 5.3

K [Mpa]		Razão da Malha da Matriz				
		0,2	0,4	0,8	1,6	Médias
Razão da Malha das Heterogeneidades	0,2	1782,19	1806,45	1856,96	1861,48	1826,77
	0,4	1781,79	1802,68	1863,60	1872,21	1830,07
	0,8	1784,17	1809,82	1860,78	1868,58	1830,84
	1,6	1781,79	1800,86	1859,78	1867,73	1827,54
	Médias	1782,48	1804,95	1860,28	1867,50	

Tabela 5.5 – Análise de Variância considerando o efeito de dois fatores e a interação entre os mesmos (malhas diferentes para matriz e heterogeneidades)

<i>Fonte da variação</i>	<i>SQ</i>	<i>Gl</i>	<i>MQ</i>	<i>F</i>	<i>valor-P</i>	<i>F crítico</i>
Malha da Heterogeneidade	213,77	3	71,26	0,895	0,448613	2,748
Malha do Cubo	106312,80	3	35437,59	445,186	$8,44 \times 10^{-43}$	2,748
Interação	315,92	9	35,10	0,441	0,907604	2,030
Dentro dos Grupos	5094,52	64	79,60			
Total	111937	79				

Comparando os valores de F calculados com os valores críticos correspondentes, constata-se que apenas o fator malha da matriz tem influência sobre o resultado de K .

Como não se verifica efeito principal da malha da heterogeneidade nem efeito da interação entre as duas malhas, apenas o gráfico do efeito principal da malha da matriz é apresentado (Figura 5.3).

Comparação múltipla de médias

Seguindo a mesma metodologia apresentada anteriormente, o limite de decisão vale $11,97 MPa$, e se pôde identificar que os níveis de malha do cubo de 0,8 e de 1,6 pertencem a mesma distribuição aleatória, enquanto os outros níveis diferem.

Para uma maior precisão de resultados na análise de elementos finitos (malha mais refinada), a partir do gráfico da figura 5.3, pode-se escolher a malha da matriz com razão 0,2.

Para os níveis da malha das heterogeneidades, não há sentido fazer a comparação múltipla de médias, uma vez que o teste F mostrou que não há diferença significativa entre elas.

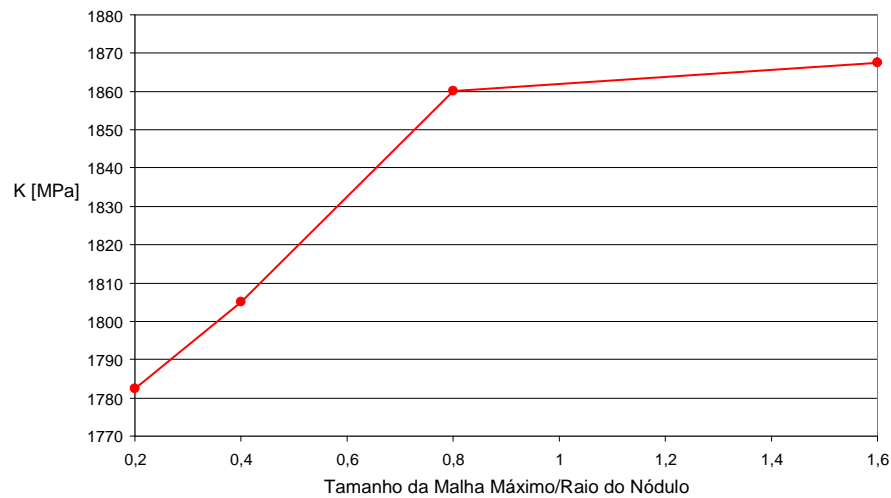


Figura 5.3 – Efeito principal da malha da matriz

5.1.2.3 Escolha da malha

As figuras 5.4 e 5.5 mostram a variação das constantes elásticas K e G em relação ao número de heterogeneidades para as diferentes malhas analisadas. N e M são as variáveis de discretização utilizadas que representam os valores da razão entre o tamanho máximo da malha e o raio das heterogeneidades para os nódulos e a matriz, respectivamente.

Procurando uma maior exatidão na análise de elementos finitos, pode-se optar por uma malha de razão entre tamanho da malha máximo e raio das heterogeneidades de 0,3 com um erro pequeno, para todo o composto (Figuras 5.4 e 5.5), já que o resultado não apresentou diferenças significativas quando se usaram malhas mais refinadas.

Também desejando minimizar o custo computacional (menor quantidade de elementos), podem-se definir duas malhas diferentes (para o cubo e para as heterogeneidades).

Uma vez que se verificou que a malha das heterogeneidades não tem influência significativa sobre o resultado, pode-se optar pela malha mais grosseira possível para elas.

Diante disso, considerando os fatores exatidão e tempo computacional, optou-se pela utilização de uma malha com razão de 0,3 para o cubo e de 1,5 para as heterogeneidades na determinação do EVR (Figuras 5.4 e 5.5). Tais figuras deixam claro que: a) a malha das heterogeneidades não exerce influência significativa nos resultados; b) malhas mais refinadas geram curvas mais abaixo, conforme já mencionado no item 4.3 e ilustrado pela figura 4.2.

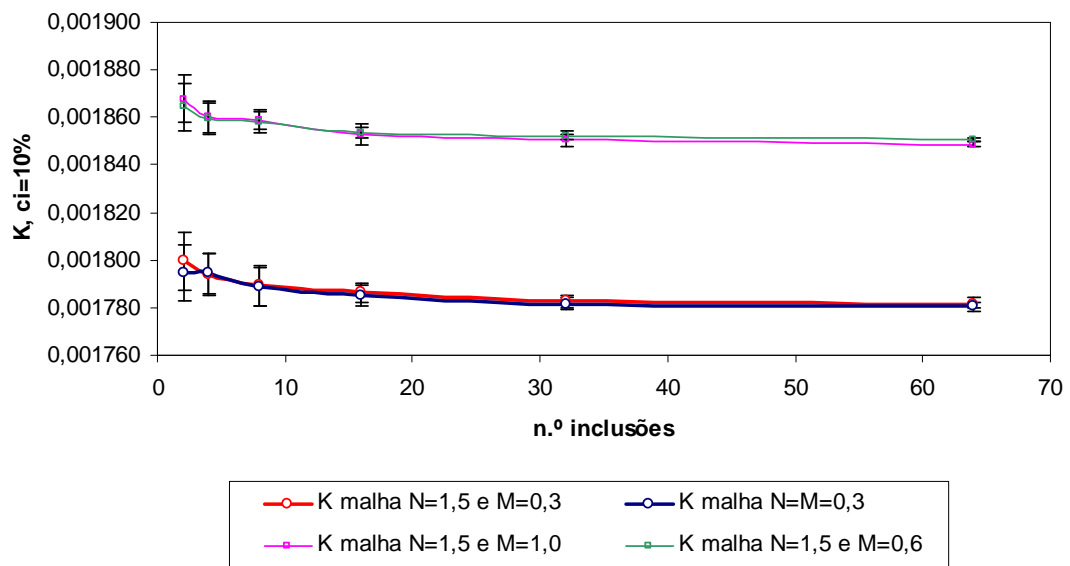


Figura 5.4 – Resultados de $K[N/\mu\text{m}^2]$ para diferentes refinamentos (N =razão da malha dos nódulos e M =razão da malha da matriz). As curvas abaixo são das malhas mais refinadas

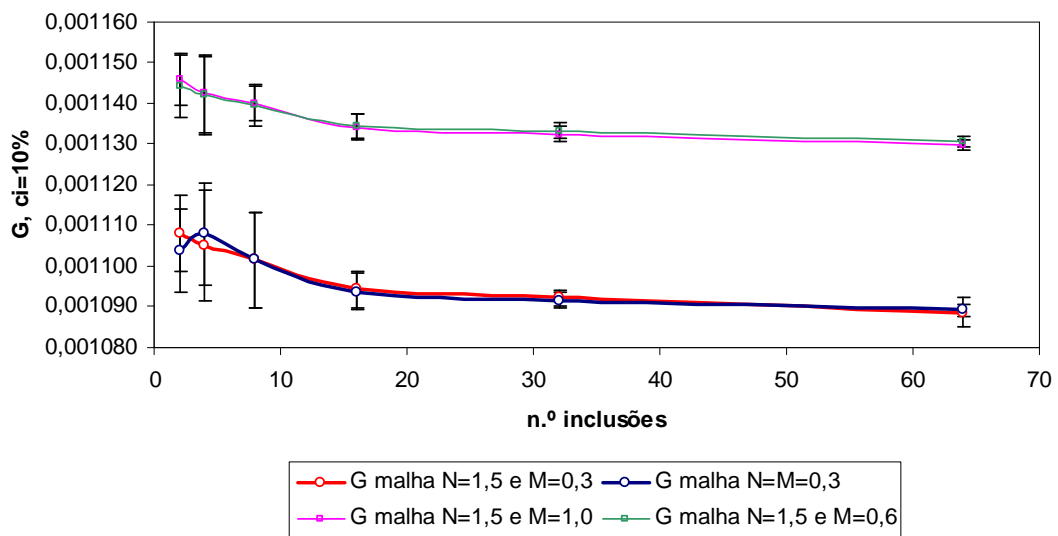


Figura 5.5 – Resultados de $G [N/\mu\text{m}^2]$ para diferentes refinamentos (N =razão da malha dos nódulos e M =razão da malha da matriz). As curvas abaixo são das malhas mais refinadas

5.1.3 Determinação do elemento de volume representativo (EVR)

Como visto anteriormente, um elemento de volume representativo (EVR) é uma amostra suficientemente pequena para ser considerada como um ponto material do domínio analisado, mas suficientemente grande para conter uma mostra estatisticamente representativa da microestrutura.

Não há um procedimento padronizado para determinar em forma direta as dimensões

e número de heterogeneidades de um EVR. Por isso suas dimensões são calculadas em cada caso de forma numérica através do sucessivo incremento do volume da amostra. O tamanho de amostra representativo será aquele que proporcione uma resposta macroscópica invariante (calculada como relação entre médias) para distintas distribuições das partículas. Um critério comumente aceito para realizar o cálculo de propriedades efetivas e definir o EVR é a condição de Hill, que pode ser encontrada em Zohdi, 2002:

$$\langle \sigma : \varepsilon \rangle_{\Omega} = \langle \sigma \rangle_{\Omega} : \langle \varepsilon \rangle_{\Omega} \quad (5.1)$$

com a relação entre médias da forma

$$\langle \sigma \rangle_{\Omega} = C^* \langle \varepsilon \rangle_{\Omega} \quad (5.2)$$

Onde:

σ e ε : são os tensores de tensão e deformação respectivamente;

C^* : é a matriz com as constantes elásticas do material;

Ω : é o volume de controle (EVR).

Nas expressões (5.1) e (5.2), e nas seguintes, onde houver, o símbolo $\langle . \rangle$ indica o cálculo do valor médio da magnitude indicada calculada componente a componente. Há dois importantes estados de carga que satisfazem a condição de Hill, a saber: os teoremas da deformação média e da tensão média. No primeiro caso a amostra é submetida a uma aplicação linear de deslocamentos da forma

$$u|_{\partial\Omega} = \varepsilon . x \Rightarrow \langle \varepsilon \rangle_{\Omega} = \varepsilon \quad (5.3)$$

enquanto que no segundo são aplicadas trações constantes no contorno, na forma

$$t|_{\partial\Omega} = \zeta . n \Rightarrow \langle \sigma \rangle_{\Omega} = \zeta \quad (5.4)$$

onde ε e ζ são tensores constantes de deformação e tensão respectivamente. Cabe salientar que, no caso de ser possível acontecer descolamento na interface matriz/nódulos, não é possível aplicar o teorema da deformação média, ou seja, com descolamento não se pode aplicar deslocamentos prescritos. Como a distribuição das partículas é aleatória, aplicar tensões prescritas pode ser não trivial. Visto que nos estudos em questão não é prevista a perda de aderência entre matriz e partículas, aplicou-se deslocamentos prescritos como

condições de contorno externas. A figura 5.6 mostra um modelo utilizado neste estudo, em que são aplicados deslocamentos prescritos com $\varepsilon=0,001$.

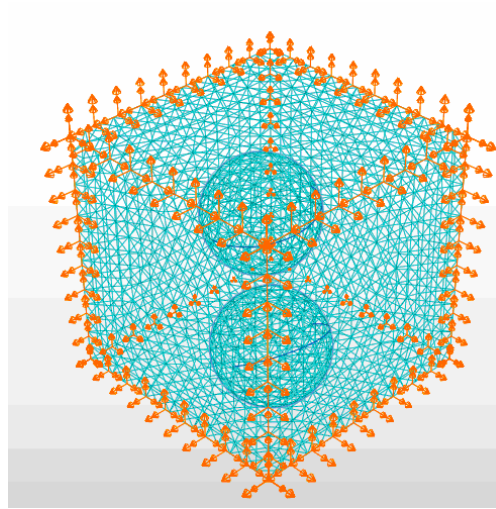


Figura 5.6 – Esquema mostrando malha, condições de contorno e distribuição aleatória de 2 nódulos esféricos de uma amostra

Como indica Zohdi, 2002, para determinação da matriz constitutiva homogeneizada C^* há, em princípio, seis tipos de deformação, como mostrado na equação (4.1).

Isso tornaria necessária a resolução das equações (4.1) para os seis casos. No entanto, como está se assumindo um material isótropo, Malvern, 1968 mostra que se pode aplicar apenas um tipo de carregamento para caracterizar a resposta do material, desde que esse carregamento contenha componentes dilatacionais

($\frac{tr \sigma}{3} = \frac{\sigma_{11} + \sigma_{22} + \sigma_{33}}{3}$ e $\frac{tr \varepsilon}{3} = \frac{\varepsilon_{11} + \varepsilon_{22} + \varepsilon_{33}}{3}$) e desviadoras:

$$\sigma' = \sigma - \frac{tr \sigma}{3} I = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} - \frac{tr \sigma}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad e$$

$$\varepsilon' = \varepsilon - \frac{tr \varepsilon}{3} I = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} - \frac{tr \varepsilon}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

não nulas. Desta forma as constantes elásticas podem ser obtidas através das equações

$$3K^* = \frac{\langle \frac{tr \sigma}{3} \rangle_{\Omega}}{\langle \frac{tr \varepsilon}{3} \rangle_{\Omega}} \quad e \quad 2G^* = \sqrt{\frac{\langle \sigma' \rangle_{\Omega} : \langle \sigma' \rangle_{\Omega}}{\langle \varepsilon' \rangle_{\Omega} : \langle \varepsilon' \rangle_{\Omega}}} \quad (5.5)$$

Os valores da expressão (5.5) foram calculados pós-processando os resultados obtidos pela análise de elementos finitos. Detalhes dos programas desenvolvidos para realizar esta análise poder ser vistos no apêndice B desta tese. O processo para determinação do elemento de volume representativo, consistiu em analisar amostras da micro-estrutura do material com diferentes quantidades de micro-heterogeneidades (no presente caso, nódulos esféricos), observando seu comportamento. O tamanho do EVR foi o da amostra que conteve o número mínimo de partículas em que a resposta do material se manteve constante. Neste estudo foram analisados seis tamanhos de amostras contendo 2, 4, 8, 16, 32 e 64 nódulos. Para cada um desses casos foram estudadas cinco configurações com posições aleatórias das esferas e considerada a média e dispersão dos resultados (K^* , G^*) em cada caso. Para se chegar ao EVR foi analisado o comportamento destas constantes elásticas diante do incremento do número de nódulos inseridos na matriz. Todo o processo foi implementado na linguagem de programação Matlab utilizando-se uma interface com o software de elementos finitos Abaqus/Simulia para gerar as malhas aleatórias, aplicar as condições de contorno, gerar a saída de tensões e deformações e aplicar as expressões (5.5). De posse da amostra considerada como elemento de volume representativo, seus resultados foram comparados com modelos analíticos da bibliografia para diferentes frações de volume de nódulos em relação ao volume da amostra. Também foi estudada a variação do EVR em função da fração de volume de partículas do composto em a influência da distância mínima entre as partículas nos valores das constantes elásticas K e G . A seguir são apresentados os resultados obtidos nesta etapa do trabalho.

5.1.4 Resultados e discussões da etapa

5.1.4.1 Convergência dos parâmetros K e G

As figuras 5.7 e 5.8 mostram a convergência da resposta do material (parâmetros K e G) com o incremento do número de nódulos para amostras com volume de nódulos de 10% em relação ao tamanho do modelo. K' é o valor de K para 64 nódulos e G' é o valor de G para 64 nódulos. Pode-se notar também a redução da dispersão dos resultados, indicada pelas barras de erros nos gráficos. Nesse estudo fica claro que as velocidades de convergência dos parâmetros K e G diferem pois, se um erro de 1% em relação à maior amostra (64 nódulos)

for admitido, estes resultados permitem afirmar que no regime linear elástico uma amostra com 2 heterogeneidades para uma $c_I = 10\%$ já é representativa, se o critério considerado for somente o K . No entanto, a análise dos valores de G mostra que, para esse erro admissível, a amostra com 10 heterogeneidades é a representativa da micro-estrutura (EVR). Portanto, seu comportamento macroscópico é representativo. Observando a figura 5.8, fica claro que após 10 nódulos os gradientes de mudança nos resultados são sensivelmente menores. Pode-se, como trabalho futuro, aprofundar esta questão verificando os fatores que influenciam na diferença da velocidade de convergência destes parâmetros elásticos.

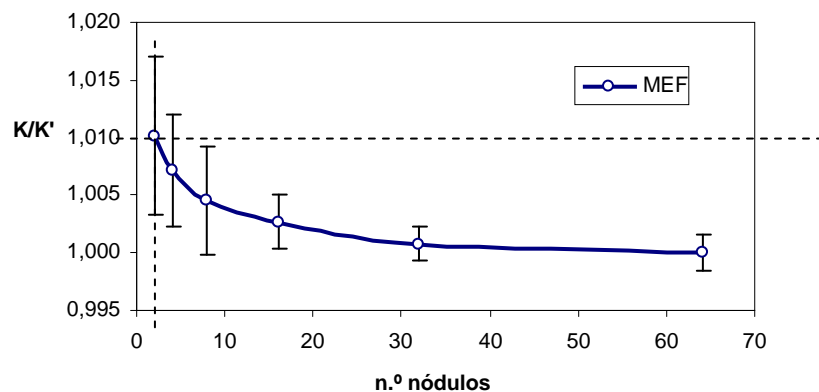


Figura 5.7 – Comportamento do módulo de compressibilidade K com o incremento do número de nódulos para um volume de nódulos de 10% em relação ao volume da amostra

5.1.4.2 Influência da fração de volume de heterogeneidades na determinação do EVR

Foi analisada também a influência da fração do volume de nódulos em relação ao volume da matriz na determinação do EVR. A figura 5.9 ilustra EVRs considerando diferentes frações de volume (5%, 10% e 20%) e erros admissíveis (0,5%, 1% e 1,5%). Pode-se observar que fica clara a influência exercida pela fração de volume de nódulos no composto na determinação do elemento de volume representativo, tanto com relação ao módulo K quanto ao G .

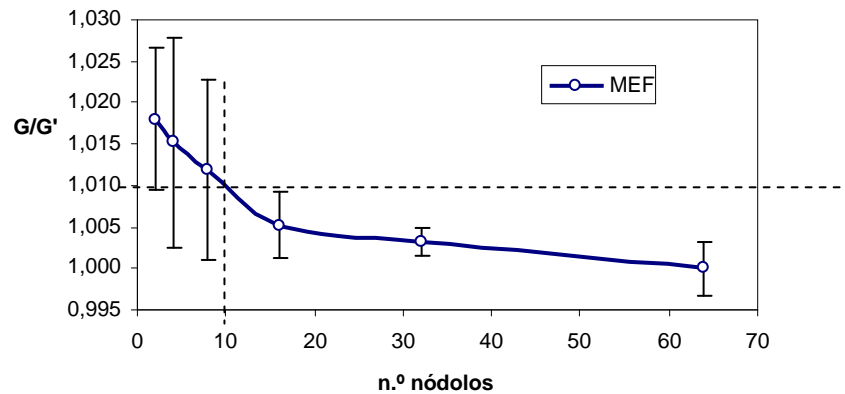


Figura 5.8 – Comportamento do módulo de elasticidade transversal G com o incremento do número de nódulos para um volume de nódulos de 10% em relação ao volume da amostra

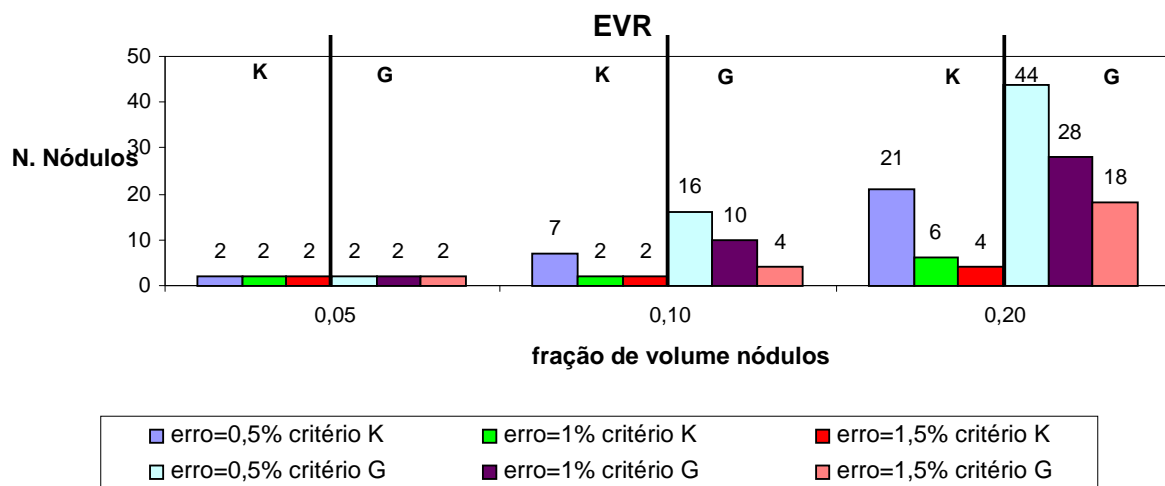


Figura 5.9 – Comportamento do EVR para diferentes frações de volume de heterogeneidades. Foram analisados os valores de K e G com erros admissíveis de 0,5%, 1% e 1,5% em relação à maior amostra (64 nódulos)

Pode-se observar que pelo critério do G , o mais desfavorável, tem-se, para um erro de 1%, o EVR de 2 heterogeneidades para $c_I = 5\%$, de 10 heterogeneidades para $c_I = 10\%$ e de 28 heterogeneidades para $c_I = 20\%$. A figura 5.10 ilustra esse fato.

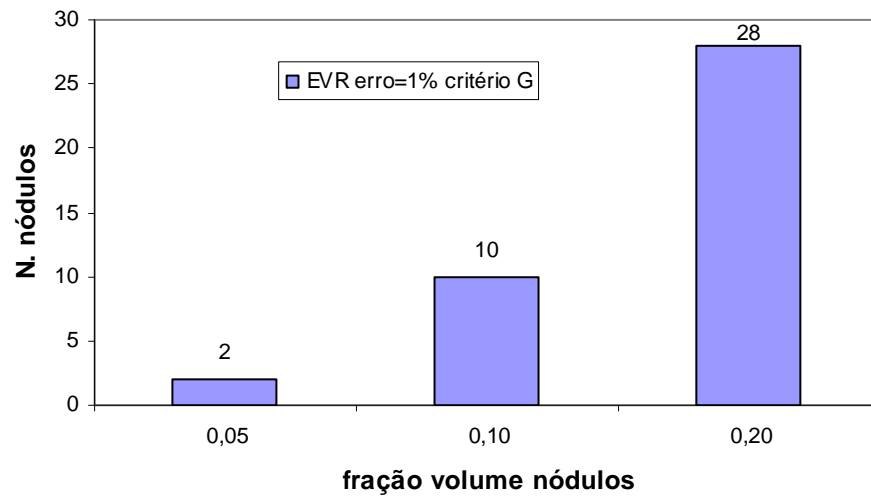


Figura 5.10 – Comportamento do EVR para diferentes frações de volume de heterogeneidades, com erro=1% e critério G

5.1.4.3 Comparação dos resultados obtidos do MEF com modelos analíticos

As figuras 5.11, 5.12, 5.13 e 5.14 apresentam a comparação dos resultados obtidos com elementos finitos (MEF) e com os modelos teóricos Voigt, Reuss, DILUTE, Mori-Tanaka, Auto-consistente e Diferencial, para as amostras consideradas representativas (2, 10 e 28 nódulos) analisando diferentes frações de volume de nódulos em relação ao volume da amostra. O sistema do esquema diferencial (Equação 2.101) foi resolvido numericamente através do Método de Runge-Kutta [Forsythe et al.,1977], aplicando-se uma variação de 0,05 (5%) a 0,20 (20%) para a fração volumétrica c_1 .

Observa-se que, em todos os casos, os limites de Voigt e Reuss foram respeitados. Na comparação dos modelos em MEF com os demais modelos pode-se observar uma boa aproximação dos valores, especialmente para as frações de volume de nódulos menores. À medida que se aumentou esta fração de volume os valores do modelo DILUTE tenderam a se afastar dos demais como era de se esperar, já que este modelo não leva em conta a interação entre heterogeneidades.

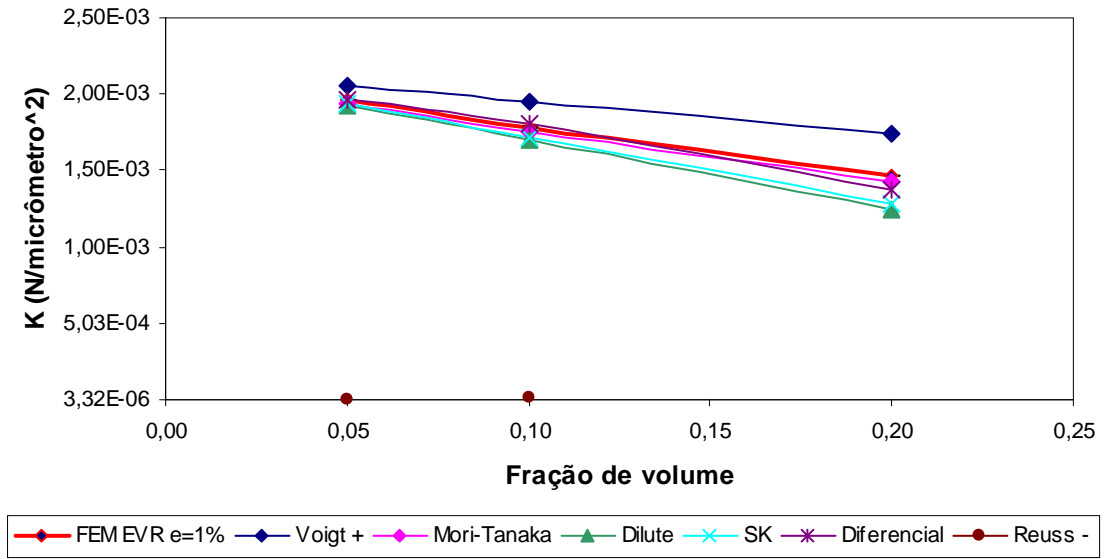


Figura 5.11 – Comparação do módulo de compressibilidade K obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos

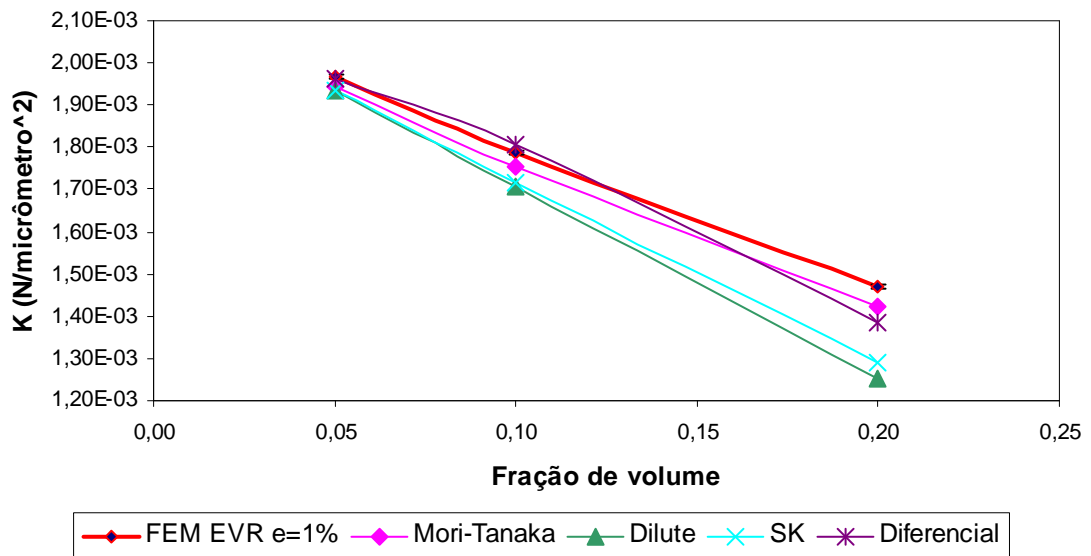


Figura 5.12 – Detalhe sem os limites de Voigt e Reuss da comparação do módulo de compressibilidade K obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos

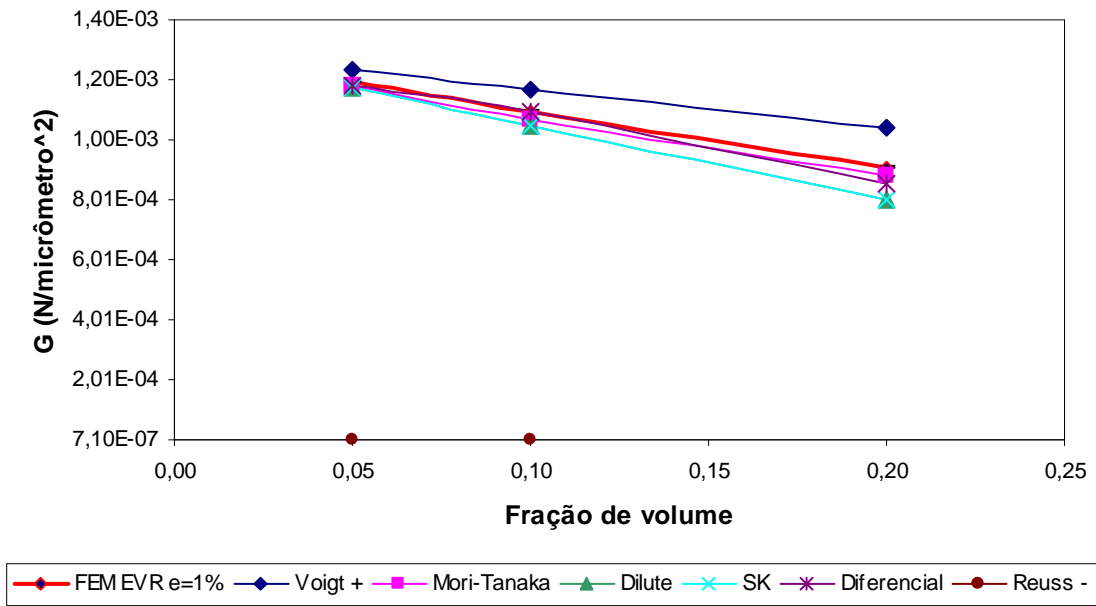


Figura 5.13 – Comparação do módulo de elasticidade transversal G obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos

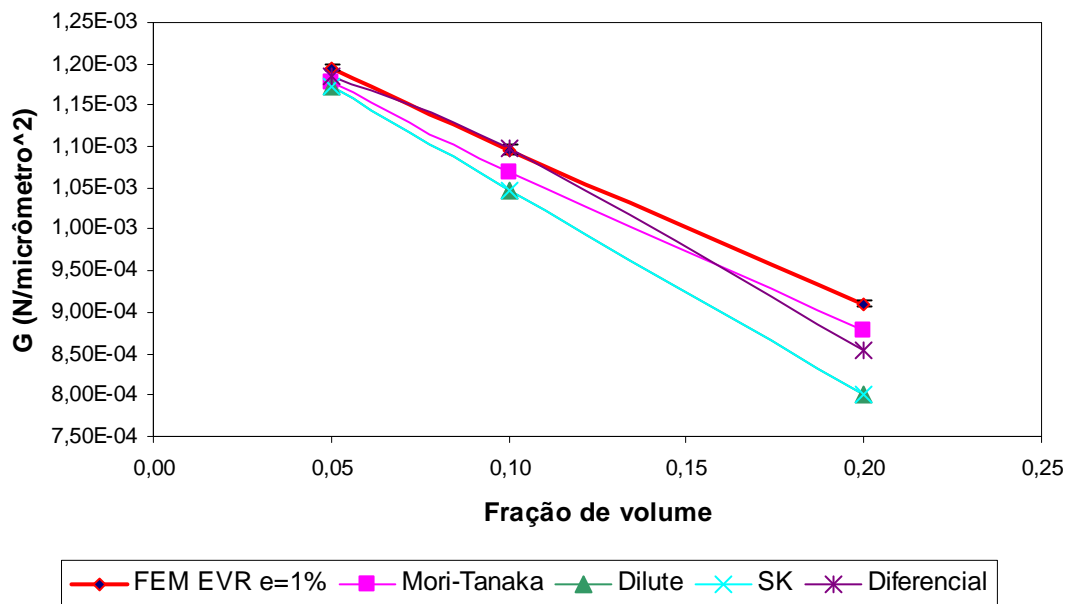


Figura 5.14 – Detalhe sem os limites de Voigt e Reuss da comparação do módulo de elasticidade transversal G obtido por este estudo (MEF) para EVRs com erro=1%, com os modelos teóricos

5.1.4.4 Estudo paramétrico da influência da distância mínima entre as heterogeneidades nos resultados de K e G de amostras com 10 nódulos (EVR).

Outro estudo realizado foi a verificação se há ou não influência significativa da distância mínima entre as superfícies das heterogeneidades nos valores das constantes elásticas K e G . A análise foi verificada utilizando os valores médios das constantes já citadas para cinco amostras de 10 nódulos, que corresponde ao EVR encontrado, onde se atribuiu as distâncias mínimas entre as partículas esféricas de $0,005r$, $0,01r$, $0,04r$, $0,07r$, $0,1r$ e $0,25r$, onde r é o raio do nódulo, mantido constante. As figuras 5.15 e 5.16 mostram cortes e tensões de dois modelos com distâncias mínimas entre os nódulos de $0,005r$ e $0,25r$, respectivamente, enquanto as figuras 5.17 e 5.18 mostram as curvas obtidas que representam os valores de K e G nestes casos.

Pode-se observar claramente que não se registra influência significativa da distância mínima entre os nódulos nos valores das constantes elásticas K e G . Na figura 5.17, vê-se que a relação entre o maior e o menor valor de K possíveis é da ordem de 1,00956, que corresponde à uma diferença de aproximadamente 1%. Quanto ao módulo G , esta diferença corresponde a aproximadamente 2% (Figura 5.18).

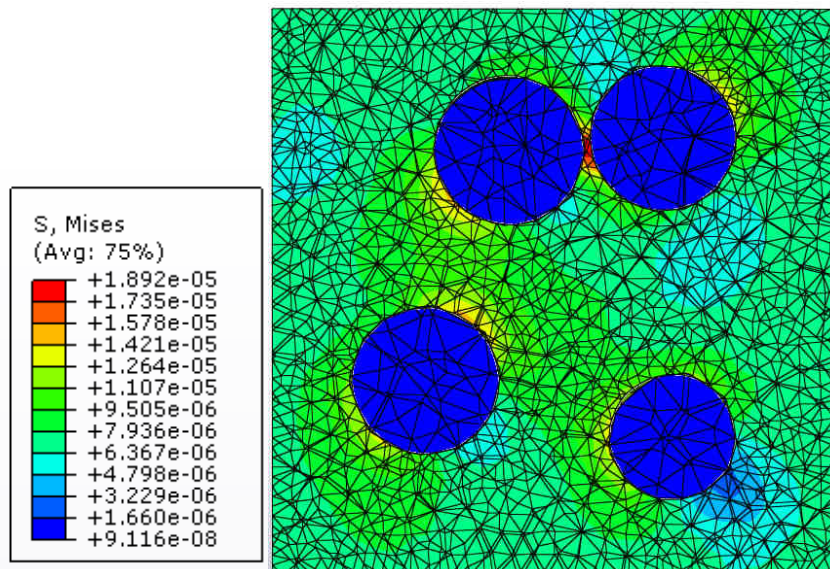


Figura 5.15 – Corte em um modelo com 10 heterogeneidades e distância mínima entre elas de $0,005r$

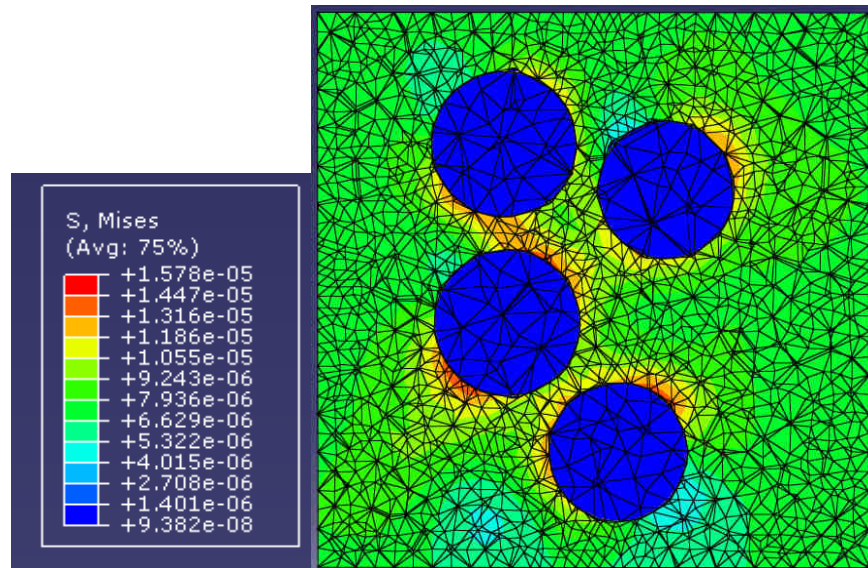


Figura 5.16 – Corte em um modelo com 10 heterogeneidades e distância mínima entre elas de $0,25r$

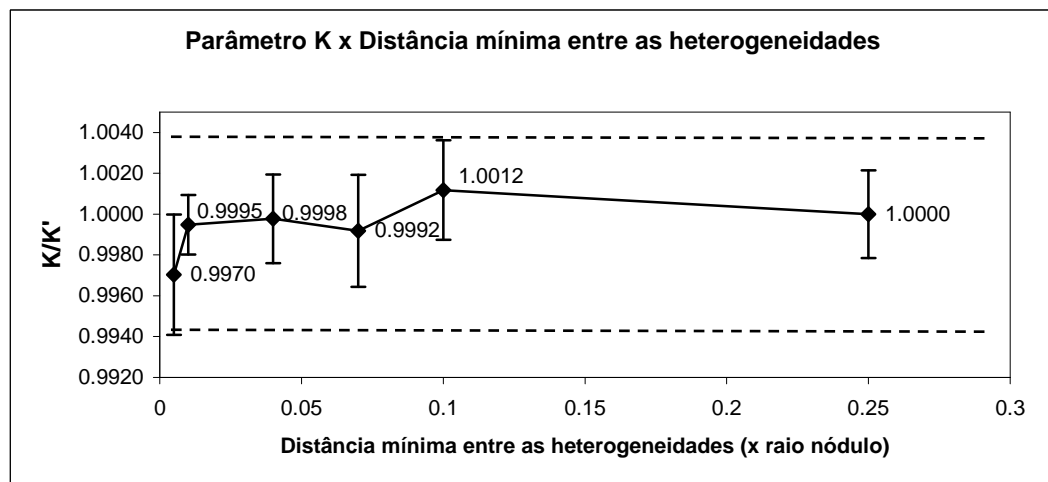


Figura 5.17 – Evolução do módulo K com o aumento da distância mínima entre as heterogeneidades, onde K' é o valor de K em $0,25r$

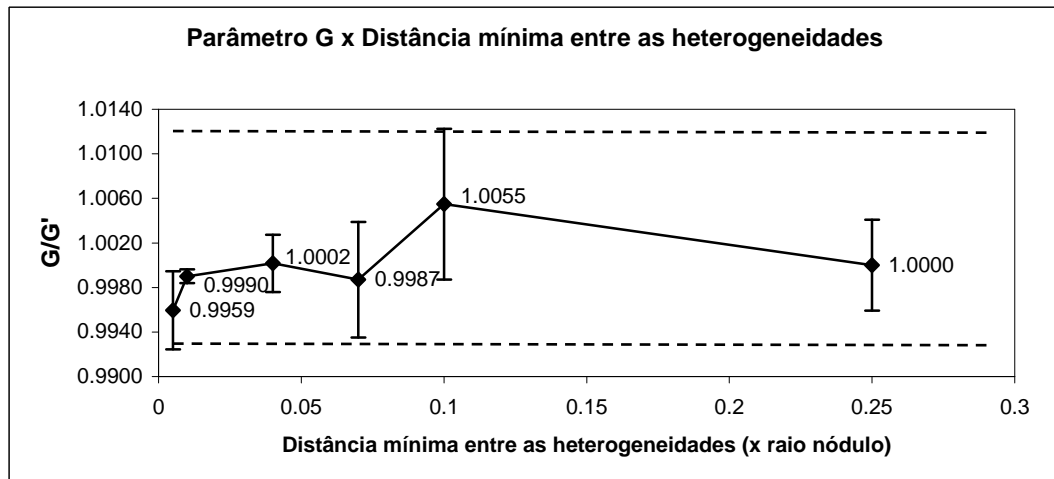


Figura 5.18 – Evolução do módulo G com o aumento da distância mínima entre as heterogeneidades, onde G' é o valor de G em $0,25r$

5.2 Homogeneização numérica elástica não-linear de um composto com distribuição aleatória de heterogeneidades esféricas hiperelásticas dentro de uma matriz de PMMA.

5.2.1 Caracterização do modelo

A micro-estrutura estudada é um material composto formado por uma matriz elástica de PMMA ($E = 3240\text{MPa} = 3240 \times 10^{-6} \frac{\text{N}}{\mu\text{m}^2}$, $\nu = 0,25$), com heterogeneidades esféricas de um material elastomérico de borracha natural (N55) com $\nu = 0,49$, distância mínima entre as superfícies de $0,25r$, onde r é o raio esférico, distância mínima entre as superfícies e a borda da matriz de $0,06r$, com configuração geométrica similar a analisada no item 5.1. A lei constitutiva do elastômero foi ajustada utilizando o modelo Yeoh (3 termos) com os seguintes parâmetros: $C_{10} = 0,7211 \times 10^{-6}$, $C_{20} = -0,1785 \times 10^{-6}$, e $C_{30} = 0,0567 \times 10^{-6}$. A equação geral deste modelo foi apresentada na equação (2.64b), cuja forma expandida para 3 termos, no Abaqus/Simulia, 2009, é representada por:

$$W = C_{10}(I_{G1} - 3) + C_{20}(I_{G1} - 3)^2 + C_{30}(I_{G1} - 3)^3 \quad (5.6)$$

onde:

C_{10} , C_{20} , C_{30} : são constantes;

I_{G1} : é o primeiro invariante de deformação;

W : é a densidade de energia de deformação.

Abaixo são apresentadas as curvas *tensão x deformação*, experimental e ajustada pelo software Abaqus/Simulia, 2009, do modelo hiperelástico adotado.

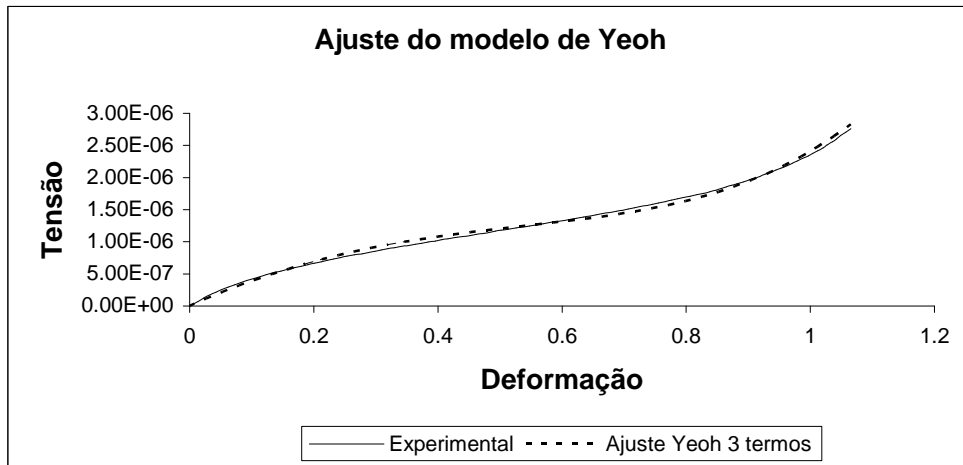


Figura 5.19 – Curvas experimental e ajustada, do modelo hiperelástico utilizado para a obtenção dos parâmetros C_{10} , C_{20} e C_{30}

O raio das heterogeneidades foi mantido constante em $r = 1\mu m$ e o tamanho do corpo de prova foi variando à medida que nele eram adicionados nódulos, de forma a manter constante a fração volumétrica de heterogeneidades em relação ao volume da matriz em 10%. A distribuição das posições das esferas na matriz foi dada de forma aleatória utilizando a metodologia proposta por Ortiz, 2001. Cabe salientar que se teve a necessidade de aumentar a distância mínima entre as partículas para que não ocorressem distorções excessivas em elementos da matriz elástica localizados próxima aos nódulos. Percebeu-se que, aumentando essa distância, essas distorções diminuam, embora não tenham sido quantificadas. O sistema foi modelado com uso do software comercial de elementos finitos Abaqus/Simulia, 2009 considerando a matriz elástica linear e as heterogeneidades hiperelásticas. O comportamento do cubo foi verificado através de uma análise transiente, aplicando-se um esquema explícito de integração (Método das Diferenças Centrais), conforme apresentado na seção 2.6.

Abaixo se apresenta um quadro com as variáveis envolvidas na análise explícita utilizada.

Tabela 5.6 – Variáveis envolvidas na análise explícita

	PMMA		ELASTÔMERO	
	$[\frac{Kg}{m^3}]$	$[\frac{\hat{K}g}{\mu m^3}]$	$[\frac{Kg}{m^3}]$	$[\frac{\hat{K}g}{\mu m^3}]$
Massa específica	1240	1240×10^{-24}	950	950×10^{-24}
Amortecimento	Default: <i>Linear bulk viscosity</i> =0,06 <i>Quadratic bulk viscosity</i> =1,2			
Δt	<i>Automatic time incrementation - Global</i>			
Amplitude carga	<i>Tabular – com incremento uniforme de carga</i>			

Discussão sobre a transformação de unidades realizada

Para que o sistema Abaqus/Simulia, ao realizar a integração explícita, guarde a quantidade de algarismos significativos desejados, trabalhou-se com unidades não convencionais. As unidades de referência deste trabalho foram *Newton [N]* para as forças e *micrômetro [μm]* para os comprimentos. Desta forma houve a necessidade de uma adequação da unidade da massa específica para que seja consistente com $[N]$ e $[\mu m]$. Esta unidade de massa foi nomeada de $\hat{K}g$, sendo que a mesma deve respeitar a seguinte relação:

$$F = ma \rightarrow 1N = \frac{1\hat{K}g1\mu m}{s^2} \quad (5.7)$$

Isolando a massa na expressão (5.7) fica

$$1\hat{K}g = \frac{1Ns^2}{\mu m} \quad (5.8)$$

Se

$$1Kg = \frac{1Ns^2}{m} \quad (5.9)$$

Então de (5.8) e (5.9) chega-se a

$$1Kg = 10^{-6} \hat{K}g \quad (5.10)$$

Assim, a unidade de massa específica sofre a transformação:

$$\left[\frac{Kg}{m^3} \right] = \frac{10^{-6} \hat{K} g}{(10^6 \mu m)^3} = 10^{-24} \left[\frac{\hat{K} g}{\mu m^3} \right] \quad (5.11)$$

Então, na análise no sistema Abaqus foi utilizada como unidade de massa o $\hat{K} g$. Na tabela 5.6 são especificados os valores de densidade introduzidos para cada uma das fases do modelo.

5.2.2 Caracterização do material elástico não-linear através de um estado de tensão-deformação característico e do conceito de “material map”

Conforme método já apresentado no item 3.3 procedeu-se a aplicação da técnica de homogeneização elástica não-linear apresentada por Temizer e Zohdi, 2007, em que o comportamento do material é caracterizado por uma equação constitutiva e também pelo conceito de *material map*. Admite-se nesse estudo realizado pelos autores acima, que o material em estudo é elástico não-linear, isotrópico e homogêneo. Para que a hipótese da permanência da isotropia durante o ensaio possa ser considerada, foram estudados corpos de prova submetidos a níveis moderados de deformação, como será visto logo adiante, ainda nesta seção. Com isso, evita-se que as heterogeneidades, ao se deformarem demasiadamente, orientem-se e percam a condição de isotropia. Para aplicar esta metodologia a maiores níveis de deformação será necessário verificar que a perda de isotropia pelo alinhamento das heterogeneidades deformadas não seja significativa. As tensões são apresentadas através do segundo tensor de Piola-Kirchhoff (T_2), cujas transformações são dadas pela equação (2.56). No entanto, o presente estudo foi estendido a um caso tridimensional, onde as relações constitutivas assumiram a forma da equação (2.65) em que foi considerado o coeficiente α_2 . Esta equação pode ser expandida na forma matricial, sendo apresentada na forma

$$\begin{pmatrix} 1 & \lambda_E^1 & (\lambda_E^1)^2 \\ 1 & \lambda_E^2 & (\lambda_E^2)^2 \\ 1 & \lambda_E^3 & (\lambda_E^3)^2 \end{pmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} \lambda_S^1 \\ \lambda_S^2 \\ \lambda_S^3 \end{Bmatrix} \quad (5.12)$$

onde λ_E^i podem ser determinados, a partir do tensor F , como indica as equações (2.69) e ilustra a figura (2.9), e S (segundo tensor de Piola-Kirchhoff) está, nesse contexto,

simbolizado por T_2 . Assim, foi realizada a transformação apresentada na equação (2.56), uma vez que a tensão de saída no Abaqus/Simulia, 2009, foi a de Cauchy.

A expressão (5.12) estabelece que, para cada nível de estiramento λ_F^i , há um valor específico de deformação e tensão λ_E^i e λ_S^i e, conseqüentemente, um α_i relacionado. Estes coeficientes α_i são função, no caso de um material isotrópico, do primeiro e segundo invariantes do tensor de deformações de Green (I_E e II_E) já apresentados nas equações (2.59) e (2.60). Estes coeficientes α_i foram calculados para configurações com 2, 4, 8, 16, 32, e 64 heterogeneidades, considerando 4 amostras de cada para os modelos de 2 a 16 nódulos, 2 amostras e 1 amostra para os modelos de 32 e 64 nódulos respectivamente, uma vez que se observou um pequeno e decrescente coeficiente de variação nas amostras, conforme gráficos da evolução do CV(%) que serão apresentados na seção 5.2.4 adiante, através do mapeamento discreto do material (*discrete material map*). Para calcular este mapeamento foi seguida a metodologia apresentada por Temizer e Zohdi, 2007, sendo que o algoritmo se apresenta a seguir:

```

 $\lambda_F^1 = \lambda_F^{\min}$ 
 $\lambda_F^2 = \lambda_F^{\min} + a$ 
 $\lambda_F^3 = \lambda_F^{\min} + a + b$ 
do ( $\lambda_F^1 = \lambda_F^1 + \lambda_F^{inc}$ ) while ( $\lambda_F^1 \leq \lambda_F^{max}$ )
  do ( $\lambda_F^2 = \lambda_F^2 + \lambda_F^{inc}$ ) while ( $\lambda_F^2 \leq \lambda_F^1 + a$ )
    do ( $\lambda_F^3 = \lambda_F^3 + \lambda_F^{inc}$ ) while ( $\lambda_F^3 \leq \lambda_F^2 + b$ )
      calcular:  $\{\lambda_E^1, \lambda_E^2, \lambda_E^3\}$ 
      medir:  $\{\lambda_S^1, \lambda_S^2, \lambda_S^3\}$ 
      Obter:  $\mathbf{S} = \alpha_0 \mathbf{I} + \alpha_1 \mathbf{E} + \alpha_2 \mathbf{E}^2$ 
      resolver e registrar:  $\{\alpha_0, \alpha_1, \alpha_2\}$ 
    end do
  end do
end do
end do

```

Figura 5.20 – Algoritmo para criação do mapeamento discreto do material elástico não-linear proposto por Temizer e Zohdi, 2007

Pode-se observar no algoritmo da figura 5.20, que o mapeamento se dá partindo de incrementos dos estiramentos λ_F^i em um intervalo de deformação caracterizado por

$\lambda_F^i \in [\lambda_F^{\min}, \lambda_F^{\max}]$. Nesse algoritmo, os valores de a e b são acréscimos nos estiramentos para não causar indeterminação no sistema (5.12). A figura (5.21) mostra um exemplo trabalhado por Temizer e Zohdi, 2007 em que um material hiperelástico de Mooney-Rivlin é modelado de forma bidimensional. Nesse caso, foram utilizados apenas os dois primeiros termos da equação constitutiva (2.65).

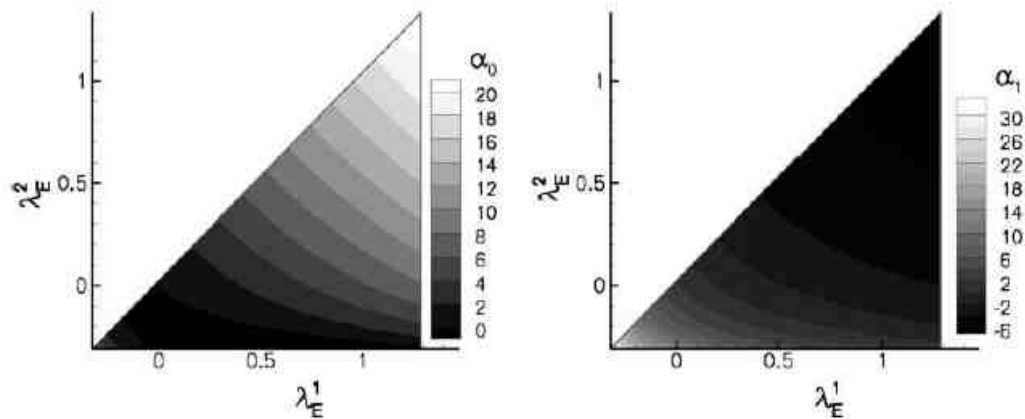


Figura 5.21 – Mapeamento discreto de um material de Mooney-Rivlin em uma formulação bidimensional [Temizer e Zohdi, 2007]

O mapeamento do material, similar à forma apresentada na figura 5.21, porém num caso tridimensional com heterogeneidades hiperelásticas imersas numa matriz elástica, foi realizado no domínio $\lambda_F^i \in [0.80, 1.19]$, isto é, foram aplicados deslocamentos prescritos moderados que variaram de uma compressão de 20% até uma tração de 19% no material, para evitar que a hipótese de isotropia fosse perdida.

O valor de λ_F^{inc} adotado foi de 0.10, $a=b=0.045$ com $n=3$ incrementos, de forma que os valores de λ_F^i foram

$$\begin{aligned}\lambda_F^1 &= \{0,80;0,90;1,00;1,10\} \\ \lambda_F^2 &= \{0,845;0,945;1,045;1,145\} \\ \lambda_F^3 &= \{0,89;0,99;1,09;1,19\}\end{aligned}\tag{5.13}$$

Ou, aplicando as equações (2.69), em termos de deformação, tem-se

$$\begin{aligned}\lambda_E^1 &= \{-0,18; -0,095; 0,0; 0,105\} \\ \lambda_E^2 &= \{-0,14299; -0,05349; 0,046013; 0,155513\} \\ \lambda_E^3 &= \{-0,10395; -0,00995; 0,09405; 0,20805\}\end{aligned}\quad (5.14)$$

O número de estados de carga utilizados na simulação de cada amostra, resultante da combinação dos valores de λ_F^i foi $N=(n+1)^3=64$.

5.2.3 Malha

A malha utilizada foi formada por elementos finitos tetraédricos de 4 nós com razão entre tamanho da malha máximo e raio dos nódulos de 0,3. Foi considerada uma união perfeita na interface partícula/matriz sem permitir o descolamento entre elas. Sendo esta uma primeira aproximação neste tipo de problemas é esperado descolamento entre a heterogeneidade e a matriz. Mas esta é a sugestão natural para a continuação do trabalho.

5.2.4 Tempo de análise e velocidade de deformação

A determinação do tempo a ser empregado na análise foi baseada na convergência dos valores de α_0 , α_1 e α_2 em função da velocidade de deformação $\dot{\lambda}_E^3 = \frac{\lambda_E^3}{t}$ aplicada, onde t é o tempo total aplicado à simulação e λ_E^3 a deformação na direção 3, já apresentada na expressão (2.69). O estudo foi realizado para os tamanhos de amostras envolvidos com o estado de carga $\lambda_F^1 = 0,90$, $\lambda_F^2 = 0,945$ e $\lambda_F^3 = 1,19$. A tabela 5.7 mostra os tempos aplicados e suas respectivas velocidades de deformação e as figuras 5.22 a 5.27 apresentam o resultado deste estudo, onde os valores no eixo x estão em ordem inversa e escala logarítmica.

Tabela 5.7 – Tempos de análise explícita aplicados e suas respectivas velocidades de deformação

Tempo [s]	Velocidade $\dot{\lambda}_E^3$ [1/s]
1×10^{-9}	2.08×10^8
1×10^{-8}	2.08×10^7
3×10^{-8}	6.94×10^6
1×10^{-7}	2.08×10^6
1×10^{-6}	2.08×10^5

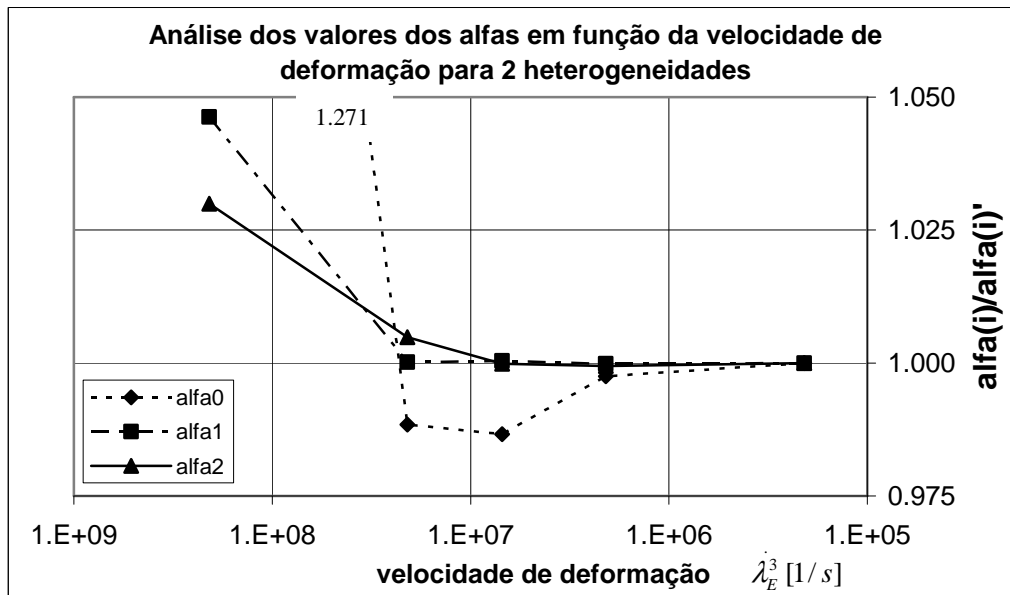


Figura 5.22 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 2 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

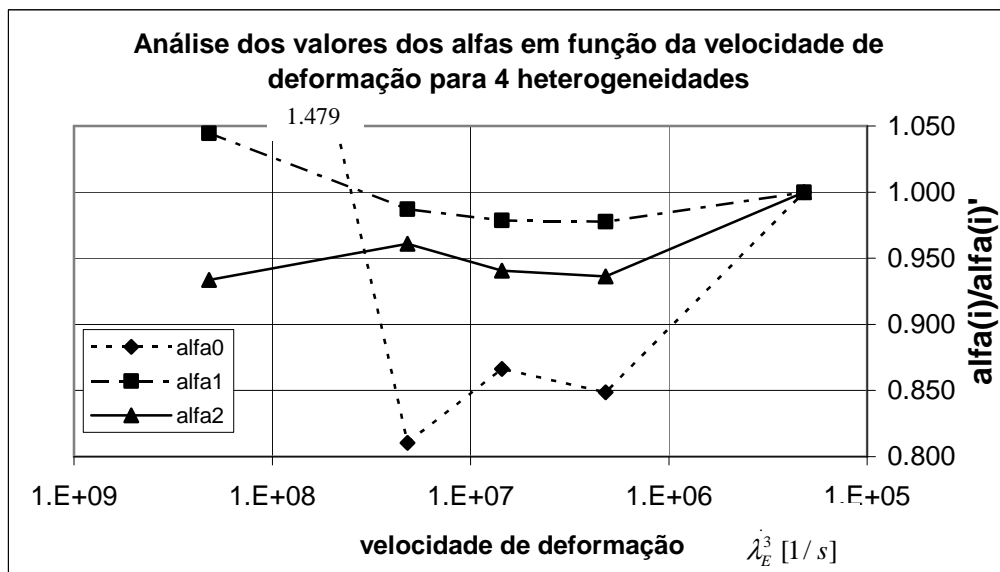


Figura 5.23 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 4 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

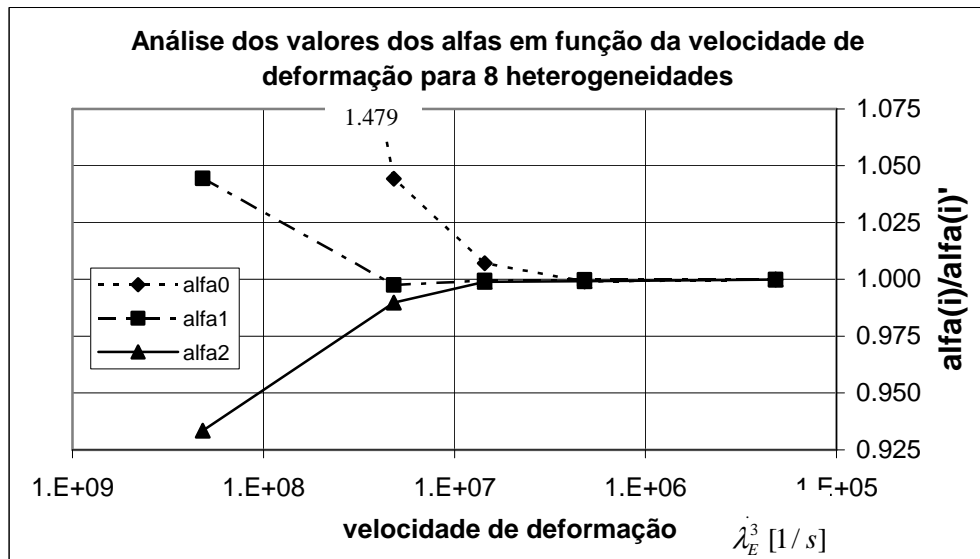


Figura 5.24 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação λ_E^3 para uma amostra de 8 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

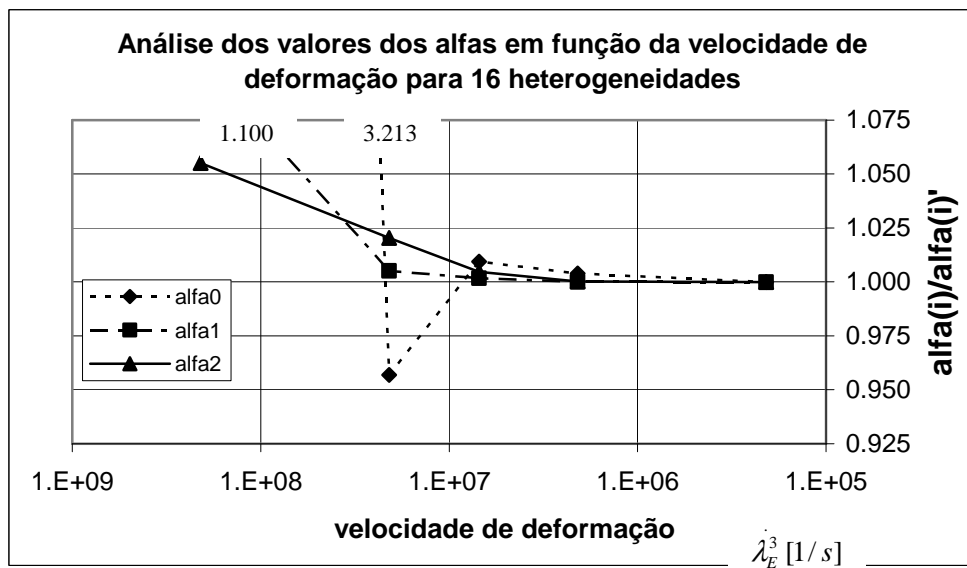


Figura 5.25 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação λ_E^3 para uma amostra de 16 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

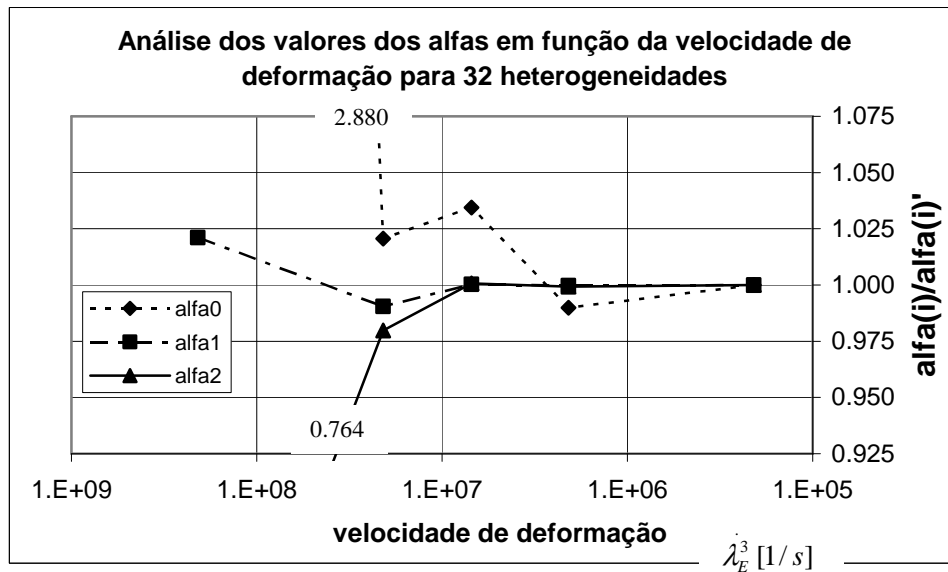


Figura 5.26 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 32 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

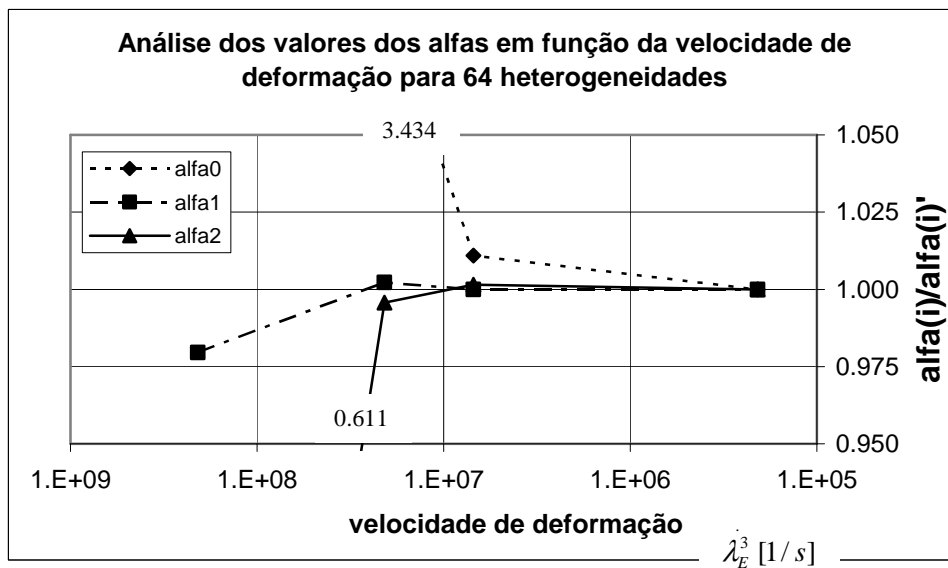


Figura 5.27 – Estudo de convergência dos valores de alfa em relação à velocidade de deformação $\dot{\lambda}_E^3$ para uma amostra de 64 heterogeneidades; alfa(i)' é o valor de alfa(i) obtido com a menor velocidade de deformação usada

Pode-se observar nos casos acima que os valores de α_i ficam praticamente inalterados até a velocidade de deformação 6.94×10^6 . Nesta velocidade, o maior erro, em relação aos valores de menor velocidade analisados, foi verificado na amostra de 4 nódulos, sendo de

aproximadamente 13% para o α_0 , 2% para o α_1 e 6% no caso do α_2 . Como nos demais casos essa diferença foi de, no máximo, em torno de 1%, como se observa na figura 5.28, foi utilizada esta velocidade, $\dot{\lambda}_E^3 = 6.94e + 6$, para análise dos modelos desta etapa.

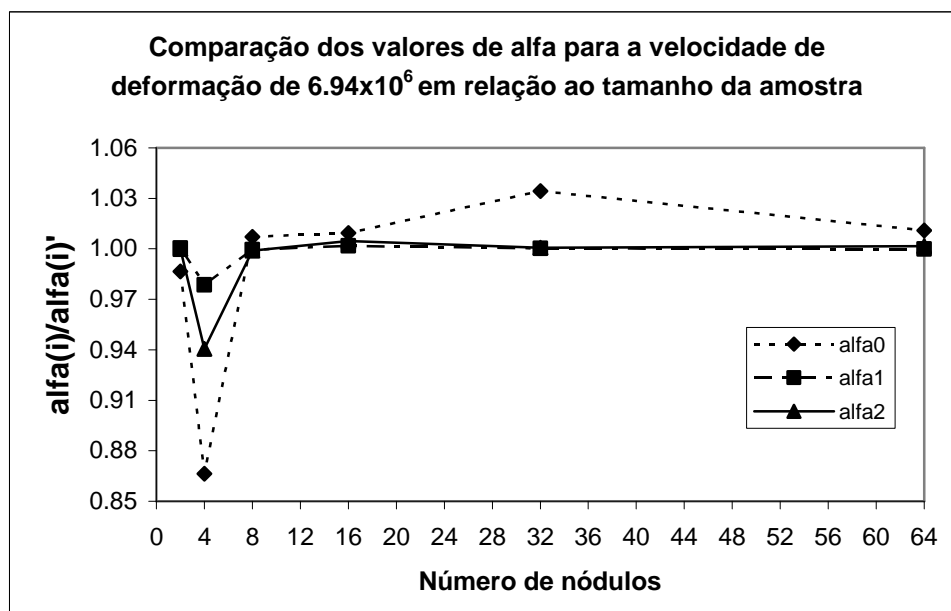


Figura 5.28 – Compara o dos valores de alfa na velocidade $\dot{\lambda}_E^3 = 6.94e + 6$, entre os tamanhos de amostras estudados; $\alpha(i)'$   o valor de $\alpha(i)$ obtido com a menor velocidade de deformaç o usada

5.2.5 Determina o do elemento de volume representativo

A t cnica proposta para determina o do elemento de volume representativo   a an lise da converg ncia dos valores de α_i para modelos com n mero crescente de heterogeneidades, no caso, 2, 4, 8, 16, 32 e 64 n dulos, no dom nio de deforma es dado pelo intervalo de estiramentos $\lambda_F^i \in [0,80; 1,19]$. Para confec o dos mapas de α_i fixou-se os valores de λ_E^1 no dom nio indicado em (5.14) e calculou-se para cada estado de deforma o determinado os valores de α_0 , α_1 , e α_2 para cada amostra de 2, 4, 8, 16, 32 e 64 heterogeneidades. Calculou-se a m dia aritm tica e o desvio padr o entre as amostras e os mapas foram gerados com os valores m dios resultantes. Estes mapas foram obtidos por programa o em Matlab, em que foram aplicados o algoritmo apresentado na figura 5.20 e as equa es (5.12) em interface com o software comercial de elementos finitos Abaqus/Simulia, 2009. A figura 5.29 mostra um esquema com os pontos do dom nio λ_E^i estudados. Os valores

intermediários foram obtidos por interpolação linear entre os valores de λ_E^2 e λ_E^3 . Pode-se perceber através da figura 5.29 que, com os quatro valores de deformação de λ_E^1 fixos apresentados na expressão (5.14), tem-se os 64 estados de deformação aplicados às amostras.

Pontos mapeados no domínio das deformações

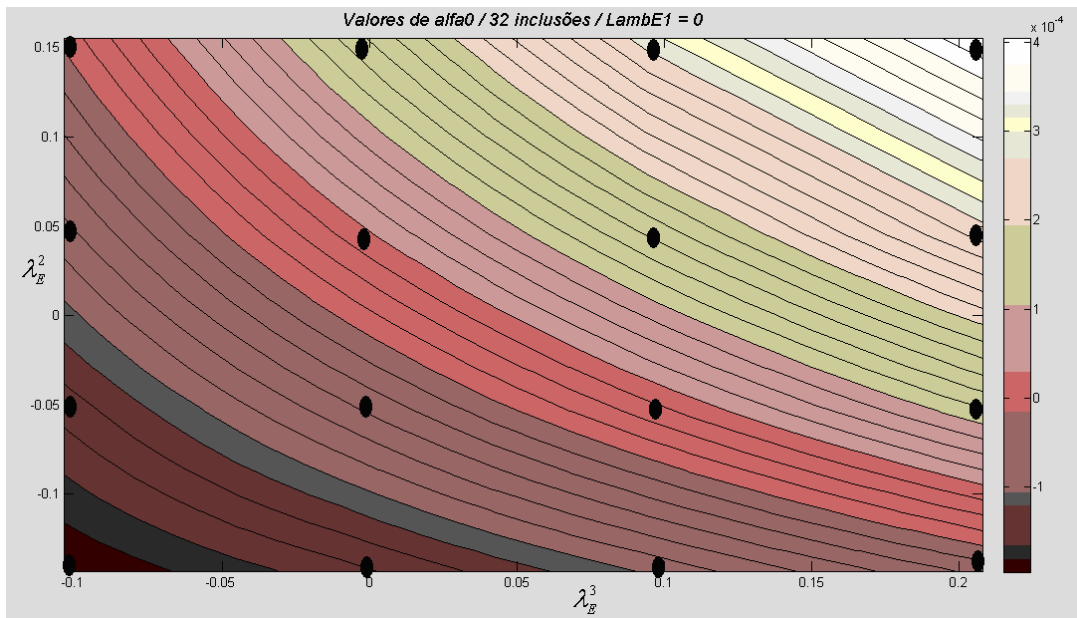
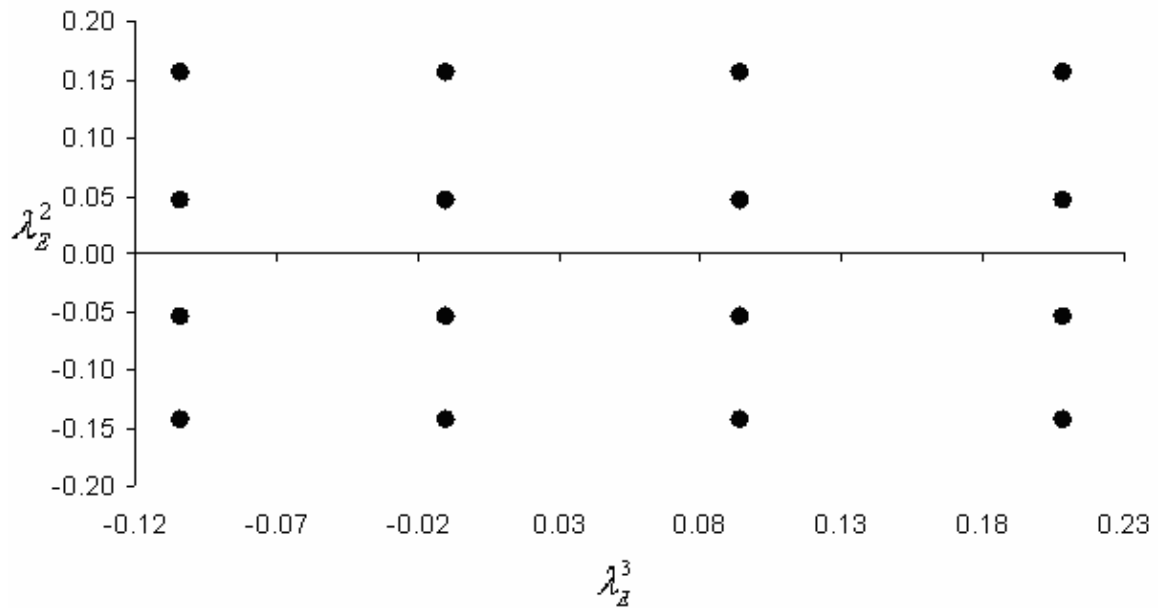


Figura 5.29 – Esquema com os pontos do domínio λ_E^i mapeados para determinação dos α_i da lei constitutiva do material. Acima esquema ilustrativo e abaixo exemplo do mapeamento gerado em Matlab para o caso de α_0 de 32 heterogeneidades e $\lambda_E^1 = 0$, com os pontos mapeados

Na sequência, a figura 5.30 apresenta como ilustração, os mapas de valores de α_0 para alguns casos estudados.

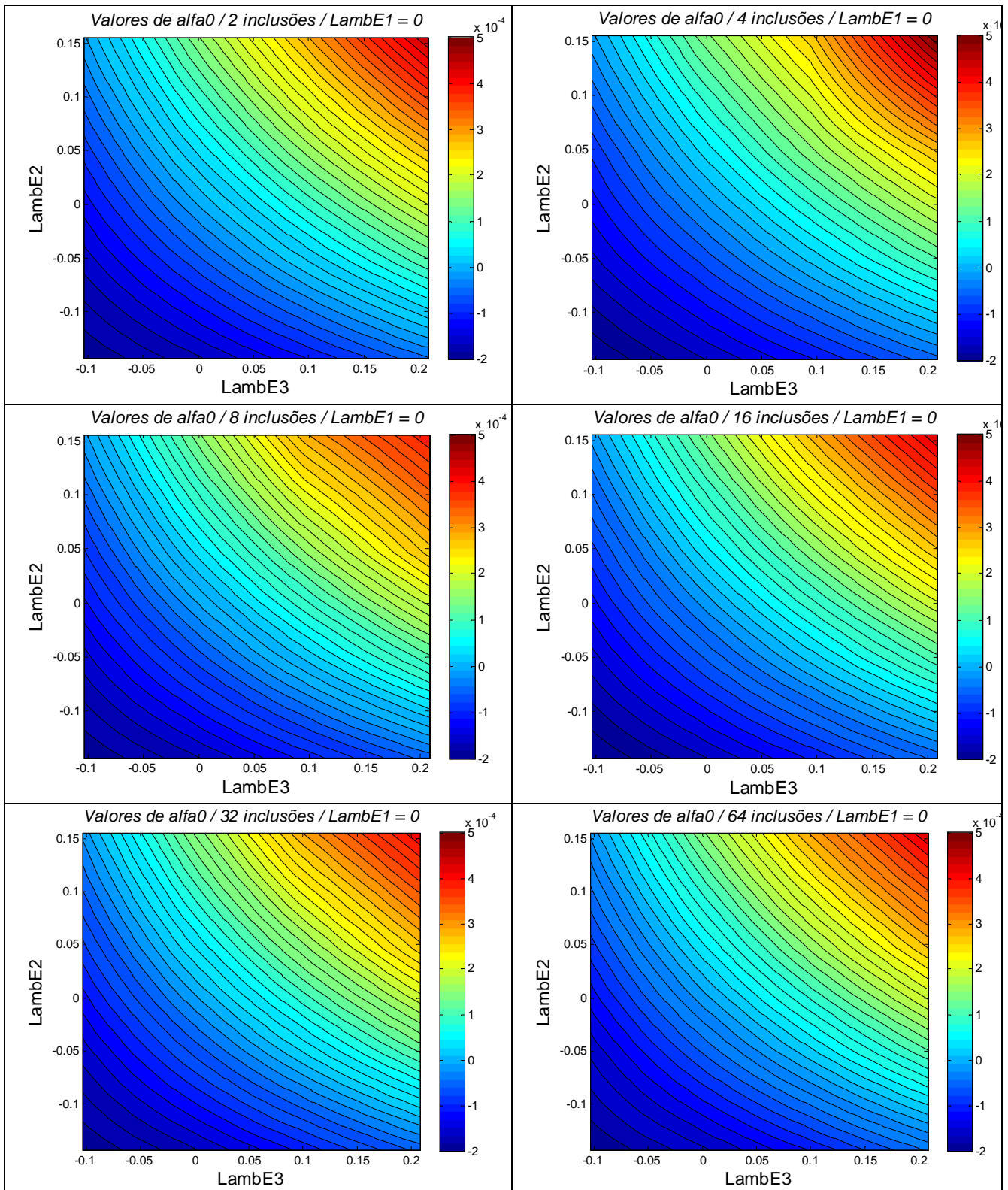


Figura 5.30 – Mapeamento dos valores de α_{0ij} médios do composto com a fixação de $\lambda_E^1 = 0$, para todos os tamanhos de amostras

As linhas da Figura 5.30 foram geradas em Matlab, onde os valores de α_i dos estados de carga intermediários aos pontos do domínio de λ_E^i foram obtidos por interpolação linear.

No anexo A encontram-se os gráficos dos demais estados de carga mapeados na análise realizada.

A determinação do elemento de volume representativo foi realizada através da convergência dos índices I_0 , I_1 e I_2 que são funções dos os valores de α_0 , α_1 e α_2 , respectivamente, segundo as relações

$$I_{0(jk)} = \frac{\left[\sum_{i=1}^N \frac{(\overline{\alpha_{0(2j,i)}} - \alpha_{0(j,ik)})^2}{(\overline{\alpha_{0(2j,i)}})^2} \right]}{N} \quad (5.15)$$

$$I_{1(jk)} = \frac{\left[\sum_{i=1}^N \frac{(\overline{\alpha_{1(2j,i)}} - \alpha_{1(j,ik)})^2}{(\overline{\alpha_{1(2j,i)}})^2} \right]}{N} \quad (5.16)$$

$$I_{2(jk)} = \frac{\left[\sum_{i=1}^N \frac{(\overline{\alpha_{2(2j,i)}} - \alpha_{2(j,ik)})^2}{(\overline{\alpha_{2(2j,i)}})^2} \right]}{N} \quad (5.17)$$

onde:

k : índice do número da amostra para cada modelo, que varia de 1 a M amostras, sendo M=4 para as amostras de 2, 4, 8 e 16 nódulos, M=2 para a amostra de 32 nódulos e M=1 para a amostra de 64 nódulos;

N : índice que indica o número total de estados de deformação ($N=64$);

i : índice que controla o número de estados de carga, que varia de 1 a N . Por exemplo $i=1$ corresponde a $\lambda_E^1 = -0.18$, $\lambda_E^2 = -0.142987$ e $\lambda_E^3 = -0.10395$; para $i=64$ tem-se $\lambda_E^1 = 0.105$, $\lambda_E^2 = 0.155513$ e $\lambda_E^3 = 0.20805$;

j : índice que indica número de nódulos da amostra (2, 4, 8, 16, 32);

$2j$: índice que indica número de nódulos da amostra sucessiva;

$\overline{\alpha_l}$: valor médio do coeficiente α_l ($l=0, 1, 2$), $\overline{\alpha_l} = \frac{\sum_{k=1}^M \alpha_{l(k)}}{M}$;

$\alpha_{l(j,ik)}$: valor de α_l para uma amostra k , de j nódulos para um estado de deformações aplicado i . Lembrando que ($l=0, 1, 2$), ($k=1, \dots, M$), ($j=2, 4, 8, 16, 32$), ($i=1, \dots, 64$);

$$\text{E, finalmente, } \overline{\alpha_{l(2j,i)}} = \frac{\sum_{k=1}^M \alpha_{l(2j,ik)}}{M}.$$

Pode-se observar pelas equações (5.15), (5.16) e (5.17) que os índices de convergência são obtidos pela relação entre desvios normalizados, de tamanhos de amostras subsequentes, para evitar influência do valor absoluto do coeficiente no valor do índice. Estes índices são calculados para cada amostra, de forma que é possível obter suas médias. Definem-se então os índices médios para cada número de nódulos como

$$I_{0(j)} = \overline{I_{0(jk)}} \quad (5.18)$$

$$I_{1(j)} = \overline{I_{1(jk)}} \quad (5.19)$$

$$I_{2(j)} = \overline{I_{2(jk)}} \quad (5.20)$$

Desta forma, espera-se que os valores dos índices de convergência tendam a zero quando se atinja o elemento de volume representativo (EVR).

Abaixo, nas figuras (5.31) até (5.33), se apresentam os gráficos mostrando a convergência dos índices propostos.

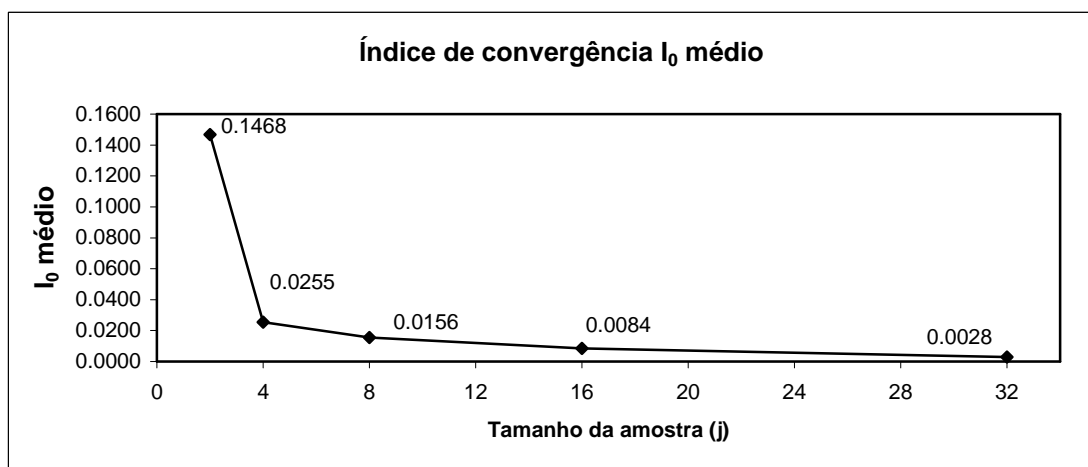


Figura 5.31 – Convergência do índice I_0 em relação ao tamanho da amostra

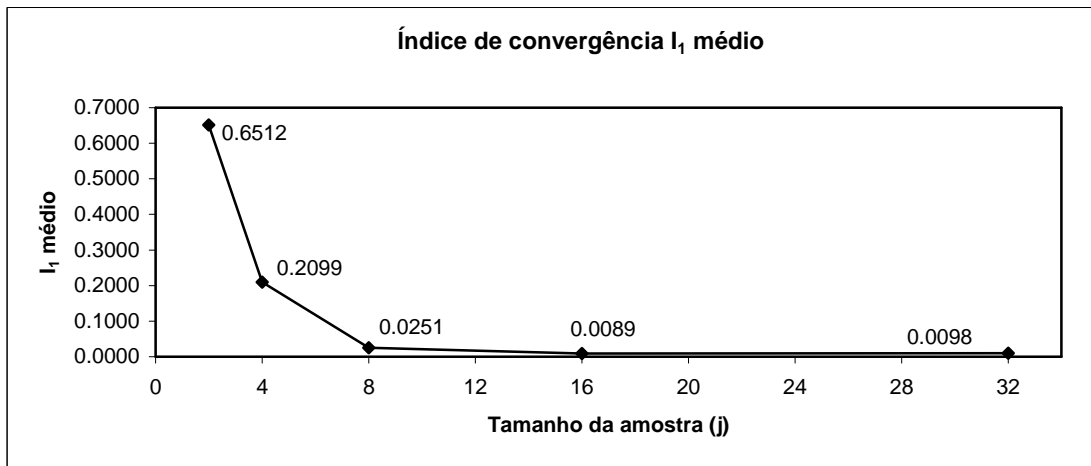


Figura 5.32 – Convergência do índice I_1 em relação ao tamanho da amostra

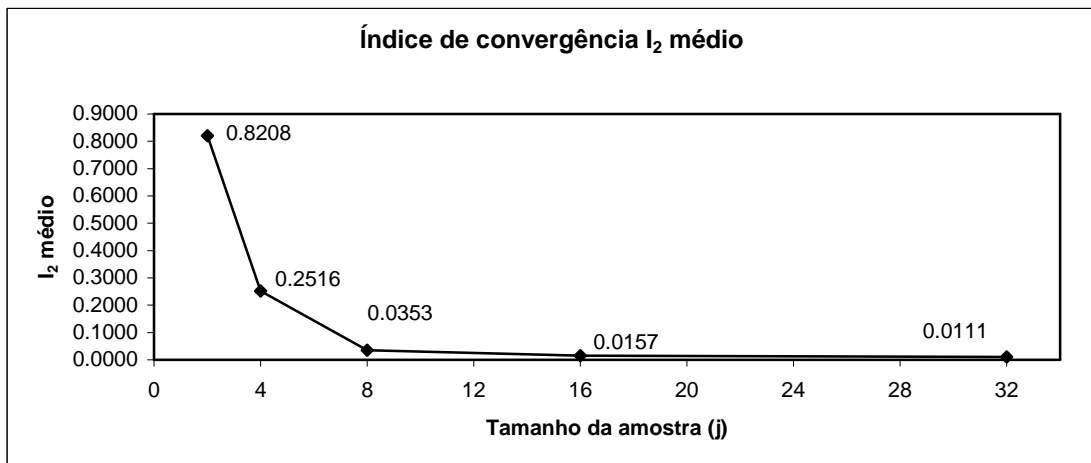


Figura 5.33 – Convergência do índice I_2 em relação ao tamanho da amostra

A figura 5.34 mostra os três índices convergindo a valores muito próximos de zero a partir da análise entre 8 e 16 nódulos, como detalha a figura 5.35 em escala do índice reduzida.

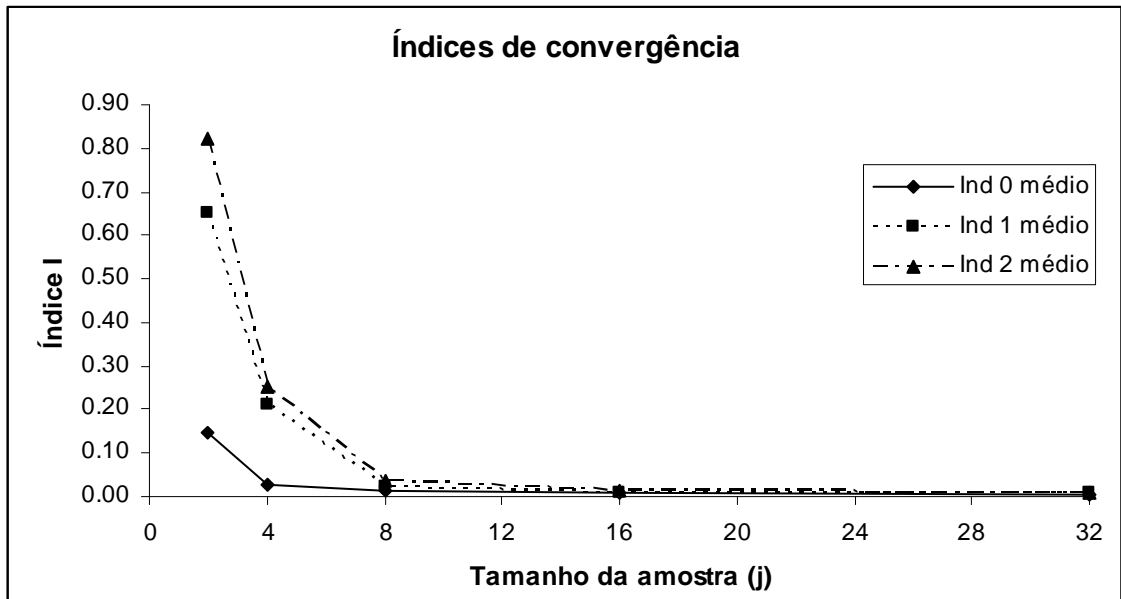


Figura 5.34 – Convergência dos índices I_l em relação ao tamanho da amostra

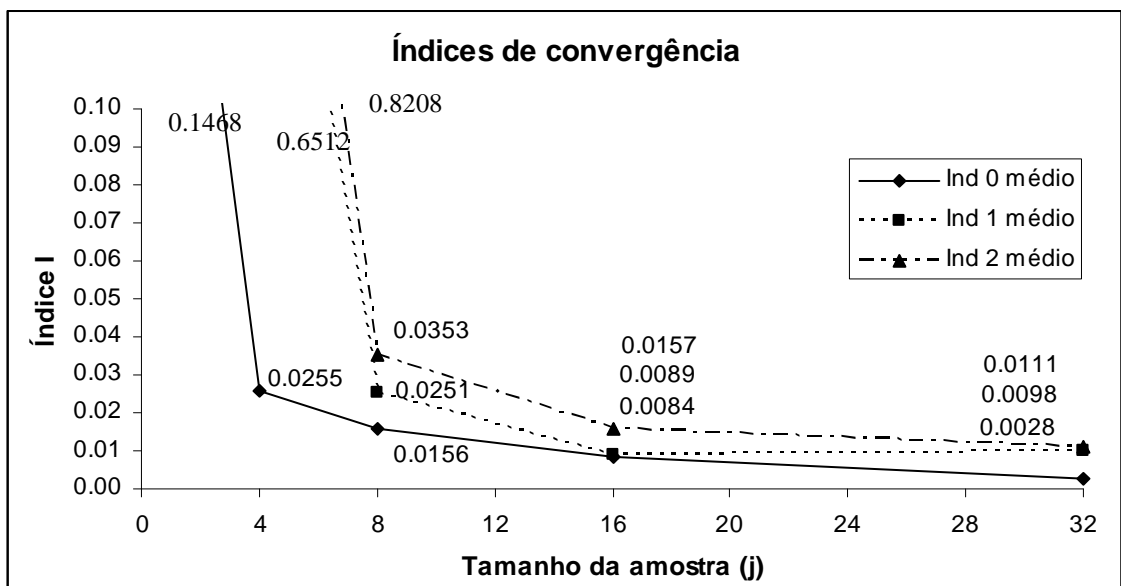


Figura 5.35 – Detalhe da figura 5.34 com redução da escala do índice

Uma forma alternativa de verificar o tamanho do modelo a partir do qual a resposta mecânica permanece inalterada foi escolher alguns pontos, no caso 12 pontos, e verificar a partir de qual tamanho houve a convergência dos valores de α_{ij} médios. Os estados de carga escolhidos são mostrados na tabela 5.8 e graficados nas figuras 5.36 e 5.37. Cabe salientar que esta verificação foi realizada com o intuito de corroborar se a informação global dos índices

I_0 , I_1 e I_2 sobre a qual está sendo baseada a determinação do EVR não estava mascarando perturbações locais.

Tabela 5.8 – Estados de deformação para análise da convergência dos valores de α_{ij} médios

Ponto	λ_F^i			λ_E^i		
	λ_F^1	λ_F^2	λ_F^3	λ_E^1	λ_E^2	λ_E^3
1	0.90	0.945	0.89	-0.095	-0.05349	-0.10390
2	0.90	1.145	0.89	-0.095	0.15550	-0.10390
3	0.90	0.945	1.19	-0.095	-0.05349	0.20805
4	0.90	1.145	1.19	-0.095	0.15550	0.20805
5	1.00	0.945	0.89	0	-0.05349	-0.10390
6	1.00	1.145	0.89	0	0.15550	-0.10390
7	1.00	0.945	1.19	0	-0.05349	0.20805
8	1.00	1.145	1.19	0	0.15550	0.20805
9	1.10	0.945	0.89	0.105	-0.05349	-0.10390
10	1.10	1.145	0.89	0.105	0.15550	-0.10390
11	1.10	0.945	1.19	0.105	-0.05349	0.20805
12	1.10	1.145	1.19	0.105	0.15550	0.20805

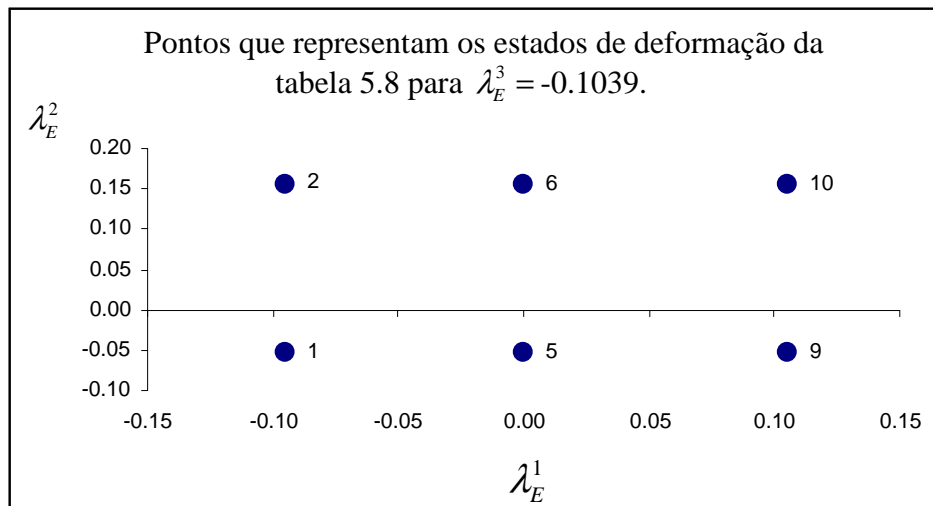


Figura 5.36 – Pontos 1, 2, 5, 6, 9 e 10 da tabela 5.8 para $\lambda_E^3 = -0.1039$

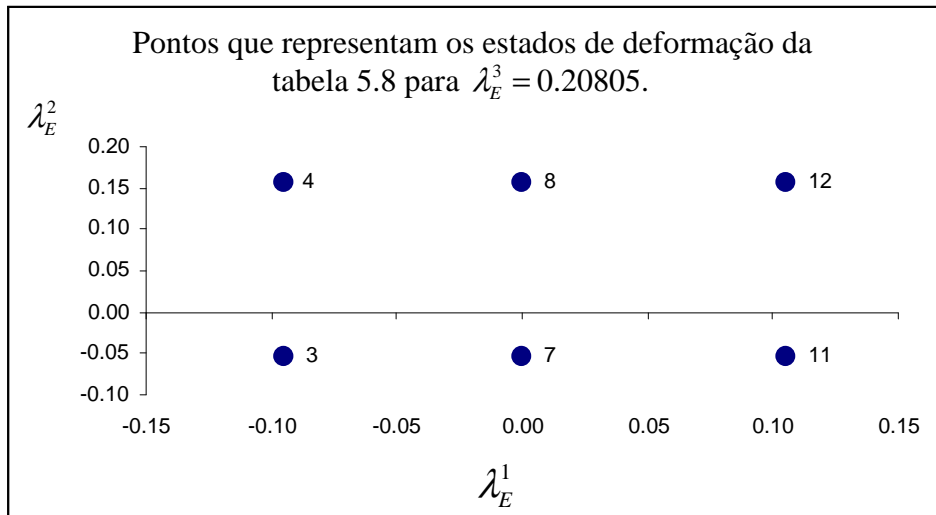


Figura 5.37 – Pontos 3, 4, 7, 8, 11 e 12 da tabela 5.8 para $\lambda_E^3 = 0.20805$

A seguir, nas figuras 5.38 a 5.52 se apresentam os gráficos da convergência de α_{ij} médios de alguns dos pontos da tabela 5.8 onde os valores estão normalizados em relação ao valor de $\alpha_{i(i,j=64)}$ da maior amostra (64 nódulos). Nas figuras 5.53 e 5.54 tem-se a evolução do coeficiente de variação (CV) dos α_i para os carregamentos $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ (Ponto 2 da tabela 5.8) e $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ (Ponto 12 da tabela 5.8) em relação ao tamanho da amostra (em número de nódulos), sendo o coeficiente de variação dado pela razão entre os desvios dos valores de alfas e seus valores médios.

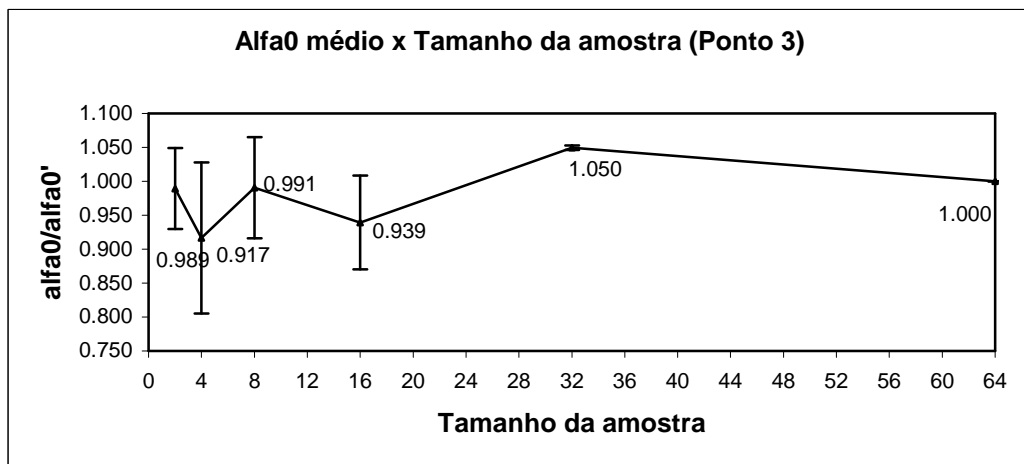


Figura 5.38. Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos).

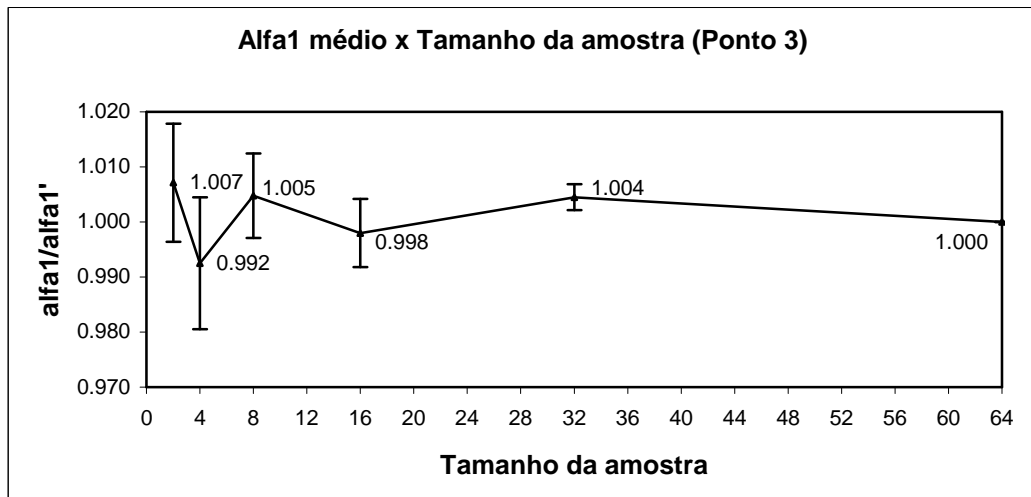


Figura 5.39 – Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos)

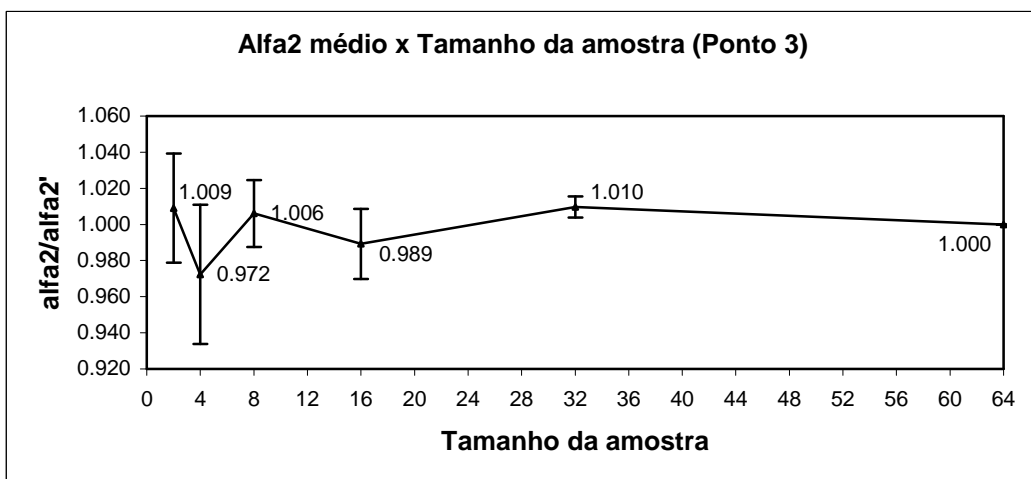


Figura 5.40 – Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos)

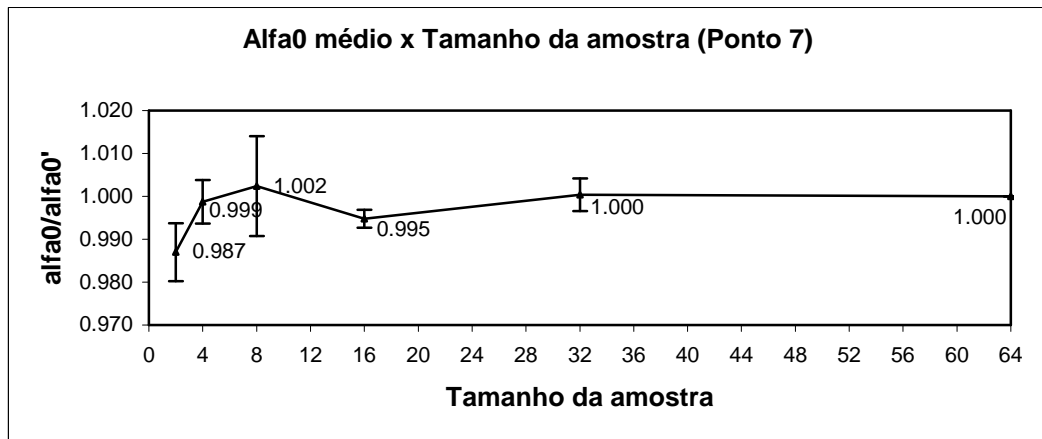


Figura 5.41 – Convergência do $\alpha_{0,j}$ médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos)

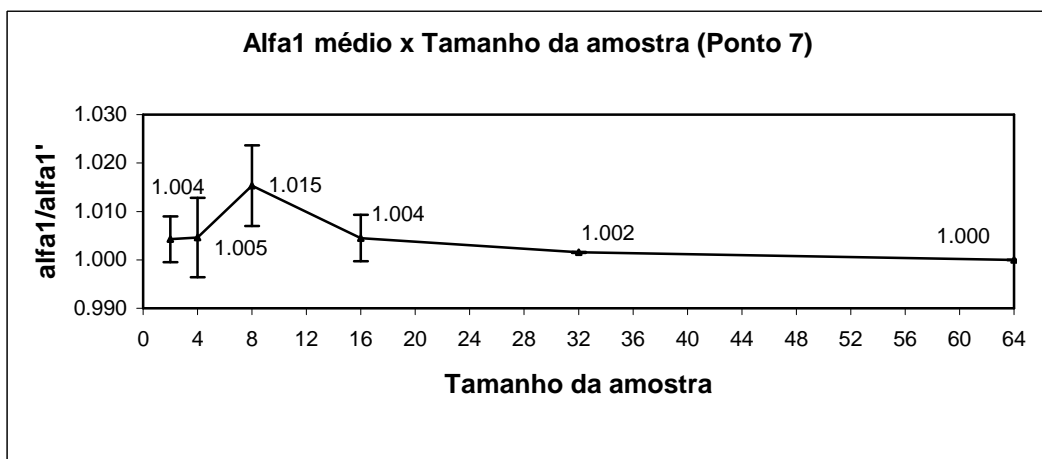


Figura 5.42 – Convergência do $\alpha_{1,j}$ médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos)

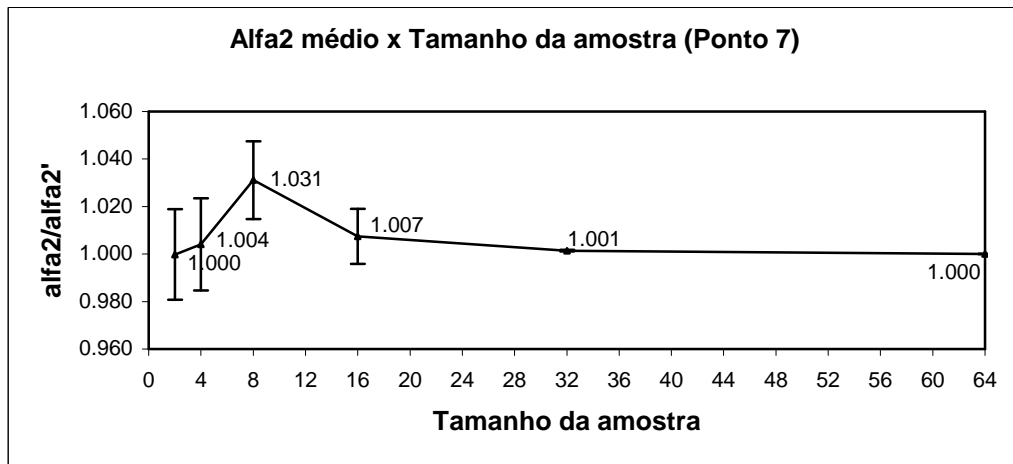


Figura 5.43 – Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 0.945$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos)

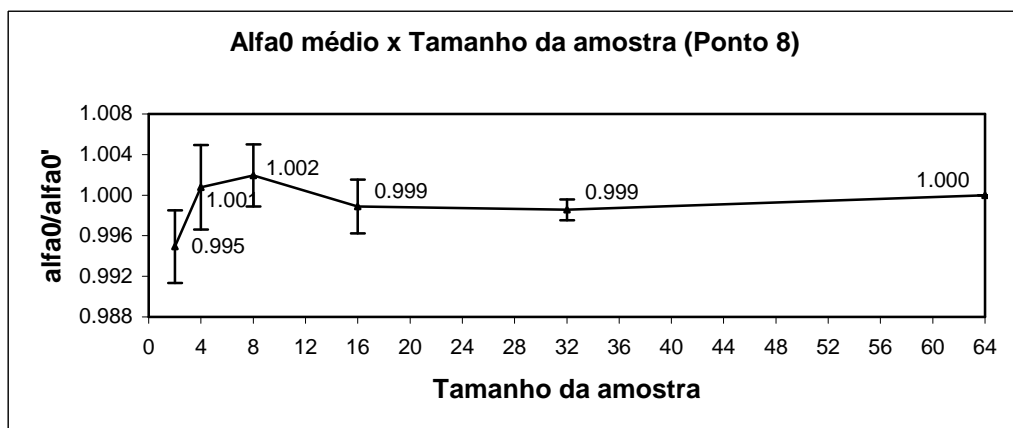


Figura 5.44 – Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos)

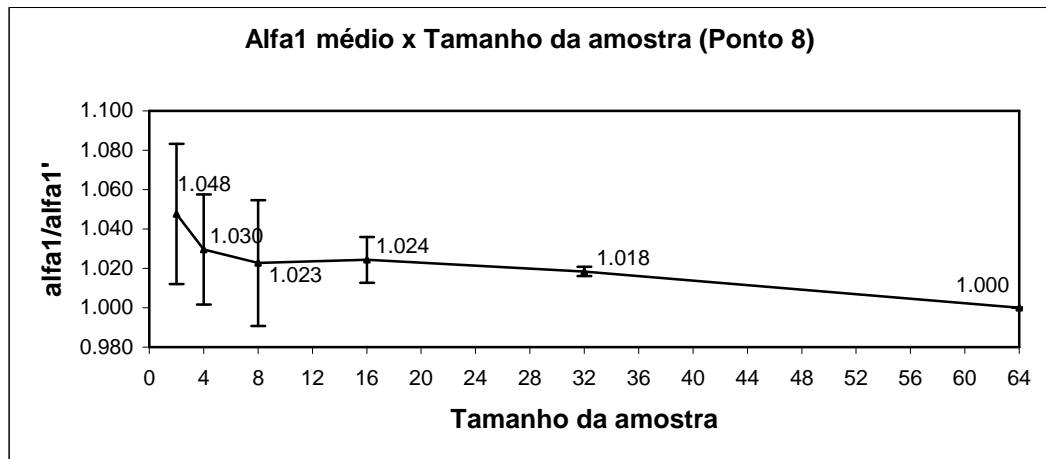


Figura 5.45 – Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos)

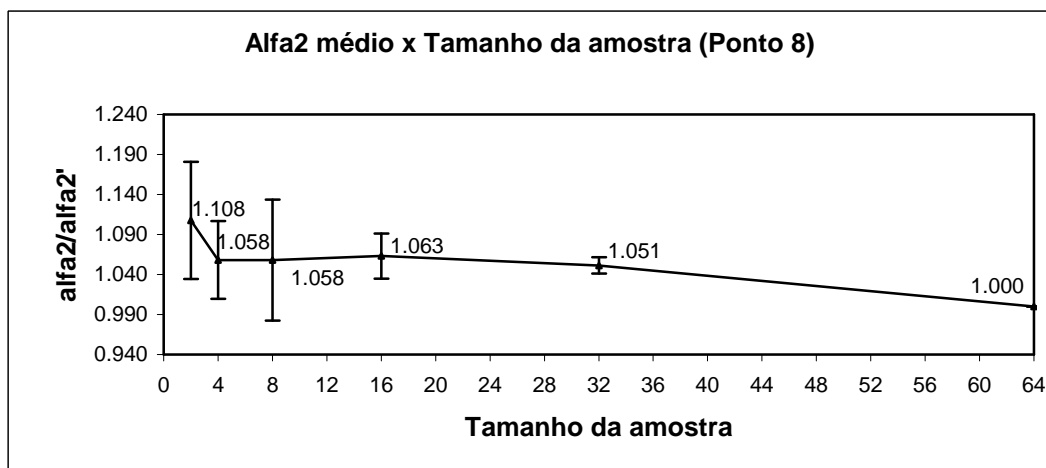


Figura 5.46 – Convergência do α_{2j} médio para o carregamento $\lambda_F^1 = 1.00$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos)

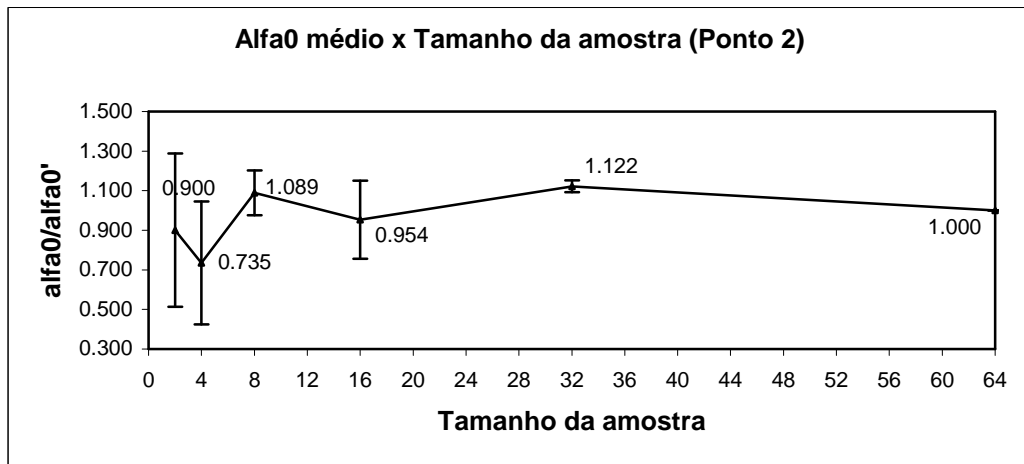


Figura 5.47 – Convergência do α_{0j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos)

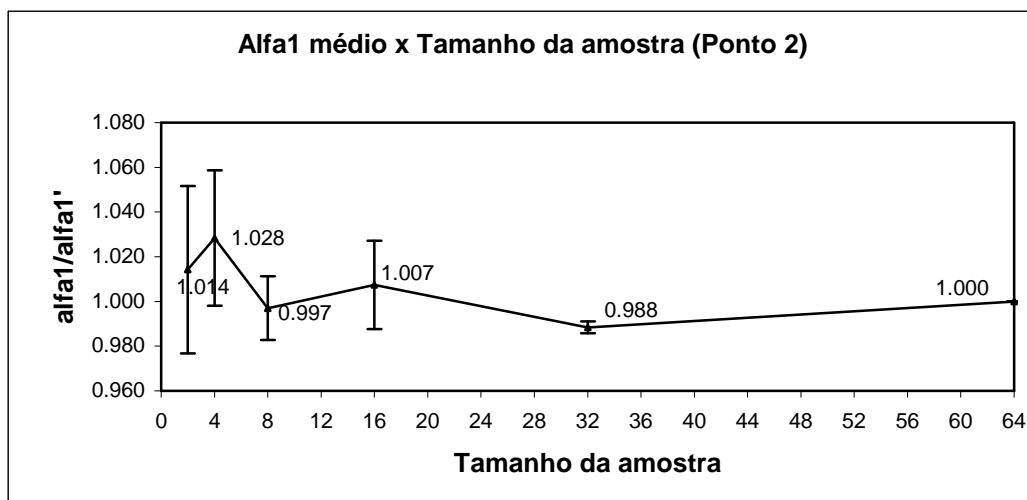


Figura 5.48 – Convergência do α_{1j} médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos)

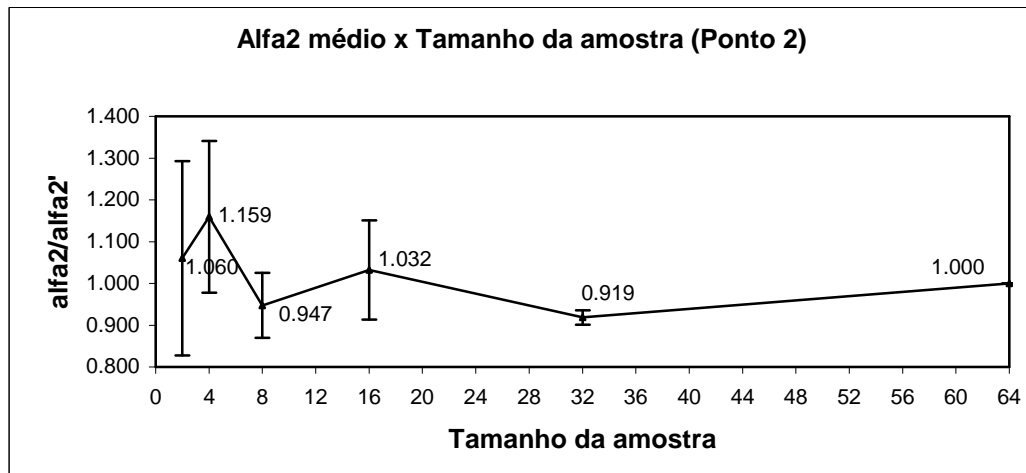


Figura 5.49 – Convergência do $\alpha_{2,j}$ médio para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos)

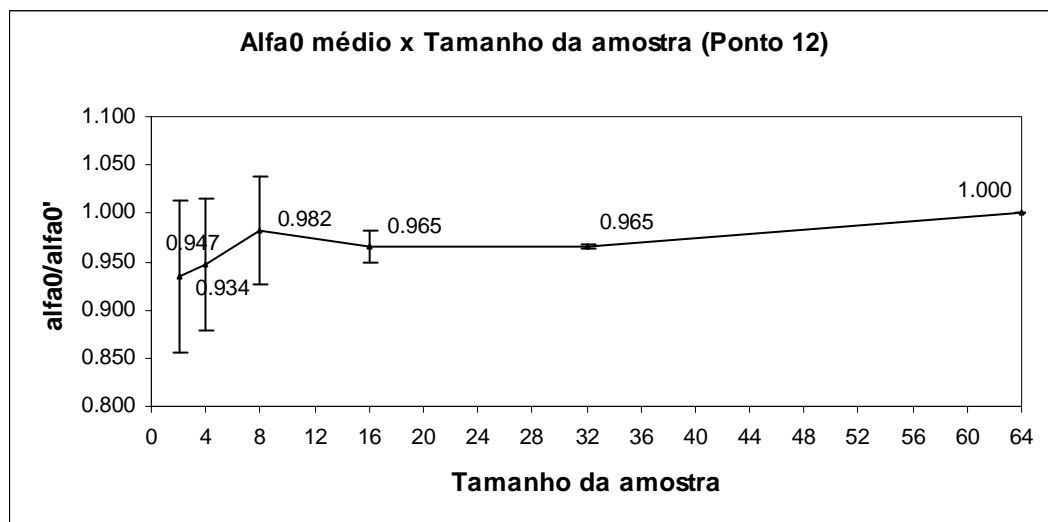


Figura 5.50 – Convergência do $\alpha_{0,j}$ médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa0' é o valor de $\alpha_{0,j=64}$ da maior amostra (64 nódulos)

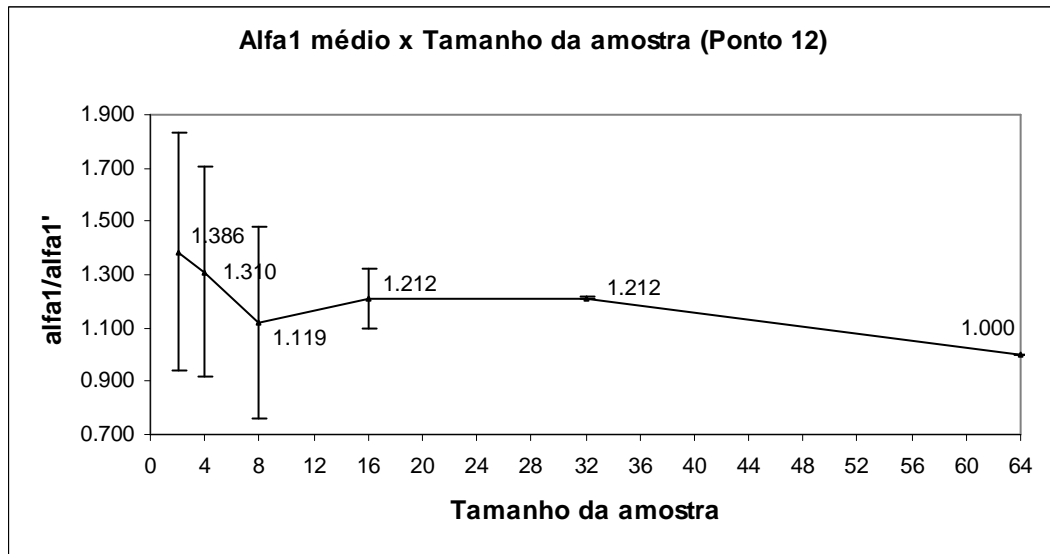


Figura 5.51 – Convergência do $\alpha_{1,j}$ médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa1' é o valor de $\alpha_{1,j=64}$ da maior amostra (64 nódulos)

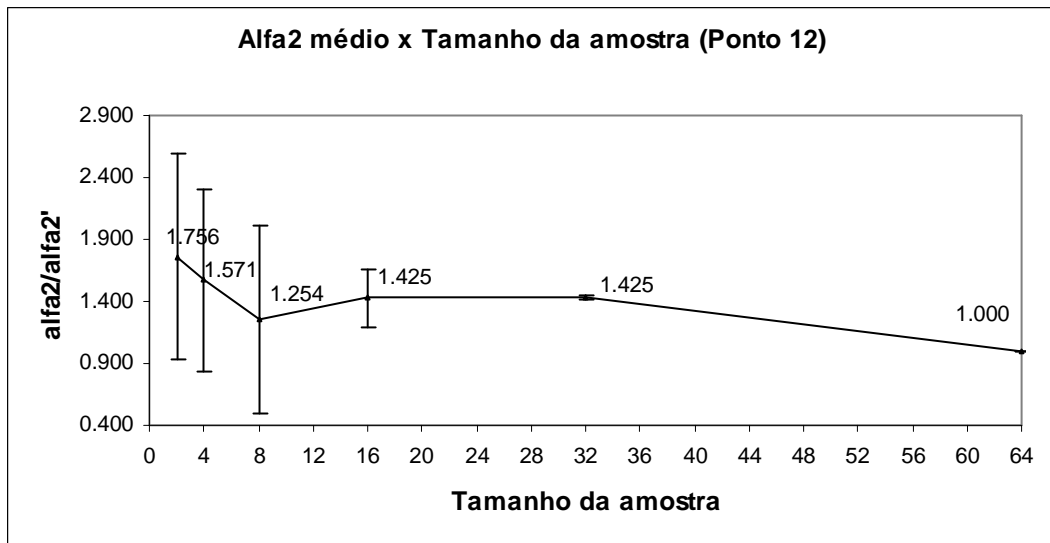


Figura 5.52 – Convergência do $\alpha_{2,j}$ médio para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ em relação ao tamanho da amostra, onde alfa2' é o valor de $\alpha_{2,j=64}$ da maior amostra (64 nódulos)

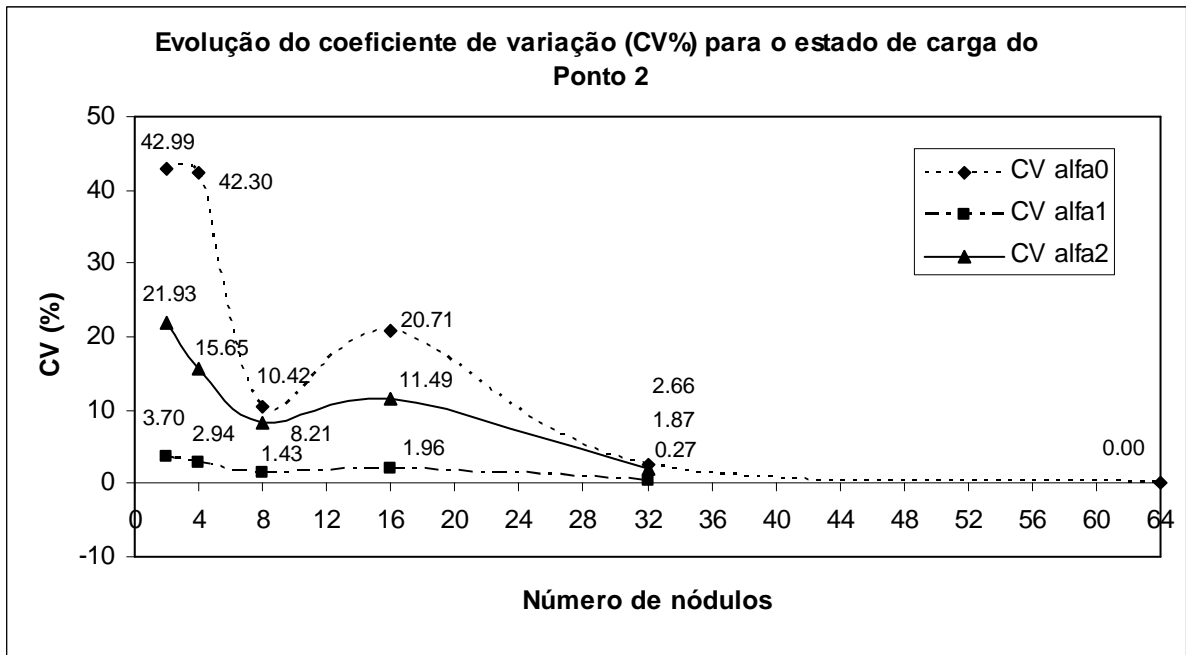


Figura 5.53 – Evolução do coeficiente de variação dos α_{ij} para o carregamento $\lambda_F^1 = 0.90$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.89$ (Ponto 2 da tabela 5.8) em relação ao tamanho da amostra

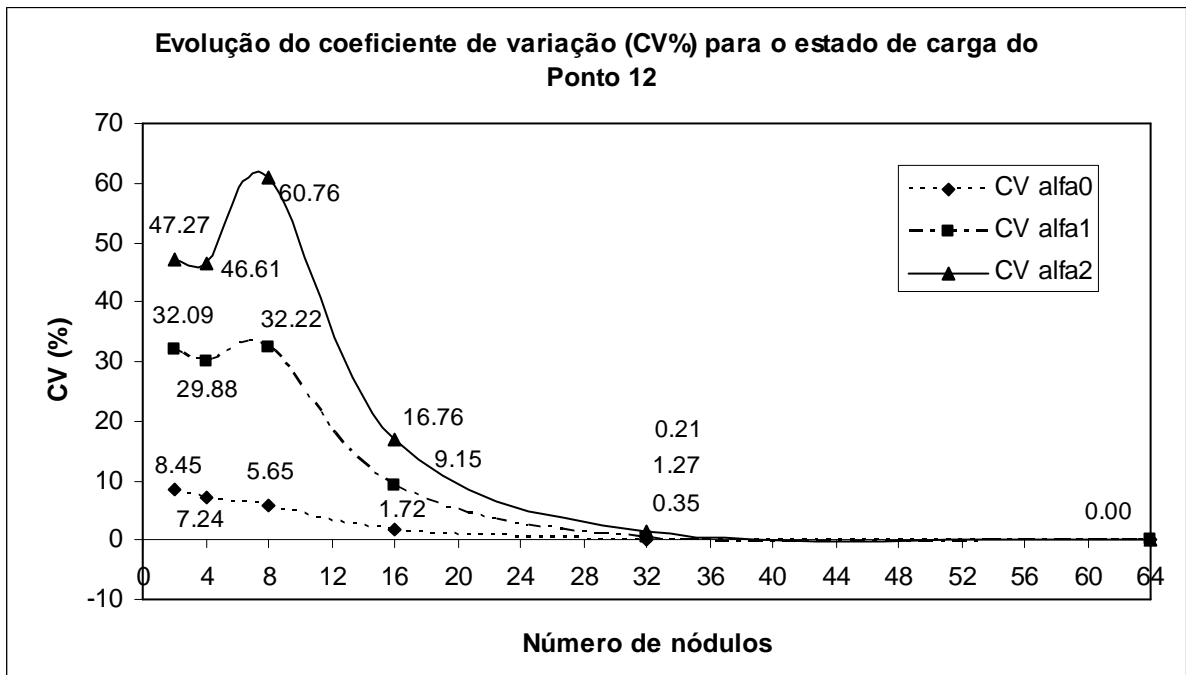


Figura 5.54 – Evolução do coeficiente de variação dos α_{ij} para o carregamento $\lambda_F^1 = 1.10$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 1.19$ (Ponto 12 da tabela 5.8) em relação ao tamanho da amostra

5.2.6 Análise e discussão dos resultados da etapa

Nesta seção visa-se discutir os resultados obtidos na seção 5.2.5 onde se aplicou uma técnica de homogeneização elástica não-linear semelhante à apresentada por Temizer e Zohdi, 2007, porém em modelos tridimensionais, buscando-se determinar o elemento de volume representativo de um material composto formado por heterogeneidades esféricas elastoméricas imersas aleatoriamente em uma matriz de PMMA considerada como elástica linear. Apresentou-se um estudo de convergência de parâmetros visando à determinação do EVR, baseada na análise global do comportamento dos índices I_0 , I_1 e I_2 apresentados nas expressões (5.18), (5.19) e (5.20) em relação ao tamanho da amostra. Tais índices são obtidos pela relação entre desvios normalizados, de tamanhos de amostras subsequentes, conforme se apresenta nas expressões (5.15), (5.16) e (5.17). O que se observa, analisando os gráficos das figuras 5.31 a 5.35 é que houve a convergência dos índices supracitados ao valor próximo de zero até o número máximo de amostras analisado, referente a 64 nódulos. Cabe salientar que alguns casos pontuais de deformação, em particular aqueles em que os estiramentos dados foram muito próximos em duas direções, como os pontos 1 e 2 da tabela 5.8, em que $\lambda_F^1 = 0.90$ e $\lambda_F^3 = 0.89$, e aqueles que se aproximam de um estado hidrostático, como o ponto 12, apresentaram coeficientes de variação elevados, revelando uma instabilidade da matriz de deformações λ_E^i para valores de deformação com estas características. Isto pode ser observado com clareza na análise dos parâmetros α_i do ponto 2 da tabela 5.8, ilustrados nas figuras 5.47 a 5.49 em que se observam os elevados desvios em relação à média e no gráfico da figura 5.53 que mostra a evolução do coeficiente de variação para o referido estado de carga, o qual somente assume valores reduzidos a um padrão aceitável a partir de 32 nódulos. Este aspecto também é ilustrado em relação ao ponto 12 da tabela 5.8, em que as figuras 5.50 a 5.52 e 5.54 revelam uma situação semelhante ao ponto 2, com valores aceitáveis de CV a partir de modelos com 32 heterogeneidades.

Estudo paramétrico da influência da posição dos pontos no domínio das deformações principais λ_E^i para a determinação dos coeficientes α_i

Adiante se apresenta um estudo paramétrico sobre a influência da posição dos pontos no domínio das deformações principais. Como se observa na equação (5.12) da seção 5.2.2, é necessário resolver um sistema de equações lineares para achar os valores dos coeficientes,

mas este sistema de equações fica mal condicionado se os valores de deformação são muito próximos. Por esta razão, no fluxograma da figura 5.20, nos laços responsáveis pelas mudanças dos valores das deformações a serem avaliadas, são incorporados os valores de a e b a serem incrementados, evitando, assim, situação de singularidade. Seguindo, se analisa o que acontece com os CVs de α_0 , α_1 e α_2 para o modelo de 2 heterogeneidades, empregando 4 amostras quando são utilizados os valores de $\lambda_F^1 = 0.89$, $\lambda_F^2 = 1.145$ e $\lambda_F^3 = 0.90$. Com estes três valores de deformação, devido à proximidade entre λ_F^1 e λ_F^3 , o coeficiente de variação dos coeficientes alfas calculados é muito elevado, como se observa na figura 5.53. Modificando o valor de λ_F^3 no sentido de aumentar a diferença entre os λ_F , obtiveram-se os resultados apresentados na tabela 5.9 e na figura 5.55, onde se observam os coeficientes de variação dos alfas para diferentes incrementos de deformação $\Delta\lambda_F^{1-3} = \lambda_F^1 - \lambda_F^3$. Quando este incremento aumenta, há uma redução significativa nos CVs dos coeficientes alfas.

Tabela 5.9 – Coeficiente de variação (%) de 4 amostras de 2 heterogeneidades com estados de carga diferentes mostrando a instabilidade da matriz de deformação para deformações próximas.

Estado de carga		CV(%)		
		α_0	α_1	α_2
1	$\lambda_F^1 = 0.90$ $\lambda_F^2 = 1.145$ $\lambda_F^3 = 0.89$ $ \Delta\lambda_F^{1-3} = 0.01$	42.99	3.70	21.93
2	$\lambda_F^1 = 0.90$ $\lambda_F^2 = 1.145$ $\lambda_F^3 = 0.915$ $ \Delta\lambda_F^{1-3} = 0.015$	6.24	0.69	3.82
3	$\lambda_F^1 = 0.90$ $\lambda_F^2 = 1.145$ $\lambda_F^3 = 0.94$ $ \Delta\lambda_F^{1-3} = 0.04$	2.62	0.23	1.12

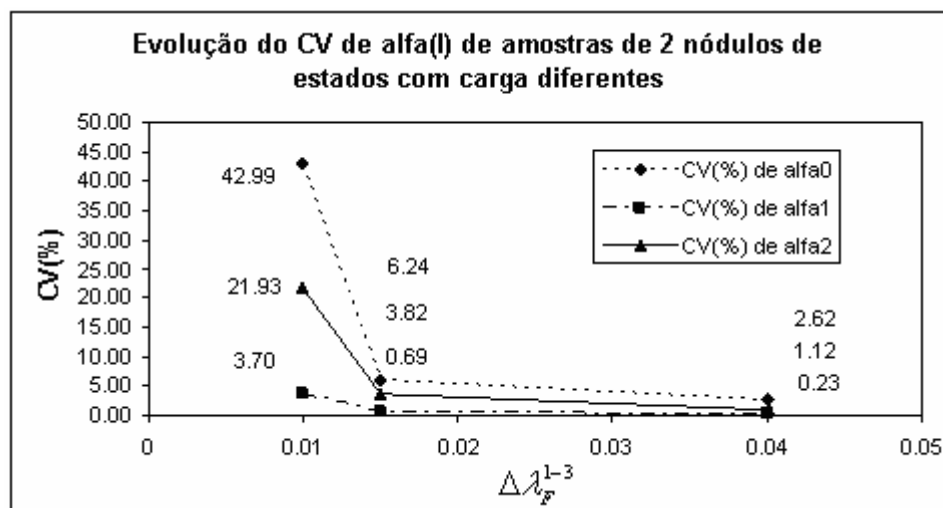


Figura 5.55 – Coeficiente de variação (%) de 4 amostras de 2 heterogeneidades vs $\Delta\lambda_F^{1-3}$, sendo para todos os casos $\lambda_F^1 = 0.90$ e $\lambda_F^2 = 1.145$

Da Figura 5.55 se infere que é preciso haver uma distância mínima entre os valores de deformações para não perturbar os resultados dos coeficientes alfas.

Determinação do EVR para o material estudado

Avaliando os resultados apresentados nas figuras 5.31 a 5.35 seria possível determinar o EVR em 8 nódulos, mas como foi mostrado na seção anterior, valores de deformação em estado triaxial e também muito próximos implicam em valores de coeficientes alfas com elevadas flutuações. Como este problema foi detectado, mas, em algum dos casos analisados, mantiveram-se estas condições de proximidade e triaxialidade nas deformações, para obter valores confiáveis de EVR foi utilizado um elemento de volume representativo de 32 nódulos, pois, como se observa nas figuras 5.53 e 5.54, para esta quantidade de nódulos, ainda que para coeficientes alfas calculados com valores de deformações próximos e com situação de triaxialidade, os resultados apresentaram CVs razoáveis (menores que 5%).

Tamanho da malha dos modelos analisados

A tabela 5.10 mostra o número de nós e de elementos das malhas de cada um dos modelos analisados, de 2, 4, 8, 16, 32 e 64 heterogeneidades, onde cada uma delas contém 30 nós e 74 elementos.

Tabela 5.10 – Número de nós e elementos das malhas dos modelos estudados

Número de heterogeneidades	Números de nós	Número de elementos
2	5436	26763
4	9639	48833
8	18978	98573
16	36836	195724
32	66180	354924
64	126850	689128

Tempo de processamento

O tempo total de processamento de todas as amostras dos modelos (2, 4, 8, 16, 32 e 64 heterogeneidades) foi de aproximadamente 130 horas. Os modelos foram executados no laboratório do GMAp/UFRGS.

5.2.7 Aplicações com o EVR determinado

Após a determinação do EVR, neste caso, 32 heterogeneidades, foram realizadas duas aplicações, ensaios de tração uniaxial e corte puro, realizadas no Abaqus/Simulia, 2009, com o objetivo de verificar a metodologia empregada na determinação do EVR.

5.2.7.1 Aplicação 1: Teste de tração uniaxial

Neste teste, aplicou-se ao EVR uma tração uniaxial, como ilustrado na Figura 5.56, com deformação máxima de 19% referente ao estiramento $\lambda_F^3 = 1.19$ que corresponde a uma deformação $\lambda_E^3 = 0.20805$, de onde se traçou a curva *tensão x deformação* (T_2^3 x λ_E^3), sendo T_2^3 a tensão de Piola-Kirchhoff de segunda espécie na direção de aplicação do deslocamento (3), determinado pela expressão (2.56). Através desta deformação λ_E^3 imposta, os outros valores de deformação λ_E^1 e λ_E^2 são naturalmente obtidos pelos deslocamentos nos nós das faces destas direções. Observou-se que estas duas últimas deformações variaram nos intervalos $\lambda_E^1 = [0, -0.03767]$ e $\lambda_E^2 = [0, -0.03831]$. Com estes valores de deformação

obtiveram-se nos mapas, semelhantes aos apresentados na figura 5.30, os valores de α_i para o caso de 32 nódulos.

Assim, para estes α_i as tensões T_2^3 foram determinadas pela resolução da expressão (5.12) de onde se tem

$$T_2^3 = \alpha_0 + \alpha_1 \lambda_E^3 + \alpha_2 (\lambda_E^3)^2 \quad (5.21)$$

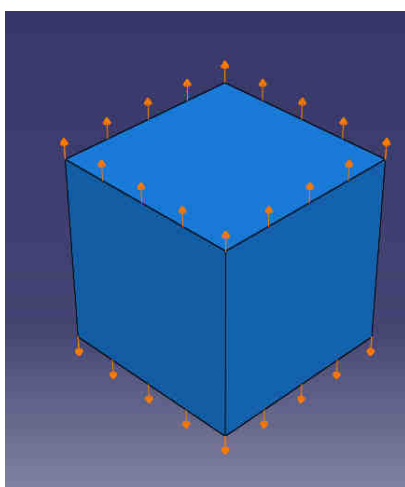


Figura 5.56 – Condições de contorno aplicadas no teste de tração uniaxial. Deslocamento prescrito aplicado em uma direção com as outras duas livres

Desta forma, pode-se obter qualquer valor de T_2^3 no domínio de deformações do ensaio, através dos valores de α_i mapeados. A figura 5.57 mostra a comparação entre as curvas tensão x deformação do teste de tração uniaxial do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades, conforme tabela 5.11. Verifica-se uma correlação muito boa entre as duas curvas, o que nos leva a concluir que a metodologia empregada apresenta resultados satisfatórios.

Cabe justificar que, para obter os valores de α_i foram realizadas interpolações lineares entre os valores de $\lambda_E^1 = -0.095$ e $\lambda_E^1 = 0$, como se observa na tabela 5.11, uma vez que o mapeamento foi gerado com quatro valores fixos dessa deformação, cujos valores são apresentados na expressão (5.14).

Tabela 5.11 – Valores de α_i e T_2^3 obtidos através da metodologia *material map* no domínio de deformações aplicado ao ensaio de tração uniaxial

λ_E^1	λ_E^2	λ_E^3 (λ_F^3)	Valores obtidos dos mapas de α_i						α_i interpolados			Tensão T_2^3
			λ_E^1 (fixo)=-0.095			λ_E^1 (fixo)=0			α_0	α_1	α_2	
			α_0	α_1	α_2	α_0	α_1	α_2				
0	0	0 (1.00)	0	0	0	0	0	0	0	0	0	0
- 0.0112	- 0.0113	0.0506 (1.05)	-7.9 $\times 10^{-5}$	2.17 $\times 10^{-3}$	-7.2 $\times 10^{-3}$	3.64 $\times 10^{-5}$	2.15 $\times 10^{-3}$	-6.0 $\times 10^{-3}$	2.29 $\times 10^{-5}$	2.15 $\times 10^{-3}$	-6.1 $\times 10^{-3}$	1.16 $\times 10^{-4}$
- 0.0207	- 0.0208	0.1058 (1.10)	-3.1 $\times 10^{-5}$	2.20 $\times 10^{-3}$	-6.5 $\times 10^{-3}$	5.46 $\times 10^{-5}$	2.15 $\times 10^{-3}$	-5.3 $\times 10^{-3}$	3.60 $\times 10^{-5}$	2.16 $\times 10^{-3}$	-5.6 $\times 10^{-3}$	2.02 $\times 10^{-4}$
- 0.0325	- 0.0316	0.1614 (1.15)	7.69 $\times 10^{-6}$	2.23 $\times 10^{-3}$	-6.2 $\times 10^{-3}$	1.00 $\times 10^{-4}$	2.20 $\times 10^{-3}$	-5.0 $\times 10^{-3}$	6.84 $\times 10^{-5}$	2.21 $\times 10^{-3}$	-5.4 $\times 10^{-3}$	2.84 $\times 10^{-4}$
- 0.0377	- 0.0383	0.2080 (1.19)	4.62 $\times 10^{-5}$	2.30 $\times 10^{-3}$	-5.8 $\times 10^{-3}$	1.50 $\times 10^{-4}$	2.20 $\times 10^{-3}$	-5.3 $\times 10^{-3}$	1.09 $\times 10^{-4}$	2.24 $\times 10^{-3}$	-5.6 $\times 10^{-3}$	3.34 $\times 10^{-4}$

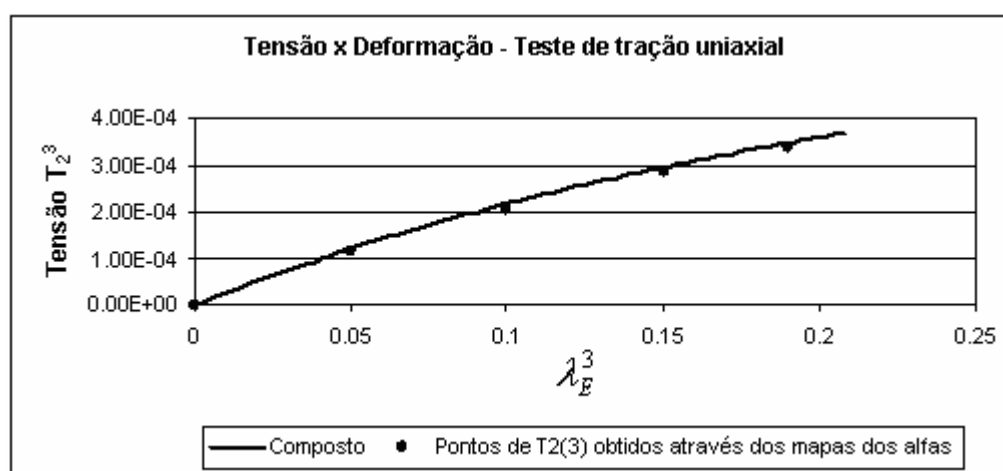


Figura 5.57 – Comparação entre as curvas *tensão x deformação* do teste de tração uniaxial do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades (Tabela 5.11)

A figura 5.58 mostra a comparação entre as curvas dos materiais envolvidos na análise, PMMA, elastômero e compósito resultante da mistura dos dois. Observa-se que a curva do compósito está posicionada de forma correta, entre as outras duas, ou seja, caracterizando-se, no domínio de deformações estudado, como um material com características intermediárias, entre a matriz elástica de PMMA e as heterogeneidades elastoméricas.

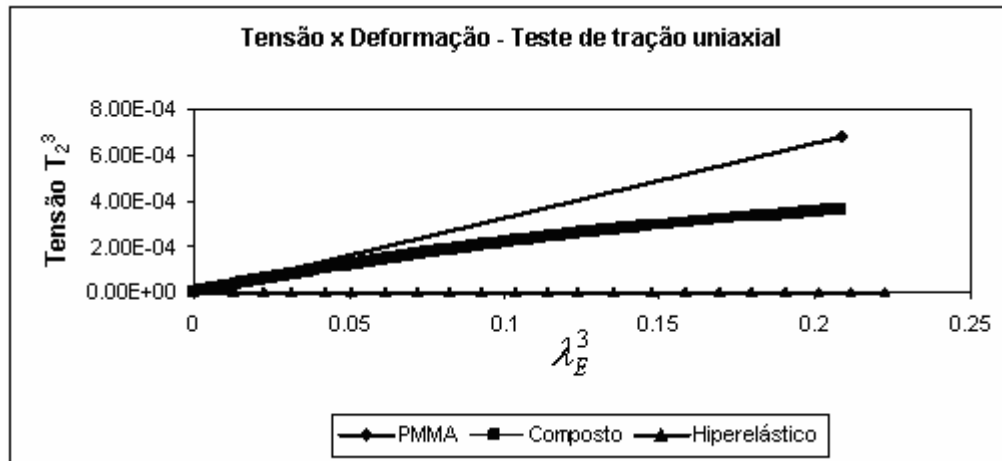


Figura 5.58 – Comparação entre as curvas *tensão x deformação* da matriz de PMMA, das heterogeneidades elastoméricas e do ensaio de tração uniaxial do material composto resultante

As figuras 5.59 e 5.60 apresentam detalhes da figura 5.58 em escala de tensão menor mostrando as inclinações das curvas de *tensão x deformação* nos níveis de deformação iniciais.

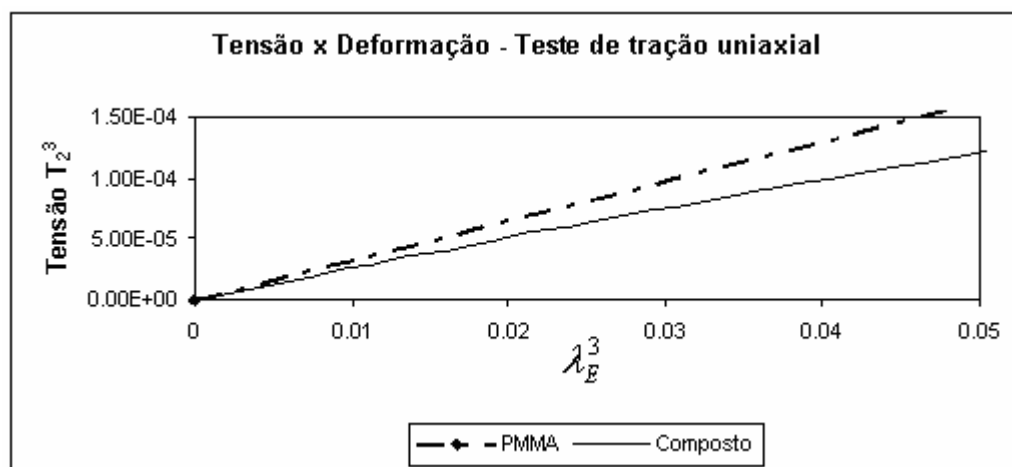


Figura 5.59 – Detalhe das inclinações iniciais das curvas de *tensão x deformação* da matriz de PMMA e do material composto resultante para o ensaio de tração uniaxial

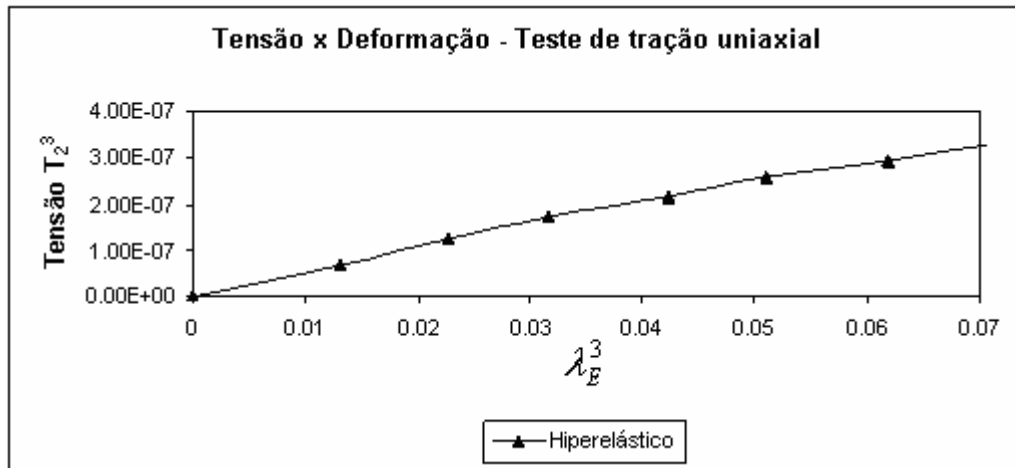


Figura 5.60 – Detalhe da inclinação inicial da curva de *tensão x deformação* das heterogeneidades elastoméricas

A figura 5.61 mostra o detalhe da curva elastomérica até o nível de deformação analisado.

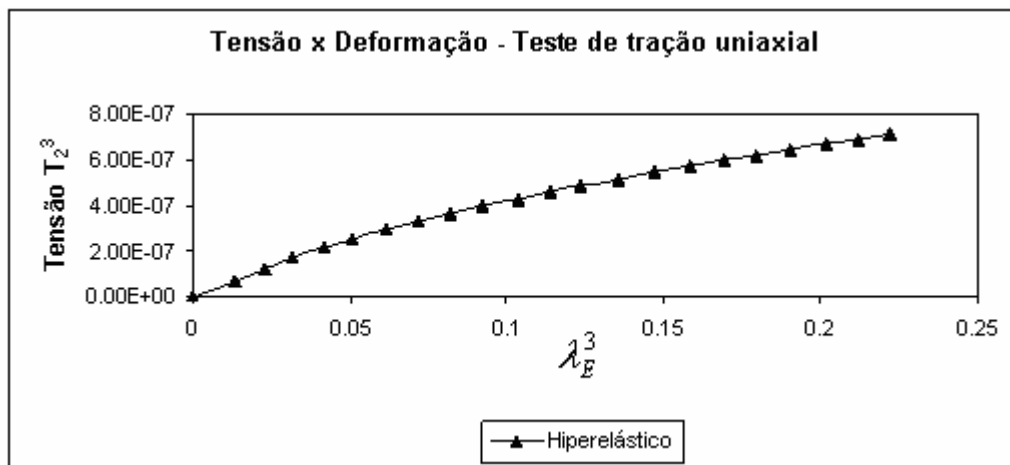


Figura 5.61 – Detalhe da curva do material elastomérico hiperelástico utilizado, até o nível de deformação analisado

5.2.7.2 Aplicação 2: Teste de corte puro

Foi aplicada ao EVR uma tração em uma direção e uma restrição em outra conforme a figura 5.62. Da mesma forma que no caso da aplicação interior, a deformação máxima foi de 19%, referente ao estiramento $\lambda_F^3 = 1.19$ que corresponde a uma deformação $\lambda_E^3 = 0.20805$. Nesse caso, $\lambda_F^1 = 1$ e, conseqüentemente, $\lambda_E^1 = 0$.

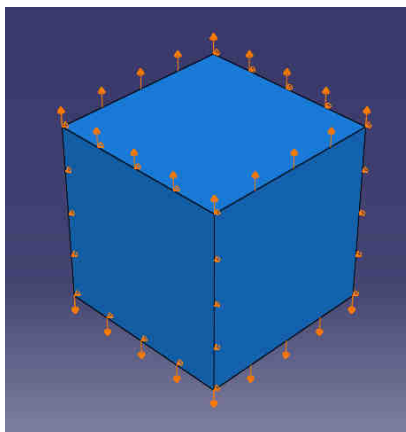


Figura 5.62 – Condições de contorno aplicadas no teste de corte puro. Deslocamento prescrito aplicado em uma direção, restrição em uma direção e a terceira livre

Os valores da deformação na direção (2) ficaram entre $\lambda_E^2 = [0, -0.04845]$. A tabela 5.12 mostra os valores de α_i e T_2^3 obtidos e a figura 5.63 mostra a comparação entre as curvas tensão x deformação do teste de corte puro do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades. Também neste teste foi verificada uma correlação muito boa entre as duas curvas, permitindo reafirmar e eficácia da metodologia empregada.

Tabela 5.12 – Valores de α_i e T_2^3 obtidos através da metodologia *material map* no domínio de deformações aplicado ao ensaio de corte puro

λ_F^3	λ_E^1	λ_E^2	λ_E^3	Valores obtidos dos mapas de α_i			T_2^3
				λ_E^1 (fixo)=0			
				α_0	α_1	α_2	
1.00	0	0	0	2.86×10^{-5}	2.30×10^{-3}	-6.60×10^{-3}	0
1.05	0	-0.0138	0.0506	5.46×10^{-5}	2.25×10^{-3}	-5.67×10^{-3}	1.28×10^{-4}
1.10	0	-0.0273	0.1058	9.50×10^{-5}	2.30×10^{-3}	-5.67×10^{-3}	2.29×10^{-4}
1.15	0	-0.0393	0.1614	1.20×10^{-4}	2.30×10^{-3}	-5.33×10^{-3}	3.19×10^{-4}
1.19	0	-0.0484	0.2081	2.86×10^{-5}	2.30×10^{-3}	-6.60×10^{-3}	3.68×10^{-4}

Para este ensaio não foi necessário interpolar os valores de α_i , visto que estes valores foram mapeados para $\lambda_E^1 = 0$.

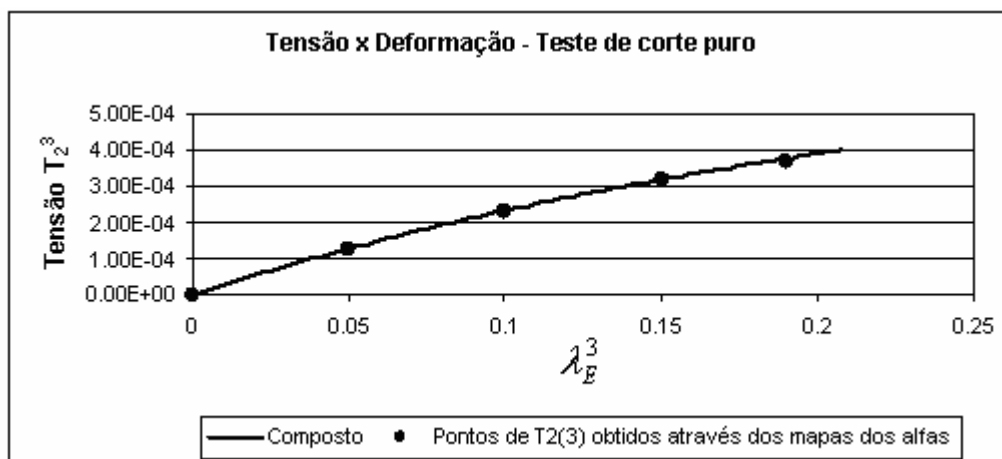


Figura 5.63 – Comparação entre as curvas *tensão x deformação* do teste de corte puro do EVR e os pontos de tensão obtidos através dos valores mapeados de α_i para os modelos de 32 heterogeneidades (Tabela 5.12)

A figura 5.64 mostra a comparação entre as curvas dos materiais envolvidos na análise, PMMA, elastômero e composto resultante da mistura dos dois. Observa-se que a curva do composto também está posicionada de forma correta neste ensaio, entre as outras duas.

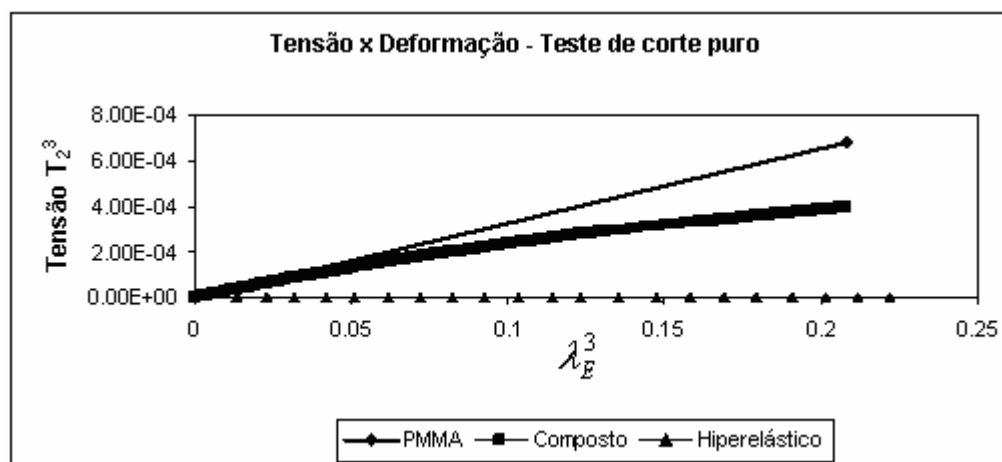


Figura 5.64 – Comparação entre as curvas *tensão x deformação* da matriz de PMMA, das heterogeneidades elastoméricas e do ensaio de corte puro do material composto resultante

A figura 5.65 apresenta detalhe da figura 5.64 em escala de tensão menor mostrando as inclinações das curvas de tensão x deformação nos níveis de deformação iniciais.

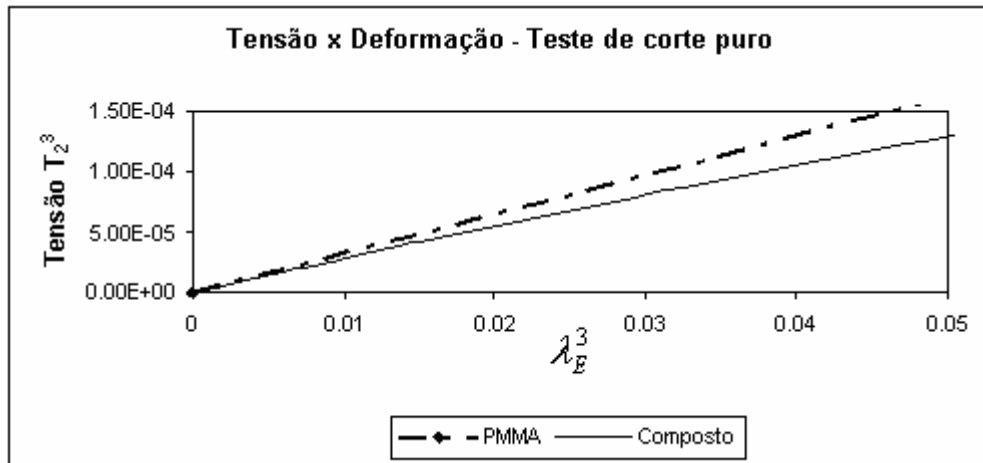


Figura 5.65 – Detalhe das inclinações iniciais das curvas de *tensão x deformação* da matriz de PMMA e do material composto resultante para o ensaio de corte puro

6 ESTUDOS REALIZADOS COM ALGORÍTMOS GENÉTICOS

Nesta seção serão apresentados os estudos de otimização realizados através da técnica dos Algoritmos Genéticos (AG) no elemento de volume representativo (EVR) de matriz polimérica termoplástica com heterogeneidades esféricas de elastômeros determinado por Soares et al., 2008. Especificamente, visou-se: a) determinar K e G das heterogeneidades fixando as propriedades da matriz e K^* e G^* do composto; b) pesquisar em uma lista de termoplásticos e elastômeros qual a melhor combinação que gerasse os K^* e G^* desejados; c) determinar K e G das heterogeneidades fixando as propriedades da matriz e minimizando a variação de tensões do composto.

Sendo assim, a seguir é apresentada a caracterização do modelo estudado, os procedimentos para a implementação do algoritmo genético e os resultados e discussões para a presente etapa.

6.1 Caracterização do modelo

Para esse estudo, adotou-se como modelo o EVR determinado em Soares et al., 2008, para uma fração de volume de nódulos de 10% em relação ao volume da amostra, que foi de 10 heterogeneidades. As propriedades da matriz eram $E=3240\text{MPa}$ e $\nu=0,25$, enquanto das heterogeneidades esféricas, de raio constante, $E=40\text{MPa}$ e $\nu=0,40$.

Cabe salientar ainda que, devido à característica elástica linear do modelo estudado, as diferentes distribuições aleatórias para a mesma fração de volume apresentaram muito pouca variação nos resultados obtidos para as constantes elásticas. Por esta razão, neste estudo optou-se por realizar uma simplificação no modelo a ser otimizado, no que se refere à distribuição das esferas na matriz, implementando-se uma distribuição ordenada das mesmas e com oito heterogeneidades. Desta forma evitou-se trabalhar com uma média de diferentes configurações aleatórias que geram resultados muito próximos.

Na Figura 6.1 apresenta-se a malha e a distribuição ordenada do modelo otimizado.

6.2 Implementação do algoritmo genético

A implementação da programação dos algoritmos genéticos nesta fase foi realizada com a linguagem de programação *Fortran*, cujo programa foi desenvolvido pelo Professor Dr.

Herbert Martins Gomes, do GMAp/UFRGS, e adaptado aos presentes casos durante o desenvolvimento da disciplina de Algoritmos Neurais e Evolutivos para a Engenharia, com o mesmo professor. Características da implementação realizada podem ser encontradas na seção 4.4.

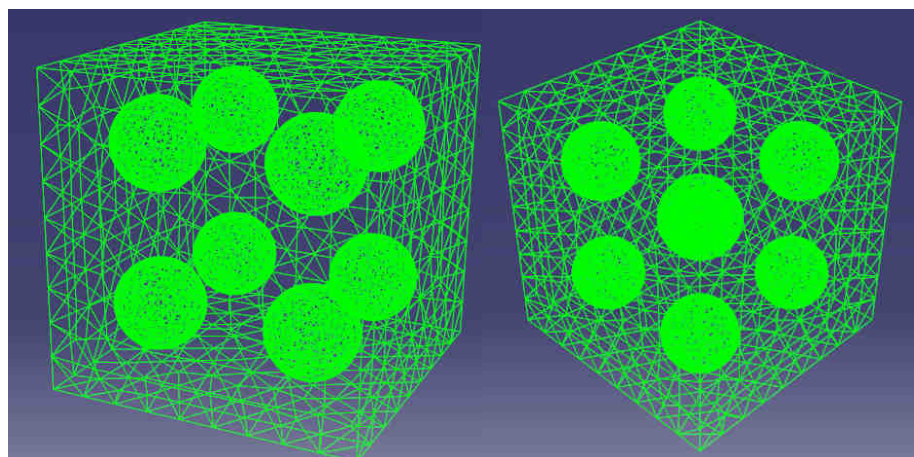


Figura 6.1 – Esquema mostrando malha e distribuição regular de 8 nódulos esféricos do modelo utilizado

6.3 Resultados e discussões da etapa

Foi então realizada uma otimização através da técnica dos Algoritmos Genéticos no modelo descrito anteriormente. Para tanto foram estudados os seguintes exemplos: Caso A: determinação de K e G das heterogeneidades através da fixação das propriedades da matriz e do composto; Caso B: busca, entre materiais reais termoplásticos e elastômeros [NDSM, 2009], da melhor combinação que gerasse as propriedades efetivas desejadas para o composto; Caso C: determinação de K e G das heterogeneidades fixando as propriedades da matriz e minimizando a variação de tensões do composto.

6.3.1 Caso A

Usando o cubo regular já descrito, fixou-se as propriedades da matriz e se deixou variável as das heterogeneidades. Procurou-se obter o K^* e o G^* do composto que se obteve com a distribuição aleatória das heterogeneidades. Devido ao fato de que as variações nos resultados das distribuições aleatórias foram muito pequenas é de se esperar que as propriedades a serem encontradas com a otimização sejam muito próximas às da borracha

utilizadas em Soares et al., 2008.

Para este caso, a função objetivo (*Fitness*) a maximizar, para cada indivíduo *i*, foi:

$$Fitness(i) = \frac{1}{1 + \frac{|K(i) - K^*|}{K^*} + \frac{|G(i) - G^*|}{G^*}} \quad (6.1)$$

Quando as diferenças entre o $K(i)$ e o $G(i)$ e os valores desejados K^* e G^* forem mínimas, ter-se-á a maior aptidão ou o valor desejado.

As propriedades efetivas desejadas para o composto em estudo são: $K^*=1784,86\text{MPa}$ e $G^*=1095,05\text{MPa}$.

Anteriormente ao cálculo da função objetivo, foi necessária a determinação das constantes elásticas do composto a partir das propriedades fixas da matriz e variáveis das heterogeneidades para cada indivíduo. Como o tempo computacional estava demasiado, uma vez que se fossem gerados indivíduos iguais eles seriam trabalhados novamente, optou-se por gerar um banco de dados a fim de se verificar a existência ou não do próximo indivíduo. Desta forma teve-se a garantia de que um mesmo indivíduo não foi analisado mais de uma vez.

Na tabela 6.1 apresenta-se os dados referentes ao presente caso. Indicam-se a quantidade de bits em que foi representada cada variável e seus intervalos de variação para uma população de 20 indivíduos. As outras variáveis necessárias para a implementação dos algoritmos genéticos utilizadas são: a probabilidade de crossover (P_c) de 90%, a probabilidade de mutação (P_m) de 10% e a tolerância máxima do desvio padrão da diversidade dos indivíduos (Tol) de 1.0×10^{-5} .

Tabela 6.1 – Número de bits e limites aplicados às variáveis de estudo do Caso A

Variável	Nº. de bits	Limite inferior	Limite superior
v das heterogeneidades	5	0,0	0,5
E das heterogeneidades	9	1,0 MPa	4000,0 MPa

A partir dos valores de K e G , as propriedades das heterogeneidades obtidas com este algoritmo foram: $E=47,955\text{MPa}$ e $v=0,403$. Pode-se observar a semelhança para com os valores esperados ($E=40\text{Mpa}$, $v=0,40$).

Com as características apresentadas na tabela 6.1 tem-se 16384 combinações possíveis

de E e ν . Com a otimização implementada foram necessárias 26 gerações, o que equivale a gerar 540 indivíduos (considerando a população inicial). No entanto, com a implementação do banco de dados foi preciso rodar somente os 89 indivíduos diferentes que foram gerados durante o processo, o que representou uma redução de 6 vezes no tempo computacional.

Cabe ainda ressaltar a vantagem do algoritmo genético de onde foram necessárias apenas 0,5% do total das combinações possíveis para encontrar o valor desejado.

6.3.2 Caso B

Como não é possível conseguir materiais que tenham relações quaisquer entre as propriedades elásticas, procurou-se encontrar quais são as combinações de materiais reais, com 10% de heterogeneidades, que podem levar a um determinado valor de E.

Nesta análise, tanto a matriz como as heterogeneidades podiam assumir valores de uma tabela de materiais. Nesta lista, apresentada na tabela 6.2, são especificados o E e o ν da cada material. A matriz poderia ser constituída de um dos 32 materiais termoplásticos relacionados, enquanto as esferas de 7 elastômeros ou vazios (estes representados por propriedades praticamente nulas).

Tabela 6.2 – Lista de polímeros utilizada no estudo do caso B. Valores médios [NDSM, 2009]

MATRIZ TERMOPLÁSTICA OU TERMOFIXA								
Material	E(MPa)	ν	Material	E(MPa)	ν	Material	E(MPa)	ν
1-ABS-Alto Impacto	900	0,4	15-PI	4225	0,4375	29-Fenólicos	6437,5	0,40625
2-ABS-Medio Impacto	1350	0,4	16-PMMA	2150	0,4075	30-Fenólicos	7312,5	0,41875
3-Nylon 11	925	0,4	17-PMMA	2850	0,4225	31-Fenólicos	8187,5	0,43125
4-Nylon 6	2500	0,4	18-Poliéster Rígido	2975	0,4125	32-Fenólicos	9062,5	0,44375
5-Nylon 6/6	2400	0,4	19-Poliéster Rígido	3925	0,4375	INCLUSÕES ELASTÔMERAS		
6-PBT	2150	0,375	20-PP	1987,5	0,425	Material	E(MPa)	ν
7-PC	2450	0,415	21-PPO	2287,5	0,375	1-Epdm	53,025	0,48375
8-PEAD	1200	0,41	22-PS	3100	0,415	2-Epdm	157,675	0,49125
9-PEBD	650	0,44	23-PTFE (teflon)	500	0,455	3-EVA	25	0,48
10-PEEK	725	0,405	24-PU	16	0,494	4-Isopreno	1,85	0,49925
11-PES	2650	0,4	25-PVC Rígido	3150	0,4	5-Neoprene	1,35	0,4925
12-PT	2525	0,3925	26-SAN	3175	0,385	6-SBR	6	0,488
13-PT	3175	0,4175	27-Alkydes	7500	0,4125	7-Silicone Rígido	5,5	0,385
14-PI	3075	0,4125	28-Alkydes	8500	0,4375	8-Vazio	1.0×10^{-7}	1.0×10^{-7}

Para este estudo a função objetivo (*Fitness*) a maximizar, para cada indivíduo i , foi:

$$Fitness(i) = \frac{1}{1 + \frac{|E(i) - E^*|}{E^*}} \quad (6.2)$$

A melhor combinação de materiais para a matriz e heterogeneidades seria aquela na qual a diferença entre o $E(i)$ e o valor desejado E^* fosse mínima. A propriedade efetiva procurada neste caso era $E^* = 2000\text{MPa}$.

Na tabela 6.3 são apresentados os dados referentes ao presente caso.

Da mesma forma que no caso anterior, antes de calcular a função objetivo foi necessário chamar o programa de elementos finitos para calcular as propriedades do composto a partir, agora, das propriedades variáveis da matriz e das heterogeneidades em cada indivíduo.

Tabela 6.3 – Número de bits e limites aplicados às variáveis de estudo do Caso B

Número de indivíduos:	20
Variável 1	5 bits lista para matriz: 32 materiais
Variável 2	3 bits lista para heterogeneidades: 8 materiais
Pc	90%
Pm	10%
Tol	1.0×10^{-5}

Também é importante salientar que se aproveitou o banco de dados implementado. Neste caso, como um cromossomo não representava mais uma propriedade elástica do indivíduo, mas sim um número identificador de um determinado material na lista, foi necessário realizar pequenas alterações na programação.

A melhor combinação resultante foi entre os materiais 21 (PPO = Polyphenylene oxide) e 2 (Epdm = ethylene propylene diene M-class rubber) para a matriz e inclusos, respectivamente. O valor encontrado para o $E(i)$ com estes materiais foi 1994MPa , representando um erro de 0,3% em relação ao valor desejado que era 2000MPa .

Entre os materiais propostos tem-se 256 combinações possíveis. Com a otimização implementada foram necessárias 319 gerações, com 130 indivíduos diferentes. O valor ótimo foi encontrado na 2ª geração e as demais foram necessárias para cumprir com o critério de diversidade imposto.

Foram necessárias pouco mais de 50% do total das combinações possíveis para encontrar o valor desejado, mas no momento em que este foi encontrado só haviam sido rodadas 13% das combinações possíveis.

Posteriormente, o mesmo estudo foi implementado para 6 e 10 indivíduos. Na figura 6.2 observa-se o erro do valor obtido em relação ao desejado para cada número de indivíduos. Também nota-se como variou para cada um deles o percentual das combinações realizadas em relação às possíveis. No gráfico da figura 6.2 é possível observar que, na medida em que se diminui a quantidade de indivíduos da população inicial diminui a quantidade de combinações realizadas em relação às possíveis (curva cinza) reduzindo assim o tempo de processamento necessário. Em contrapartida o valor encontrado fica mais distante do mais apto (curva vermelha). Considerando a população inicial de 6 indivíduos mas utilizando uma pm de 30% o valor encontrado aproxima-se muito do desejado (erro de 0,3%) sendo o número de gerações igual a 101 e o número de indivíduos diferentes 65 (25,4% das combinações possíveis). Mostra-se que é possível diminuir o erro com uma redução do número de indivíduos, mas em contrapartida isto acarreta um aumento no número de gerações e combinações necessárias (em torno do dobro).

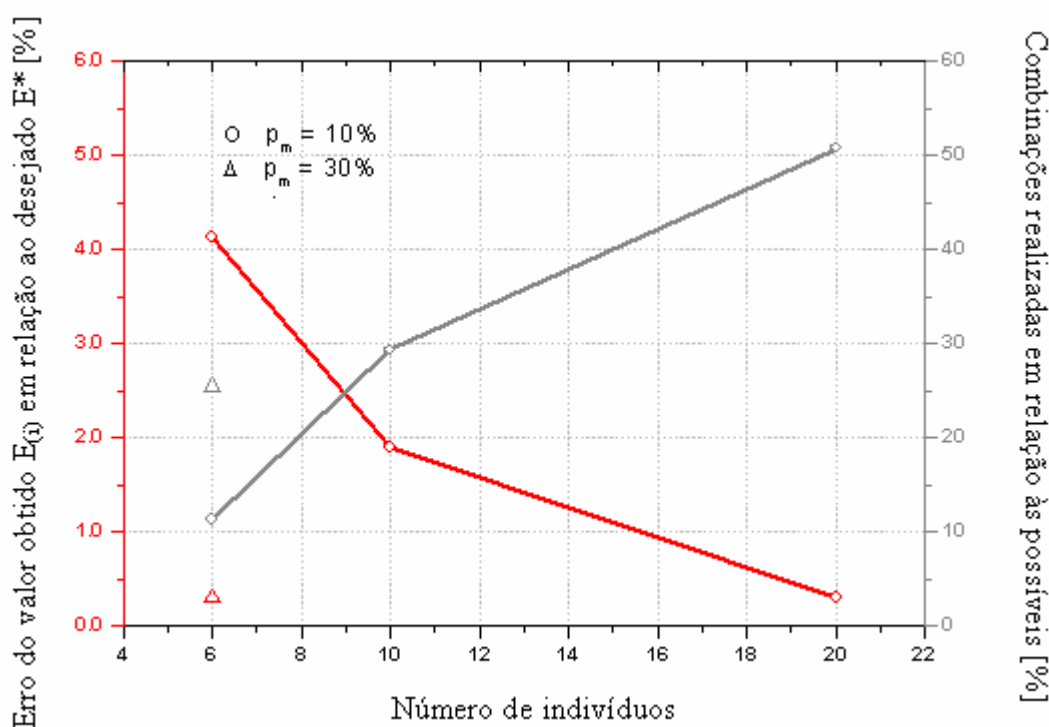


Figura 6.2 – Gráfico mostrando o erro do valor obtido em relação ao desejado e a variação do percentual das combinações realizadas em relação às possíveis, para cada número de indivíduos da população inicial

Também se pode agregar à função objetivo, variáveis como: peso, custo, leveza, ductilidade, resistência a altas temperaturas, condutividade elétrica, tenacidade, entre outros, de cada um dos materiais. Desta forma pode-se procurar, por exemplo, a melhor combinação de materiais que gere um E desejado com peso e custo mínimos.

6.3.3 Caso C

O terceiro exemplo apresentado neste trabalho serve de base para futuras implementações. Uma vez definidas as propriedades das duas fases do composto para maximizar ou minimizar alguma característica ou conjunto delas, pode-se prever modificar as formas, ângulos e distribuição das heterogeneidades, por exemplo, para minimizar as tensões de contato entre os dois materiais diminuindo ou anulando as possibilidades de acontecimento de falhas.

Diante disto, procurou-se quais as propriedades das heterogeneidades que tornassem mínimas as variações de tensões. Assim, o esperado era que, para que não houvesse variações de tensões, as propriedades da matriz e do composto fossem as mesmas.

Neste exemplo a função objetivo a maximizar para cada indivíduo i foi:

$$Fitness(i) = \frac{1}{1+CV} \quad (6.3)$$

onde CV é o coeficiente de variação calculado a partir da média e do desvio padrão das tensões de Von Mises de cada um dos elementos que compõem o modelo.

Os dados deste estudo podem ser visualizados na Tabela 6.1. Assim como no Caso A, utilizou-se uma população de 20 indivíduos, $p_c=90\%$, $p_m=10\%$ e $tol=1.0 \times 10^{-5}$.

As propriedades das heterogeneidades obtidas com este algoritmo foram: $E=3201,77\text{MPa}$ e $\nu=0,2581$, com média de tensões de Von Mises de $2,128 \times 10^{-4}\text{N}/\mu\text{m}^2$, desvio padrão de $1,305 \times 10^{-6}\text{N}/\mu\text{m}^2$ e CV de 0,61%, o que representa a pequena dispersão destas últimas. Pode-se observar a semelhança para com os valores esperados ($E=3240\text{Mpa}$, $\nu=0,25$).

Na figura 6.3(a) apresenta-se um corte com a distribuição de tensões de Von Mises do modelo regular de 8 heterogeneidades com as mesmas propriedades do estudo do EVR de Soares et al., 2008.

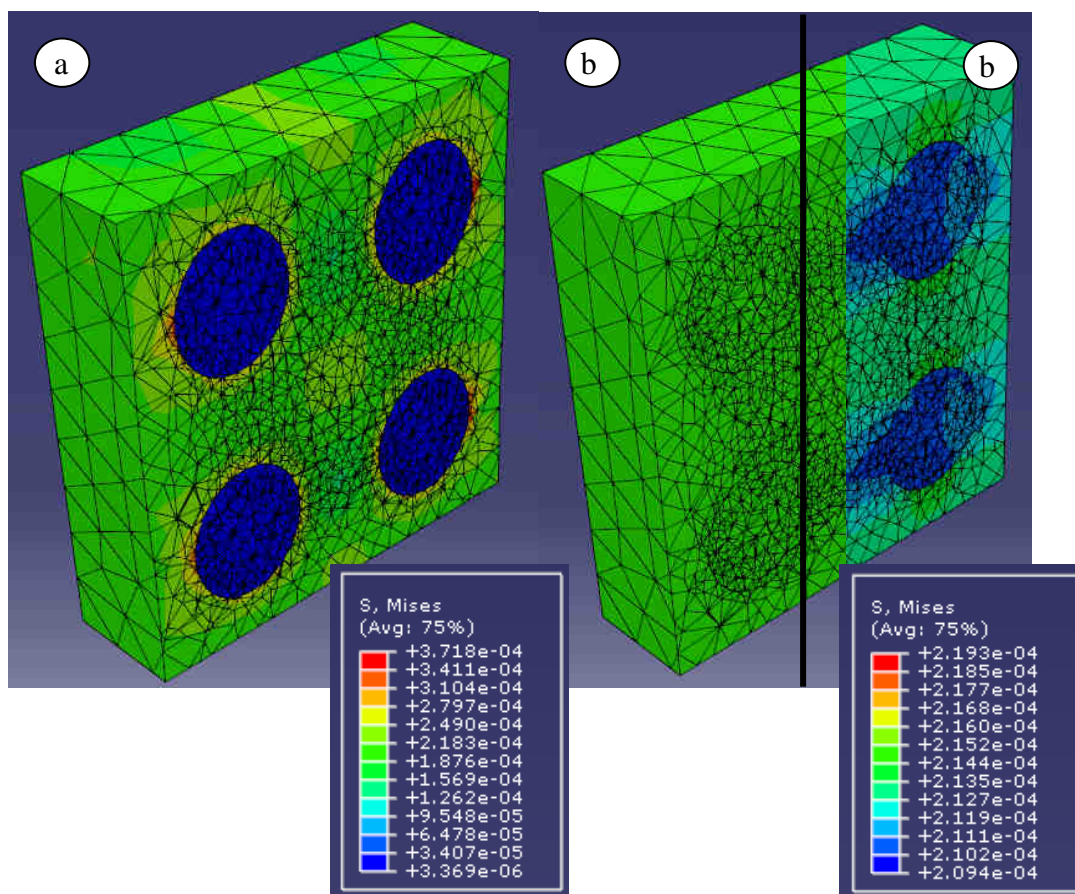


Figura 6.3 – Distribuições de tensões de Von Mises no modelo regular de 8 nódulos esféricos, a) propriedades do estudo do EVR de Soares et al., 2008; b) propriedades otimizadas para minimizar as tensões, em diferentes escalas

Observa-se que as tensões variam de $3,369 \times 10^{-6} \text{N}/\mu\text{m}^2$ a $3,718 \times 10^{-4} \text{N}/\mu\text{m}^2$. Na figura 6.3(b1), que corresponde ao modelo com as propriedades otimizadas, não se observam variações de tensões na mesma escala, mas na figura 6.4(b2), quando se muda a escala de representação, é possível observar a pequena variação de tensões.

Da mesma forma que no primeiro caso, tem-se 16384 combinações possíveis de E e ν . Foram necessárias 49 gerações, o que equivale a gerar 1000 indivíduos (considerando a população inicial) sendo 200 deles diferentes. Salienta-se ainda que com os algoritmos genéticos só foram necessárias 1,2% do total das combinações possíveis para encontrar o valor desejado.

7 ESTUDOS REALIZADOS COM ELEMENTOS DISCRETOS

Nesta seção serão descritos os procedimentos adotados para a determinação do elemento de volume representativo (EVR) de um modelo de micro-estrutura, bifásico particulado, implementado com o emprego do Método dos Elementos Discretos (MED). A matriz é constituída de PMMA enquanto a segunda fase é representada por heterogeneidades esféricas com propriedades similares à borracha, que por sua vez contém, no seu interior, uma heterogeneidade, também esférica, de PMMA (chamadas de “core shell”), conforme se pode observar na figura 7.1. Os parâmetros de análise para determinação do EVR foram as convergências dos valores da densidade de energia de deformação elástica (ENEL/V) e da densidade de energia de dano (ENGD/V) para os regimes linear elástico, não linear e de tensão máxima, onde ENEL é a energia elástica, ENGD é a energia de dano e V o volume da amostra. Foram utilizadas configurações aleatórias para caracterizar o composto considerando o valor médio e desvio da propriedade estudada.

Após esta breve introdução, tem-se na continuação a apresentação da caracterização do modelo, os procedimentos para determinação do EVR e os resultados obtidos.

7.1 Caracterização do modelo

O modelo estudado consiste de uma célula cúbica de polimetilmetacrilato (PMMA) linear elástica contendo uma heterogeneidade esférica de borracha que, por sua vez, é constituída de uma outra esfera de PMMA (partícula do tipo “core shell”), (Figura 7.1).

As propriedades físicas dos materiais que compõem o modelo são mostradas na tabela 7.1 a seguir:

Tabela 7.1 – Propriedades físicas dos materiais do modelo em elementos discretos

Propriedade	PMMA	Borracha
E (MPa)	3240	1450
ν	0,25	0,25
Gf (N/m ²)	200	10000
σ_c (MPa)	60	20
ρ (Kg/m ³)	1150	1230

Onde:

E : módulo de elasticidade longitudinal;

ν : coeficiente de Poisson;

G_f : energia específica de fratura

σ_c : tensão de ruptura

ρ : massa específica.

As barras situadas na interface matriz/borracha tiveram os valores de suas propriedades adotados como o equivalente a 50% dos valores da borracha.

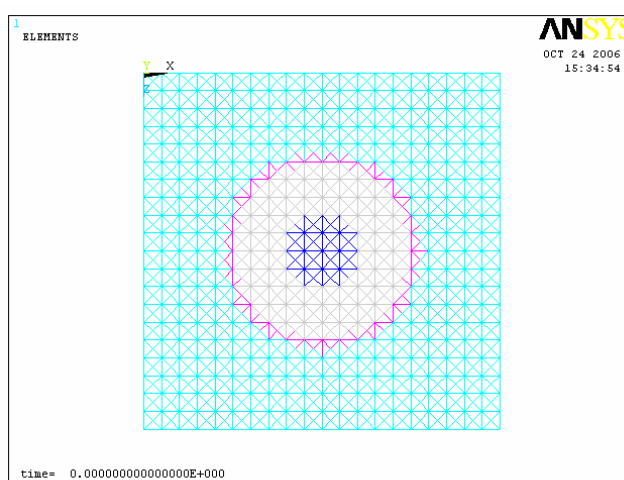


Figura 7.1 – Corte do modelo cúbico com uma heterogeneidade esférica de borracha onde a região: em azul é um nódulo interno de PMMA (raio $r=10 \times 10^{-9}$ m); em cinza é borracha ($r=25 \times 10^{-9}$ m); em rosa é a interface matriz/borracha ($r=27.5 \times 10^{-9}$ m) e o restante é a matriz cúbica (lado $L=100 \times 10^{-9}$ m)

7.2 Determinação do elemento de volume representativo (EVR)

O critério aqui adotado para determinação do EVR também foi a condição de Hill, descrita pela equação (5.1). No entanto, o estado de carga adotado para satisfazer a condição de Hill foi o teorema das tensões médias, conforme a equação (5.4). Ao contrário dos estudos com MEF, em que não foi permitido o despegue entre matriz e heterogeneidades, na aplicação do MED esta situação foi permitida, através da caracterização específica das barras da região de interface (Figura (7.1)). Em decorrência disso, não foi possível aplicar o teorema da deformação média, e a condição adequada foi a de carregamento prescrito.

Para a obtenção do EVR foi realizada uma série de análises no MED no modelo cúbico contendo um número crescente de nódulos (2, 4, 10, 15, 20, 30, 50).

Manteve-se constante e em 5% a densidade de nódulos em relação ao volume total do modelo. Como o raio dos nódulos foi mantido constante, os tamanhos dos modelos no MED foram tornando-se cada vez maiores, constituindo-se de 4913 elementos no caso de 2 nódulos e 68921 elementos para a configuração com 50 nódulos. Os modelos foram submetidos a um estado uniaxial de tração prescrita (linear crescente), e analisados utilizando a lei constitutiva bilinear. O tamanho de elemento utilizado foi de $L_c = 5 \times 10^{-9}$, o que corresponde a $0,2r$ (raio da heterogeneidade esférica de borracha). As variáveis consideradas para a determinação do EVR foram a densidade de energia de deformação elástica (ENEL/V) e a densidade de energia de dano (ENGD/V) para os regimes indicados na figura 7.2.

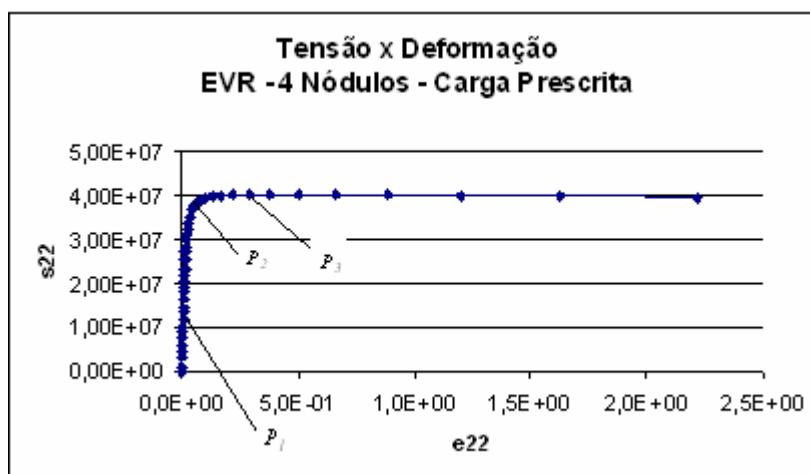


Figura 7.2 – Curva Tensão x Deformação indicando os pontos de análise, $\sigma_c = 60MPa$

Onde:

$P_1=P_A$: regime linear elástico - ENEL;

$P_2=P_B$: regime não-linear - ENGD;

$P_3=P_C$: tensão máxima - ENGD.

Foi realizada uma interpolação entre os valores de tensão e deformação de modo que fosse assegurado que os valores das energias fossem tomados sempre para a mesma deformação nos diferentes níveis de carga.

Para cada número de nódulos analisado foram realizadas 5 configurações com disposição aleatória dos nódulos e os resultados foram tratados estatisticamente, onde foram considerados os valores médios das energias e seus respectivos desvios padrões.

O método foi gerenciado na linguagem de programação *Fortran* com auxílio do software de elementos finitos *Ansys*, sendo que as configurações aleatórias das

heterogeneidades geradas em *Matlab*.

7.3 Resultados e discussões da etapa

A seguir são apresentados e discutidos os resultados desta etapa do trabalho.

7.3.1 Convergência dos parâmetros ENEL/V e ENGD/V

As figuras 7.3, 7.4 e 7.5 mostram a convergência da resposta do material (parâmetros ENEL/V e ENGD/V) com o incremento do número de nódulos para amostras com volume de nódulos de 5% em relação ao tamanho do modelo, para os níveis de carga indicados na figura 7.2. $(ENEL/V)'$ é o valor de ENEL/V médio para a amostra de 50 heterogeneidades e $(ENGD/V)'$ é o valor de ENGD/V médio também para a amostra de 50 heterogeneidades. As barras de erros, nesse caso imperceptíveis, correspondem à dispersão das configurações, de onde se pode observar, através da tabela 7.2, a tendência de redução dos valores à medida que eram inseridas heterogeneidades na matriz. Se um erro de 1% em relação à maior amostra (50 heterogeneidades) for admitido, estes resultados permitem afirmar que no regime linear uma amostra com 20 heterogeneidades para uma $c_I = 5\%$ é representativa (Figura 7.3).

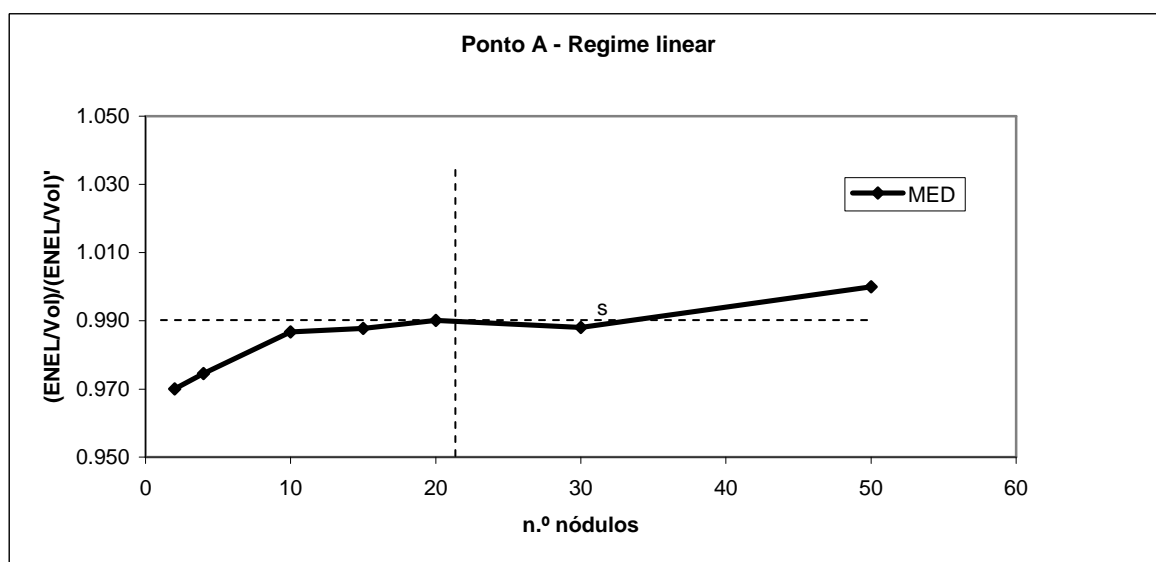


Figura 7.3 – Resultados da densidade de energia elástica para amostras contendo número crescente de nódulos de borracha, regime linear elástico (P_1). As barras de erros indicam a dispersão nos resultados e $(ENEL/V)'$ é o valor de ENEL/V médio para a amostra de 50 heterogeneidades

No entanto, se a análise for estendida para o regime não-linear, este erro admissível não pode ser aplicado com a análise até 50 heterogeneidades. Pode-se reparar, na figura 7.4, que a amostra de 20 heterogeneidades, que é representativa na análise linear, no caso não-linear foi gerada com um erro de aproximadamente 50%. Pode se notar também que os valores da densidade de energia de dano apresentam um aspecto instável e com grande diferença entre eles.

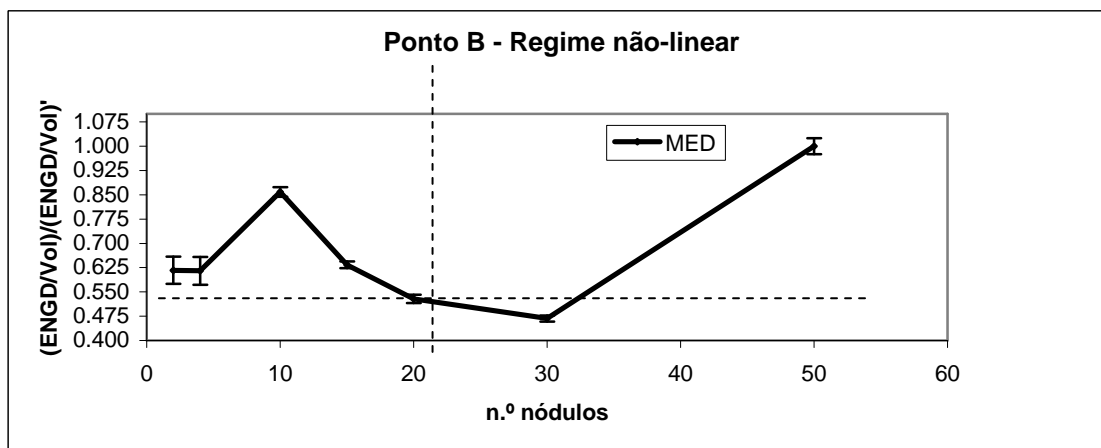


Figura 7.4 – Resultados da densidade de energia de dano para amostras contendo número crescente de nódulos de borracha, para o regime não-linear (P_2). As barras de erros indicam a dispersão nos resultados, e $(\text{ENGD}/V)'$ é o valor de ENGD/V médio para a amostra de 50 heterogeneidades

No regime de tensão máxima (Figura 7.5) observa-se uma instabilidade nos modelos com poucas heterogeneidades e uma tendência de estabilização a partir de 20 nódulos, onde se vê que entre 20 e 30 heterogeneidades os erros nos parâmetros de análise estão em torno de 5% e 3% respectivamente, em relação aos valores da maior amostra.

Tabela 7.2 – Tabela dos desvios indicando a dispersão dos resultados

Nº. de nódulos	Ponto A (%)	Ponto B (%)	Ponto C (%)
2	0,01	6,76	2,88
4	0,02	7,04	2,18
10	0,02	1,71	1,15
15	0,02	1,64	1,15
20	0,02	2,43	0,49
30	0,00	1,99	0,43
50	0,00	2,46	0,63

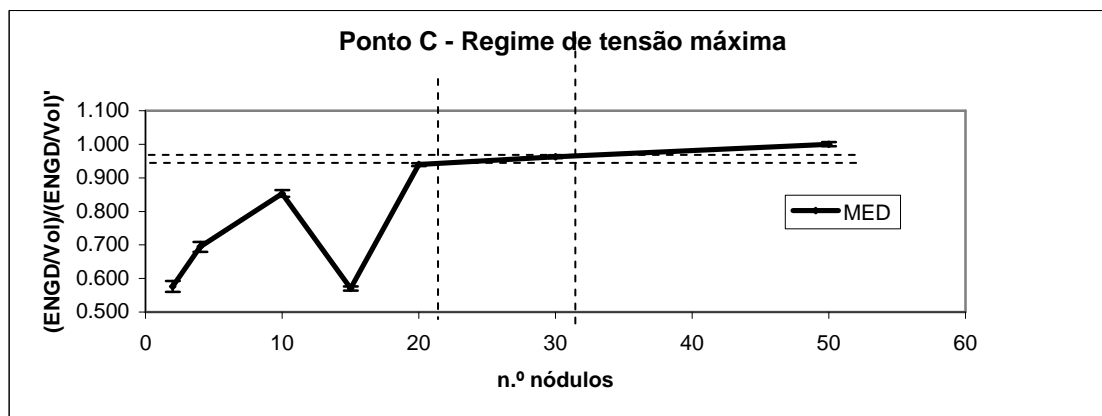


Figura 7.5 – Resultados da densidade de energia de dano para amostras contendo número crescente de nódulos de borracha, para o regime de tensão máxima (P_3). As barras de erros indicam a dispersão nos resultados, e $(\text{ENG D/V})'$ é o valor de ENG D/V médio para a amostra de 50 heterogeneidades

8 CONCLUSÕES

8.1 Conclusões da etapa de estudos com MEF

8.1.1 Caso elástico linear

Foi apresentada no capítulo 5, seção 5.1, uma aplicação do método dos elementos finitos na homogeneização das propriedades mecânicas elásticas de um material composto particulado com distribuição aleatória. O estudo foi realizado sobre uma matriz polimérica de PMMA com heterogeneidades esféricas similares a um elastômero. O material foi considerado elástico linear e não foi considerado o descolamento entre matriz e nódulos. Os resultados do estudo apontam para as seguintes conclusões:

- O estudo se mostrou adequado para a determinação do EVR. Pode-se observar a convergência dos valores das constantes elásticas à medida que eram inseridas heterogeneidades na matriz, bem como a redução da dispersão dos resultados, indicada pelas barras de erros nos gráficos das figuras 5.7 e 5.8.

- Foi realizado um estudo de erro admissível para os valores das constantes elásticas em relação aos valores da maior amostra. Os resultados mostraram-se coerentes, uma vez que, ao passo que o erro era maior, menor era o EVR (Figura 5.9).

- Também foi estudada a variação do EVR em função da fração de volume de partículas do composto (c_I). Os resultados mostraram que, para um erro de 1%, considerando os valores de G , o EVR é muito sensível à variação de volume de heterogeneidades, haja vista que, para 5% de heterogeneidades tem-se um EVR de 2 nódulos, para 10%, EVR de 10 nódulos e, para 20%, o EVR é de 28 heterogeneidades (Figura 5.10). Já para a análise dos valores de K a sensibilidade é menor: EVR de 2 nódulos para frações de volume de 5% e 10%, e de 6 nódulos esféricos para 20% de heterogeneidades, considerando também erro de 1% em relação à maior amostra.

- Foram realizadas comparações dos resultados do MEF para os EVRs obtidos com erros de 1% (2, 10 e 28 nódulos para $c_I=5%$, $c_I=10%$ e $c_I=20%$ respectivamente), com modelos teóricos clássicos da micro-mecânica. Pode-se verificar a observância dos limites de Voigt e Reuss para os dois módulos elásticos analisados (Figuras 5.11 e 5.13), bem como uma boa aproximação dos valores de Mori-Tanaka, DILUTE, Auto-consistente e Diferencial em relação ao MEF (Figuras 5.12 e 5.14), especialmente para c_I menores. À medida que se

aumentou esta fração volumétrica os valores do modelo DILUTE tenderam a se afastar como era de se esperar, já que este método não leva em conta a interação entre as heterogeneidades. Comportamento semelhante ocorreu com os métodos auto-consistente e diferencial, uma vez que derivam do DILUTE.

- Finalmente, realizou-se uma avaliação sobre a influência da distância mínima entre as heterogeneidades nos valores dos parâmetros elásticos K e G . Concluiu-se que para distâncias mínimas entre as superfícies dos nódulos que variam de 0,5% a 25% do raio dos mesmos não há influência significativa, uma vez que a maior variação nesse domínio de análise foi da ordem de 2% para o parâmetro G .

8.1.2 Caso elástico não-linear

Apresentou-se no capítulo 5, seção 5.2, uma aplicação do método dos elementos finitos na homogeneização numérica elástica não-linear de um composto com distribuição aleatória de heterogeneidades esféricas hiperelásticas em matriz de PMMA. O estudo foi realizado aplicando-se a técnica apresentada por Temizer e Zohdi, 2007, em que o comportamento do material é caracterizado por um estado de tensão-deformação e também pelo conceito de *material map*, porém em uma análise tridimensional. Os resultados do estudo apontam para as seguintes conclusões:

- Os parâmetros adotados para determinação do EVR (índices I_0 , I_1 e I_2) mostraram-se adequados. Foi possível obter a convergência dos valores à medida que se aumentou o tamanho das amostras, como indicam as figuras 5.31, 5.32 e 5.33.

- Foi verificada uma alta sensibilidade da matriz formada por valores de deformações λ_E^i quando estes são muito próximos entre si e no estado hidrostático. Por esta razão, o valor adotado para separar os valores de deformação, a e b do fluxograma da figura 5.20, é crucial para obter valores dos coeficientes confiáveis. Na seção 5.2.6 foi mostrado um estudo paramétrico sobre a influência da separação dos valores das deformações na variação dos coeficientes.

- Sobre a determinação do elemento de volume representativo, utilizando como critério a avaliação dos índices I_0 , I_1 e I_2 , estes mostraram que, para 8 nódulos, já se teria um EVR adequado para caracterizar o material estudado, até o nível de deformação aplicado. No entanto, devido ao fato de que o estudo não foi realizado com valores mais adequados para os separadores a e b , de forma a evitar perturbações ocasionadas na matriz de

deformação, com a qual se calculam os coeficientes alfas, os estudos subsequentes foram realizados com um EVR de 32 heterogeneidades.

- Após a determinação do elemento de volume representativo, dois testes foram realizados: o de tração uniaxial e o de corte puro no EVR com estiramento máximo de $\lambda_f^3 = 1,19$, objetivando verificar a eficácia do método de homogeneização empregado. Foi efetivado o traçado da curva tensão x deformação para os dois casos e realizada a comparação com os valores de tensão obtidos a partir dos valores de α_0 , α_1 e α_2 mapeados, para os níveis de deformação ocorridos em ambos os ensaios. Os resultados obtidos foram satisfatórios, como se observa nas figuras 5.57 e 5.63.

Por fim, cabe salientar que o método empregado é eficiente para determinação do EVR nos níveis de deformação empregados.

8.2 Conclusões da etapa de estudos com AG

Partindo-se de trabalho anterior, procurou-se determinar as propriedades elásticas da matriz cúbica e/ou das heterogeneidades esféricas de um modelo com distribuição regular de 8 nódulos esféricos utilizando a técnica dos Algoritmos Genéticos.

Deste estudo, apresentado no capítulo 6, foi possível se chegar às seguintes conclusões:

a) No Caso A, em que foram fixadas as propriedades da matriz, determinou-se as propriedades das heterogeneidades que gerassem o K^* e o G^* do composto. Como estas constantes eram características de um modelo já, o resultado aproximou-se muito bem do esperado;

b) No Caso B, foi pesquisado qual a combinação de materiais que proporciona um módulo de elasticidade pré-estabelecido para o composto. Com 20 indivíduos encontrou-se uma aproximação muito boa, mas com elevado número de combinações realizadas em relação às possíveis (50,8%). Diminuindo a quantidade de indivíduos para 10 e 6, e deixando constantes as outras variáveis de estudo, obteve-se bons resultados com menor quantidade de combinações realizadas (29,3% e 11,3% respectivamente). No caso de 6 indivíduos, aumentando a probabilidade de mutação obteve-se o mesmo resultado que para 20 indivíduos, reduzindo à metade as combinações estudadas (de 50,8% para 25,4%).

c) No Caso C foi realizada uma análise singela a respeito das propriedades das heterogeneidades que fazem com que as variações de tensões de Von Mises nas interfaces

sejam minimizadas. O resultado obtido foi o esperado, mostrando que o algoritmo está funcionando corretamente e pronto para ser utilizado para aplicações futuras. Esta implementação poderá tomar maior relevância quando aplicada, por exemplo, concomitantemente à mudança de forma e orientação das heterogeneidades, ou então quando aplicada uma interface coesiva entre a matriz e as heterogeneidades.

As conclusões tiradas desta etapa foram satisfatórias. O algoritmo pode ser usado com pequenas adaptações para problemas mais complexos, acrescentando outros fatores à função objetivo, como por exemplo: peso, custo, leveza, ductilidade, resistência a altas temperaturas, condutividade elétrica e tenacidade de cada um dos materiais; forma, orientação, distribuição aleatória e densidade volumétrica das heterogeneidades.

8.3 Conclusões da etapa de estudos com MED

No capítulo 7, foi apresentada uma aplicação do método dos elementos discretos na determinação do elemento de volume representativo de um material composto particulado com distribuição aleatória. O estudo foi realizado sobre uma matriz polimérica de PMMA com heterogeneidades esféricas similares a uma borracha, que por sua vez possuíam uma heterogeneidade também esférica de PMMA. Os resultados do estudo, considerando os parâmetros de análise já descritos, apontam para as seguintes conclusões:

a) De modo geral, para regime linear elástico são obtidos os menores valores de dispersão na resposta mecânica dos modelos, tanto nos valores médios para os diferentes volumes de controle verificados, quanto dentro das réplicas, como se pode verificar na figura 7.3 e na tabela 7.2. Pode afirmar-se que, sob estas condições, um volume de 20 nódulos, considerando um erro admissível de 1% entre o valor de análise da amostra em relação ao valor da maior amostra, poderia ser considerado como elemento de volume representativo da micro-estrutura considerada. Utilizar volumes maiores não significaria numa sensível melhora na resposta mecânica dos modelos e acarretaria um maior esforço computacional.

b) Em condições de não linearidade, a estabilidade na resposta mecânica das variáveis verificadas não foi alcançada para os volumes analisados. Os valores de dispersão foram maiores que no caso linear e, embora ainda pequenos (Tabela 7.2), o comportamento mostrou-se instável no intervalo estudado (Figura 7.4), como se pode reparar pela dispersão dos valores médios das variáveis de campo para os diferentes volumes de controle analisados. Portanto, com base nas observações realizadas, pode se afirmar que seria necessário um estudo com mais heterogeneidades para que se possa, no regime não linear, determinar o

volume de elemento representativo da micro-estrutura, que pudesse verificar desta forma sua resposta mecânica em condições de não linearidade e próximos da ruptura do material.

c) Na região de tensão máxima a estabilidade na resposta mecânica das variáveis verificadas foi alcançada para volumes maiores que para regime linear. No comportamento estável (entre 20 e 50 heterogeneidades) verificou-se um aumento da dispersão na resposta mecânica dos modelos em relação ao regime linear elástico, tanto entre os valores médios das variáveis de campo consideradas para os diferentes volumes de controle analisados, quanto dentro das réplicas (Figura 7.5 e Tabela 7.2). Foi possível observar que, considerando um erro admissível entre 3% e 5% entre o valor de análise da amostra em relação ao valor da maior amostra, volumes entre 20 e 30 heterogeneidades poderiam ser considerados como representativos da micro-estrutura. Contudo, cabe ressaltar aqui também a necessidade de um estudo com mais heterogeneidades, uma vez que tendência de estabilidade da resposta mecânica do material foi atingida muito próxima ao tamanho máximo analisado.

Finalmente, cabe salientar que, a partir dos resultados obtidos, o método é eficiente para o estudo em questão. No entanto, para estender este estudo do EVR na faixa não-linear do comportamento, e considerar outros fatores, como por exemplo, o descolamento entre matriz e nódulos esféricos e também leis não-lineares de comportamento para os nódulos e para a interface nódulo-matriz, são necessários aprimoramentos, entre eles a análise com amostras de volumes de controle maiores.

8.4 Sugestões para trabalhos futuros

Dos temas desenvolvidos nesta tese, foi possível perceber algumas áreas onde seria de maior interesse continuar investindo. Entre elas se destacam os estudos desenvolvidos sobre estruturas não-lineares, onde se poderia continuar a investigação:

- Aplicando a metodologia proposta por Temizer e Zohdi, 2007 a outros materiais compostos não-lineares, verificando se nestes outros casos é possível determinar desta maneira o EVR;

- Estudando de forma mais detalhada e conclusiva a influência da defasagem dos valores da matriz de deformações na determinação dos coeficientes alfas;

- Propondo e estudando a aplicabilidade de outras leis para se determinar o valor da tensão, diferente da equação (3.4), na qual possa aparecer outro termo ou que se utilize, em vez de um polinômio, uma função transcendente, como logaritmo, por exemplo;

- Estudando como a perda de isotropia, à medida que as deformações aumentam, pode invalidar a metodologia aqui proposta;

- Incluindo outros tipos de não-linearidades no material estudado, sendo a mais comum o descolamento da interface entre nódulos e matriz. Isto poderia ser estudado com elementos finitos, incluindo contato entre as fases, e também com elementos discretos.

Quanto à abordagem linear elástica, pode-se aprofundar a análise verificando os fatores que influenciam na diferença da velocidade de convergência dos parâmetros elásticos K e G .

REFERÊNCIAS BIBLIOGRÁFICAS

Abaqus/Simulia, 2009. “http://www.simulia.com/products/abaqus_fea.html”.

ABIPLAST (Associação Brasileira da Indústria do Plástico), 2009. “<http://www.abiplast.org.br>”.

Absi, E., 1971. “Théorie des equivalences-determination de quelques éléments types”, **Anales de L’institut Technique Du Balument el des Travau Publics**, Nº281, pp. 83-86.

Aidun, J.B., Lo, D.C.S., Trucano, T.G., Fye, R.M., 1999. “**Representative Volume Size: A Comparison of Statistical Continuum Mechanics and Statistical Physics**”, Sandia National Laboratories Albuquerque, New Mexico.

Aliabadi, M.H., Rooke, D.P., 1991. “**Numerical Fracture Mechanics**”, Computational Mechanics Publications, Kluwer Academics Publishers, Great Britain.

Anderson, T.L., 1995. “**Fracture Mechanics**”, CRC Press, Second Edition. Department of Mechanical Engineering, Texas A&M University, College Station, Texas, USA.

Ansys, 2009. “<http://www.ansys.com>”.

Argon, A.S., Cohen, R. E., 2003. “Toughenability of polymer”, **Polymer**, 44:6013-6032.

Armero, F., Oller, S., 2000. “A General Framework, for Continuum Damage Models. II. Integration Algorithms, With Applications to the Numerical Simulation of Porous Metals”, **International Journal of Solid and Structures**, vol. 37, pp. 7437-7464.

Barrios, D'Ambra Ricardo; Iturrioz, Ignacio; Fasce, Laura A.; Frontini, Patricia M.; Cisilino, Adrian P., 2002. “Utilización del método de los elementos discretos en la simulación numérica de ensayos de impacto para caracterización de materiales compuestos poliméricos”, MECOM 2002- Santa Fé- Paraná, Argentina, **Mecánica Computacional**, Vol. XXI, ISSN 1666-6070, pp. 1121-1134.

Bathe, K.J., 1996. “**Finite Element Procedures**”, Prentice-Hall, Englewood Cliffs, New Jersey.

Batista, R.G., 2007. “**Aplicação do método dos elementos discretos ao estudo de micromecânica do dano de materiais microporosos de matriz metálica**”, Tese de Doutorado, PROMEC, Universidade Federal do Rio Grande do Sul, pp. 157.

Batista, R.G., Iturrioz, I., Cisilino, A., 2003. “Aplicación del método de los elementos discretos a la homogenización de las propiedades mecánicas de materiales microporosos”, **XIII Congreso sobre Métodos Numéricos y sus Aplicaciones** (ENIEF), Bahía Blanca, Argentina.

Batista, R.G., Iturrioz, I., Cisilino, A., 2004. “Caracterização micromecânica do dano em materiais microporosos utilizando o método dos elementos discretos”, **2o Congresso de Ciência e Engenharia de Materiais do Mercosul** (SULMAT), Joinville, SC, Brasil.

Bermont, O.M.V., 2000. “**Fractura em Fundiciones de Hierro**”, Tese de doutorado, Universidad Nacional de Mar Del Plata, Facultad de Ingeniería, (INTEMA), Mar del plata, Argentina.

Bittencourt, E., Cisilino, A., 2005. “Study of the interface matrix-inclusion properties on fracture properties in composite materials”, **18th Congress of Mechanical Engineering**.

Biwa, S., Ito, N., Ohno, N., 2001. “Elastic properties of rubber particles in toughened PMMA: ultrasonic and micromechanical evaluation”, **Mechanics of Materials** 33:717-728.

Böhm, H.J., 1998. “**A short introduction to basic aspects of continuum micromechanics**”, CDL-FMD Report 3-1998, Vienna University of Technology.

Brara, A., Camborde, F., Klepaczko, J.R., Mariotti, C., 2001. “Experimental and Numerical Study of Concrete at High Strain Rates in Tension”, **Mechanics of Material**, vol. 33, pp. 33-45.

Buroni, F.C., 2006. “**Modelagem e projeto computacional de materiais microporosos com distribuição aleatória utilizando uma formulação de elementos de contorno**”, Dissertação de Mestrado, PROMEC, Universidade Federal do Rio Grande do Sul, pp. 112.

CAIP (Cámara Argentina de la Industria del Plástico), 2009. “<http://www.caip.org.ar>”.

Camborde, F., Mariotti, C., Donzé, F.V., 2000. “Numerical Study of Rock and Concrete Behavior by Discrete Element Modeling”, **Computer Geotechnics**, vol. 27 pp. 225-247.

Castañeda, P.P., 1989. “The overall constitutive behavior of nonlinearly elastic composites”, **ProcRSoc Lond** A422(1862):147–171.

Chou, C.S., Tseng, C.Y., Smid, J., Kuo, J.T., Hsiau, S.S., 2000. “Numerical Simulation of Flow Patterns of Disks in the Asymmetric Louvered-Wall Moving Granular Filter Bed”, **Powder Technology**, vol.110, pp. 239-245.

Cook, R. D., Malkus, D. S., Plesha, M. E., 1974. “**Concepts and Applications of Finite Element Analysis**”, University of Wisconsin – Madison, John Wiley & Sons, Nova Iorque.

Cundall, P.A., 1971. “A computer model for simulating progressive large scale movements in block rock systems”, **Symposium International Society of Rock Mechanics**, Nancy, France.

Cundall, P.A., Strack, O.D.L., 1979. “A distinct element model for granular assemblies”, **Geotechnique**, vol.29:47, pp. 65.

Dalguer, A., Irikura, K., Riera, J.D., 2003. “Generations of New Cracks Accompanied by Dynamic Shear Rupture Propagation of the 2000 Tottori (Japan), Earthquake”, **Bulletin of the Seismological Society of America**, vol. 93, pp. 2236-2252.

Drugan, W.J., Willis, J.R., 1996. “A Micromechanics-Based Nonlocal Constitutive Equation and estimates of Representative Volume Element Size for Elastic Composites”. **J.Mech.Phys.Sol**, 44:497-524.

E399-90, 1997. “**Standard Test Method for Plane-Strain Fracture Toughness of Metallic Materials**”, American Society for Testing and Materials. ASTM Standards.

Eshelby, J.D., 1957. “The determination of the elastic field of an ellipsoidal inclusion, and related problems”. **Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences**, 241: 376-396, N°. 1226.

Fasce, L.A., Frontini, P.M., Wong, S-C., Mai Y-W., 2004. “Polypropylene Modified with Elastomeric Metallocene-Catalyzed Polyolefin Blends: Fracture Behavior and Development of Damage Mechanisms”, **Journal of Polymer Science: Part B: Polymer Physics**, Vol. 42, 1075–1089.

Fasce, L.A., 2002. “**Comportamiento mecânico de polipropileno modificado con una poliolefina elastomérica**”, Tese de Doutorado, INTEMA, Universidad Nacional de Mar del Plata, pp. 132.

Forsythe, G.E., Malcolm, M.A., Moler, C.B., 1977. “**Computer Methods for Mathematical Computations**”, Englewood Cliffs, NJ: Prentice-Hall.

Galiano, B.R., Iturrioz, I., Cisilino, A., 2007. “Computacional modelling of the micromechanics of void growth and coalescence in ductile failure via the discrete element method”, **In: Solid Mechanics in Brazil-MECSOL**, São Paulo, ABCM.

Galiano, R.B., Cisilino, A., Iturrioz I., 2005. “Determinación del elemento de volumen representativo de compuestos particulados utilizando el método de los elementos discretos”, **Jornadas SAM/CONAMET**, memmat, Mar del Plata, Argentina.

Garg, A., Mai Y.W., 1998. “Failure mechanisms in toughened epoxy resins - A review”, **Composites Sci. Tech.**, 31, 179-193.

Gethin, D.T., Ransing, R.S., Lewis, R.W., Dutko, M., Crook, A.J.L., 2001, “Numerical Comparison of a Deformable Discrete Element Model and an Equivalent Continuum Analysis for the Compaction of Ductile Porous Material”, **Computers and Structures**, vol. 79, pp. 1287-1294.

Goldberg, D.E., 1989. “**Genetic Algorithms in Search, Optimization and Machine Learning**”, Addison-Wesley Publishing Company, Inc.

Goth, H., Sakai, T., 1997. “Numerical Simulation of Sheet flow as Granular Material”, **Journal of Waterway Port Coast and Ocean Engineering**, vol. 123, pp. 329-336.

Griffith, A.A., 1920. “The Phenomena of Rupture and Flow of Solids”, **Philosophical Transaction Royal Society**, A 221, pp. 163-198.

Gross, D., Seelig, T., 2006. “**Fracture Mechanics: With an Introduction to Micromechanics**”, Springer.

Hayashi, N., 1982. “**Sobre um Modelo de discretização de estruturas tridimensionais aplicado a dinâmica não linear**”, Dissertação de Mestrado, CPGEC, Universidade Federal de Rio Grande do Sul, Porto Alegre.

Hibbit, Karlsson & Sorensen INC., 1995. “**Introduction to the Abaqus / Explicit**”, U.S.A.

Hillerborg, A., 1978. “A Model for fracture analysis”, **Cód LUTVDG/TV**, BM-3005/1-8.

Holland, J. H., 1975. “**Adaptation in Natural and Artificial Systems**”, University of Michigan Press, Ann Arbor, Michigan.

Hollister, S., Kikuchi, N., 1997. “**Computational Methods in Biomechanics**”, course notes, Michigan University.

Hooley, C.J., Moore, D.R., Whale, M., Williams, M.J., 1981. “**Plastics and Rubber Processing and Applications**”, 1:345.

Hoss, L., 2009. “**Modelos Constitutivos Hiperelásticos para Elastômeros Incompressíveis: Ajuste, Comparação de Desempenho e Proposta de um Novo Modelo**”, Dissertação de Mestrado, PROMEC-UFRGS, Porto Alegre, p.290.

Hrennikoff, A., 1942. “**Solution of Problems of Elasticity by the Framework Method**”, Annual Meeting of The American Society of Mechanical Engineers, New York, USA.

Iturrioz, I., Ortiz, J., Cisilino, A., 2001. “Numerical Modelling of Crack-Microcrack Interaction and Coalescence Using the Discrete Element Method”. **Congreso de Materiales Compuestos del INTEMA - COMAT**, Mar del Plata, Argentina. Anais.

Irwin, G.R., 1957. “Analysis of stresses and strains near the end of a crack traversing a plate”, **Journal of Applied Mechanics**, vol. 24, pp. 361-364.

Kamiński, M., Figiel, Ł., 2001. “Effective Elastoplastic Properties of the Periodic Composites”, **Computational Materials Science**, vol. 22, pp.221-239.

Kanit, T., Forest, S., Galliet, I., Mounoury, V., Jeulin, D., 2003. “Determination of the size of the representative volume element for random composites: statistical and numerical approach”, **International Journal of Solids and Structures**, vol. 40, pp. 3647–3679.

Kanninen, M.F., Popelar, C.H., 1985. “**Advanced Fracture Mechanics**”, Oxford University Press, New York

Kanouté, P., Bosco, D.P., Gaboche, J.L., Schrefler, B.A., 2009. “Multiscale Methods for Composites: A Review”, **Arch Comput Methods Eng**, 16:31-75.

Kari, S., Berger, H., Rodriguez-Ramos, R., Gabbert, U., 2007. “Computational evaluation of effective material properties of composites reinforced by randomly distributed spherical particles”, **Composite Structures** 77:223–231.

Kim, J., Gao, X., Srivatsan, T.S., 2004. “Modeling of void growth in ductile solids: effects of stress triaxiality and initial porosity”, **Engineering Fracture Mechanics**, vol. 71, pp.379–400.

Knott, J.F., 1993. “**The Science and Engineering of Fracture**”, Advanced in Fractures Resistance and Structural Integrity, selected papers from the 8th International Conference on Fracture (ICF8), Kyiv, Ukraine.

Kouznetsova, V., Brekelmans, W.A.M., Baaijens, F.P.T., 2001. “An Approach to Micro-macro Modeling of Heterogeneous Materials”, **Computational Mechanics**, vol. 27, pp. 37-48.

Kouznetsova, Varvara G., 2002, “**Computational homogenization for the multi-scale analysis of multi-phase materials**”, Technische Universiteit Eindhoven, Eindhoven.

Ladislav, E.E., Dutko, M., 2003. “**Large Scale Fracturing and Rock Flow Using Discrete Element Method, 2D Application for Block Caving**”, Numerical Methods in Continuum Mechanics, Žilina, Slovak Republic.

Laurienzo, P., Malinconico, M., Martuscelli, E., Ragosta, G., Volpe, M.G., 1996. “**Advanced Routes for Polymer Toughening**”, 10:439±68.

Lin, C-J., Wei, W-C.J., Iwai, T., Hong, C-W., Greil, P., 2000. “Discrete Element Method (DEM) Simulation and Processing of Mo/Al₂O₃ Granules in a Fluidizing Bed”, **Proceeding of the National Science Council of the Republic of China**, vol. 24, pp. 394-404.

LS-Dyna, 2009. “<http://www.ls-dyna.com>”.

Malvern, L., 1968. “**Introduction to the mechanics of a continuous medium**”, Prentice Hall.

Mendonça, P.T.R., 2005. “**Materiais Compostos & Estruturas – Sanduíche**”, Manole.

Milton, G.W., 2002. “**The Theory of Composites**”, Cambridge University Press, United Kingdom.

Montgomery, D.C., 1991. “**Design and Analysis of Experiments**”, John Wiley and Sons, New York, 3rd ed.

Mori, T., Tanaka, K., 1973. “Average stress in matrix and average energy of materials with misfitting inclusions”, **Acta. Metall**, 21:571-574.

Moura Branco, C., Fernandes, A.A., Tavares de Castro, P.M.S., 1986. “**Fadiga dos Metais**”, Fundação Calouste Gulbenkian, Lisboa.

Müller, S., 1987. “Homogenization of nonconvex integral functionals and cellular elastic materials”, **Arch Ration Mech Anal**, 25:189–212.

Munjiza, A., Bangash, T., John, N.W.M., 2004. “The Combined Finite-Discrete Element Method for Structural Failure and Collapse”, **Engineering Fracture Mechanics**, vol. 71, pp. 469-483.

NDSM, 2009. “<http://www.ndsm.ufrgs.br>”.

Nemat-Nasser, S., Hori, M., 1999. “**Micromechanics: overall properties of heterogeneous materials**”. Second Edition. North-Holland.

Ogden, R., 1984. “**Non-linear Elastic Deformations**”, Dver Publications.

Ogden, R.W., 1978. “Extremum principles in non-linear elasticity and their application to composites”, **Int J Solids Struct**, 14:265–282.

Okamoto, Y., Miyagi, H., Mitsui., 1993. “New cavitation mechanism of rubber dispersed polystyrene”, **Macromolecules**, 26(24), 6547-6551.

Ortiz, J.E., Cisilino, A.P., Otegui J.L., 2001. “Effect of Microcracking on the Micromechanics of Fatigue Crack Growth in Austempered Ductile Iron”, **Fatigue and Fracture of Engineering Materials and Structures**, 24/9:591-606.

Ostoja-Starzewski, M., 2006. “Material Spatial Randomness: From Statistical to Representative Volume Element”, **Probabilistic Engineering Mechanics**, vol 21, pp.112-132.

Lovell, P.A., Ryan, A.J., Sherrett, M.N., Young, R.J., 1994. “**Proceedings of Deformation, Yield and Fracture of Polymers**”, Cambridge.

Heim, P., Wrotecki, C., Gaillard, P., 1993. “High impact cast sheets of poly(methyl methacrylate) with low levels of polyurethane”, **Polymer**, 1653-1660.

Pettarin, V., 2002. “**Análisis de la Información de Ensayos de Impacto Instrumentados Sobre Materiales Poliméricos**”, Tese de Doutorado, INTEMA, Universidad Nacional de Mar del Plata, pp. 119.

Procházka, P.P., 2004. “Application of Discrete Element Methods to Fracture Mechanics of Rock Bursts”, **Engineering Fracture Mechanics**, vol. 71, pp. 601-618.

Ren, Z.-Y., Zheng, Q.-S., 2004. “Effects of grain sizes, shapes, and distribution on minimum sizes of representative volume elements of cubic polycrystals”, **Mechanics of Materials**, 36:1217–1229.

Reuss, A., 1929. “Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle. Z. angew”, **Math. Mech**, 9:49-58.

Rice, J.R., 1968. “A Path Independent Integral and the Approximate Analysis of Strain Concentrations, by Notches and Cracks”, **Journal of Applied Mechanics**, vol. 35, pp. 379-386.

Riera, J.D., Iturrioz, I., 1998. “**Discrete Element Model for Evaluating Impact and Impulsive Response of Reinforced Concrete Plates and Shells Subjected to Impulsive Loading**”, Nuclear Engineering and Design, Elsevier, Amsterdam, The Netherlands, pp. 179, 135-144.

Riera, J.D., 1980. “A Critical Reappraisal of Nuclear Power Plant Safety Against Accidental Aircraft Impact”, **Nuclear Engineering and Design**, vol. 57, pp. 193-206, North Holland Publishing Company.

Riera, J.D., 1984. “Local Effects in Impact Problems on Concrete Structures”, **Proceedings, Conference on Structural Analysis and Design of Nuclear Power Plants**, Porto Alegre, Brasil, CPGEC/UFRGS, Porto Alegre V. III, pp.57-79.

Rios, R.D., Riera, J.D., 2004. “**Size effects in reinforced concrete structures engineering structures**”.

Rocha, M.M., 1989. “**Ruptura e Efeitos de Escala em Materiais não Homogêneos**”, Dissertação de Mestrado, CPGEC, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Roderic, L., 1995. “**Experimental methods for study of cosserat elastic solids and other generalized elastic continua**”, in Continuum models for materials with microstructure, ed. H. Mühlhaus, J. Wiley, N. Y. Ch. 1, p. 1-22.

Sanchez-Palencia, Sanchez Hubert, 1992. “**Introduction aux méthodes asymptotiques et à l’homogénéisation**”, 280 pages, ed Masson. Paris.

Sawamoto, Y., Tsubota, H., Kasai, Y., Koshika, N., Morikawa, H., 1998. “Analytical Studies on Local Damage to Reinforced Concrete Structures Under Impact Loading by Discrete Element Method”, **Nuclear Engineering and Design**, vol. 179, pp. 157-177.

Schafer, B.C., Quingley, S.F., Chan, A.H.C., 2002. “Implementation of the Discrete Element Method Using Reconfigurable Computing (FGPAS)”, **15th ASCE Engineering Mechanics Conference**, Columbia University, New York, USA.

Sigmund, O. “Topology Optimization”, 2° ed., Springer-Verlag, Berlin.

Soares, G.P., 2003. “**Cálculo dos fatores de intensidade de tensões de uma trinca elíptica-hiperbólica homofocal através do método de transformação conforme**”, Dissertação de Mestrado, DeTec – DeFEM, Universidade Regional do Noroeste do Estado do Rio Grande do Sul – UNIJUI, pp. 124.

Soares, G.P., Iturrioz, I., Cisilino, A., 2008. “Aplicação do Método dos Elementos Finitos na Homogeneização das Propriedades Mecânicas de Materiais Compostos”, **Mecânica Computacional**, vol XXVII, 697-710.

Sokolnikoff, I.S., 1956. “**Mathematical Theory of Elasticity**”, McGraw-Hill Book Company, Nova Iorque.

Spencer, A. J. M., 2004. “**Continuum mechanics**”, Mineola: Dover, 183 p. : il.

Stumpf, F.T., 2009. “**Avaliação de um modelo Hiperelástico Incompressível: Análise de Restrições, Implementação e Otimização de Parâmetros Constitutivos**”, Dissertação de Mestrado, UFRGS, Porto Alegre, p.89.

Suquet, P.M., 1985. “**Local and global aspects in the mathematical theory of plasticity, Plasticity today: modelling, methods and applications, In Sawczuk A. and Bianchi G. editors**”, Elsevier Applied Science Publishers, 279–310.

Talbot, D.R.S., Willis, J.R., 1985. “Variational principles for inhomogeneous non-linear media”, **IMA J Appl Math**, 35:39–54.

Tech, T.W., Batista, R.G., Iturrioz, I., Cisilino, A., 2004. “Aplicação do método dos elementos discretos em mecânica de fratura estática e dinâmica”, **XXV Iberian Latin American Congress on Computational Methods (CILAMCE)**, Recife, Pernambuco, Brasil.

Temizer, I., Zohdi, T.I., 2007. “A numerical method for homogenization in non-linear elasticity”, **Comput Mech**, 40:281–298.

Terada, K., Muneo, H., Kyoya, T., Kikuchi, N., 2000. “Simulation of the Multi-Scale Convergence in Computational Homogenization Approach”, **International Journal of Solid and Structure**, vol. 37, pp. 2285-2311.

Trias, D., Costa, J., Turon, A., Hurtado, J.E., 2006. “Determination of the critical size of a statistical representative volume element (SRVE) for carbon reinforced polymers”, **Acta Materialia**, 54:3471–3484.

Truesdell, C., 1966. “**The elements of continuum mechanics**”, Springer Verlag.

Van der Sluis, O., Schreurs, P.J.G., Meijer, H.E.H., 1999. "Effective Properties of a Viscoplastic Constitutive Model Obtained by Homogenization", **Mechanics of Materials**, vol 31, pp.743-759.

Voigt, W., 1889. "Über die Beziehung zwischen den beiden Elastizitätskonstanten isotroper Körper", **Wied. Ann**, 38:573-587.

Wells, A.A., 1961. "Application of Fracture Mechanics at and Beyond General Yielding", **British Welding Journal**, vol. 10, pp. 563-570.

Willis, J.R., 1994. "Upper and lower bounds for non-linear composite behavior", **Materials Sci Eng, A** 175:7-14.

Wittel, F.K., Schulte_Fischedick, J., Kun, F., Kroplin, B-H., Frie ,B.M., 2003. "Discrete Element Simulation of Transverse Cracking During the Pyrolysis of Carbon Fibre Reinforced Plastics to Carbon/Carbon Composites", **Computational Materials Science**, vol. 28, pp. 1-15.

Zohdi, T.I., 2003. "Genetic design of solids possessing a random-particulate microstructure", **The Royal Society. Phil. Trans. R. Soc. Lond.**, A 361:1021-1043.

Zohdi, T.I., 2002. "**Computational modeling and design of new random microheterogeneous materials**", CISM Course Notes.

Zohdi, T.I., 2002. "**Advanced School on Computational Micromechanics of Materials Science, Course Notes**", International Centre for Mechanical Sciences, Udine, Italia.

Zohdi, T.I., Wriggers, P., 2005. "**Introduction to Computational Micromechanic**", LNACM 20.

APÊNDICE A – Curvas do mapeamento do material

Neste apêndice são apresentados os mapas dos valores de α_i obtidos para diferentes estados de deformação. Observa-se que os valores de α_0 , α_1 e α_2 podem ser obtidos em função das deformações λ_E^1 , λ_E^2 e λ_E^3 nos tamanhos de amostras estudados, 2 a 64 heterogeneidades.

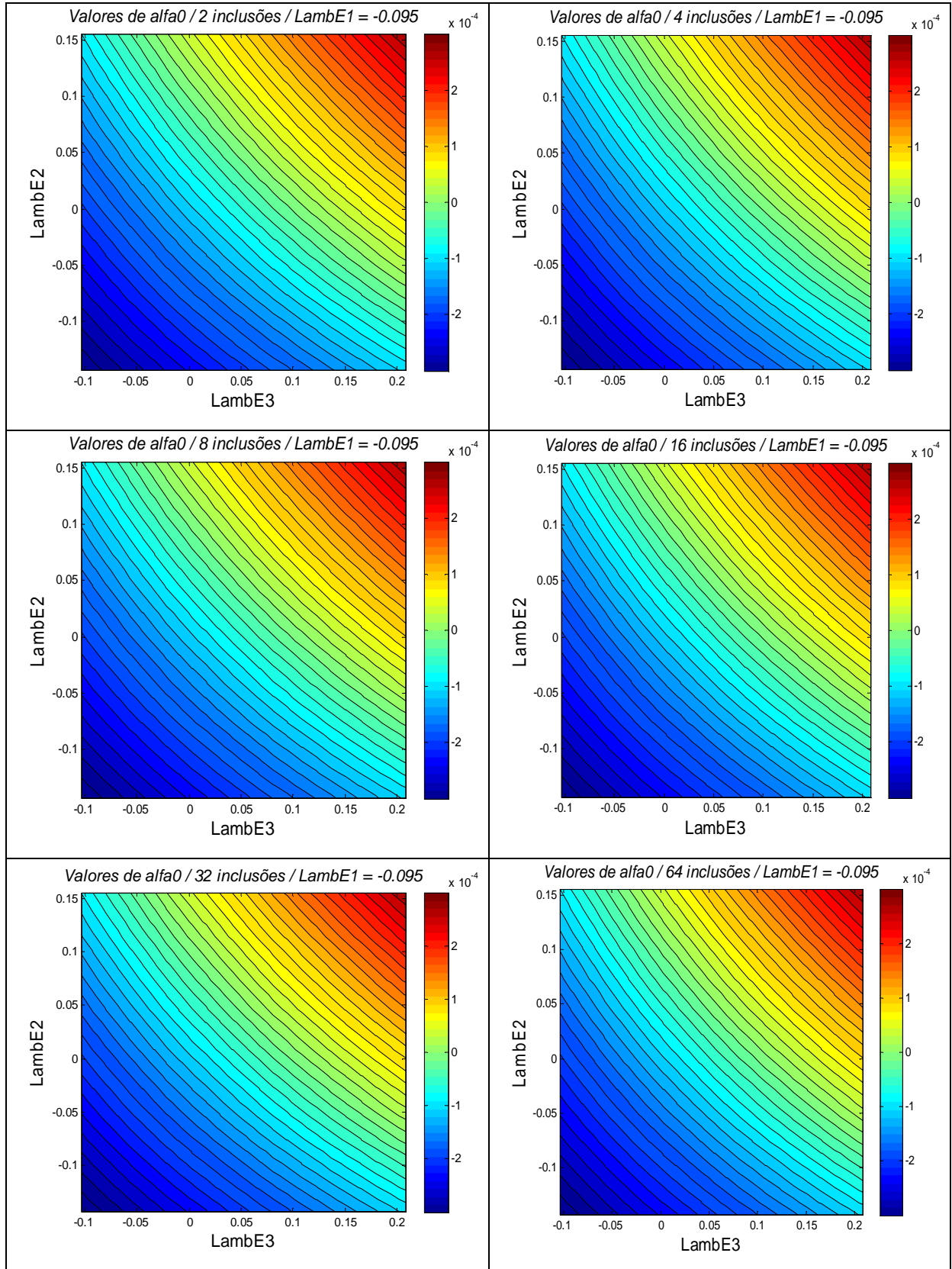


Figura A1 – Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados

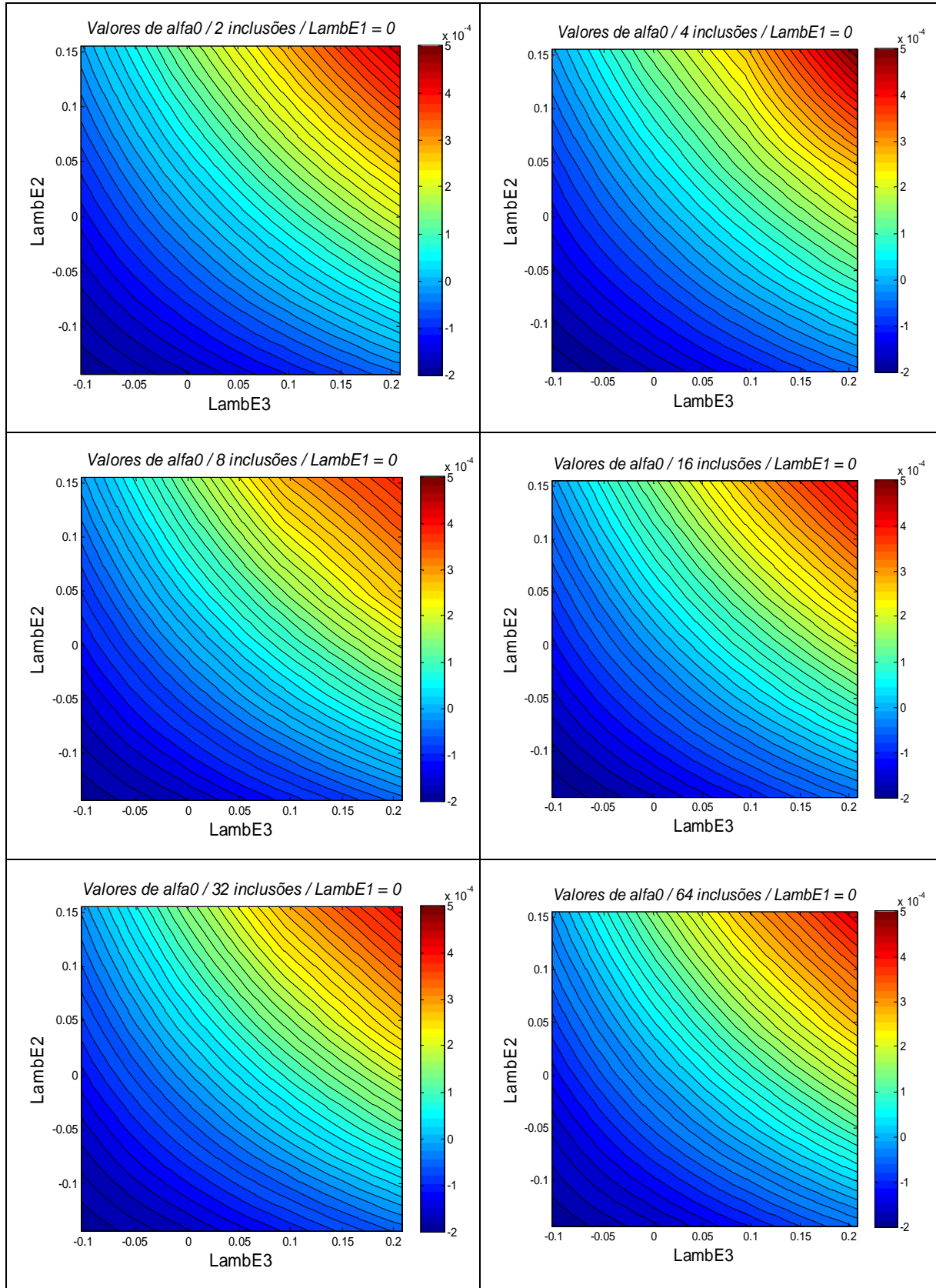


Figura A2 – Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados

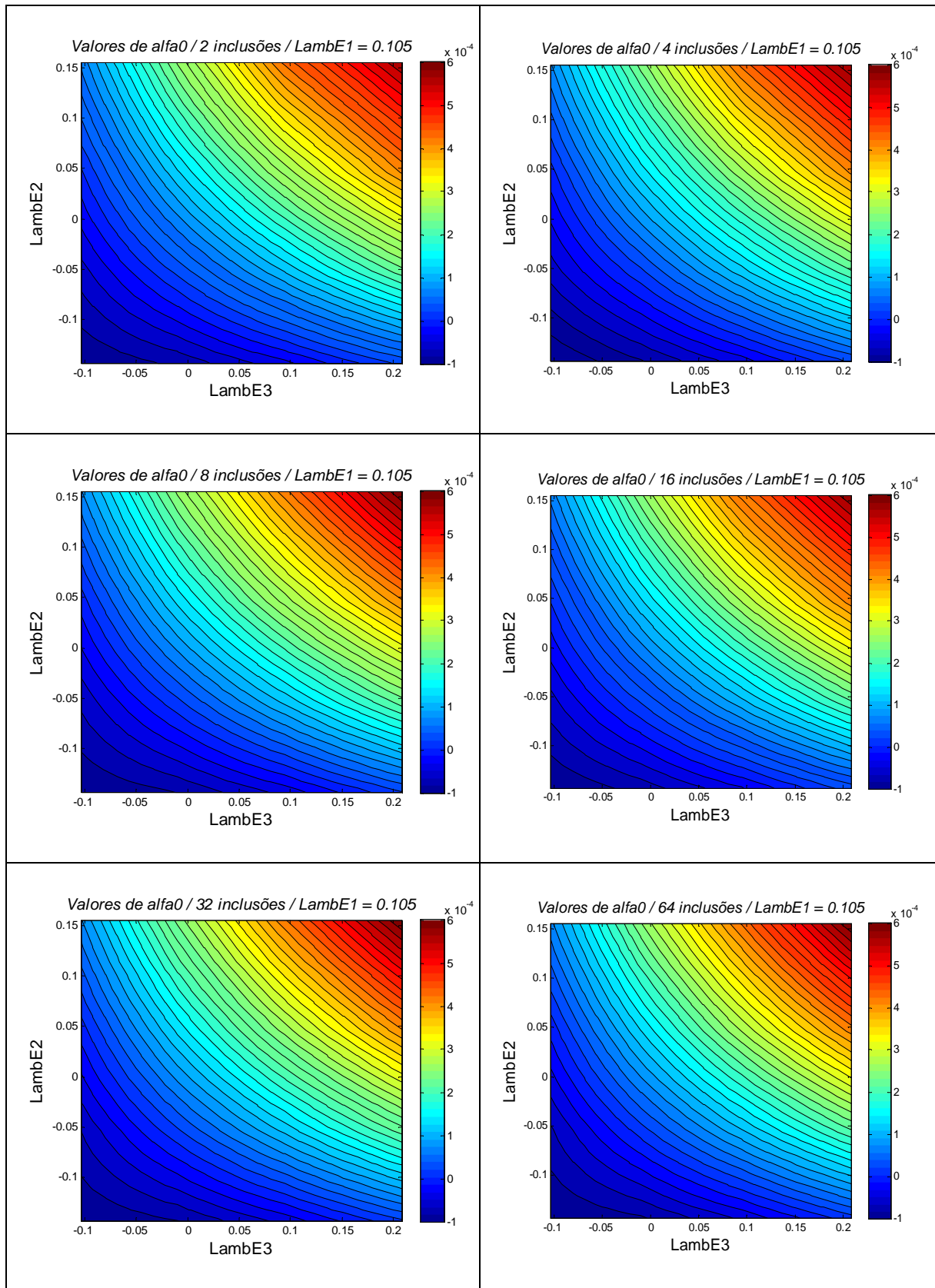


Figura A3 – Mapeamento de α_0 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados

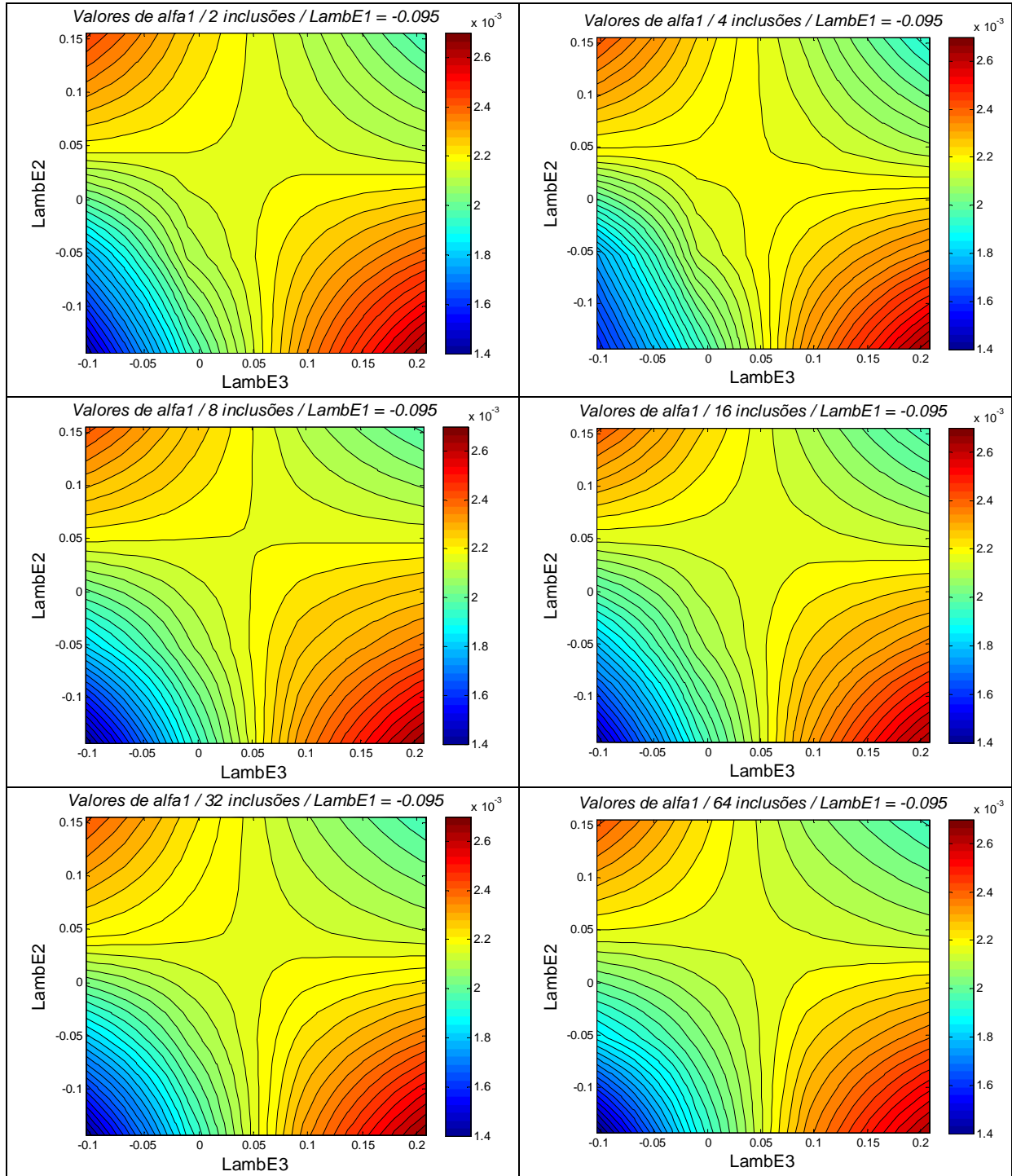


Figura A4 – Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados

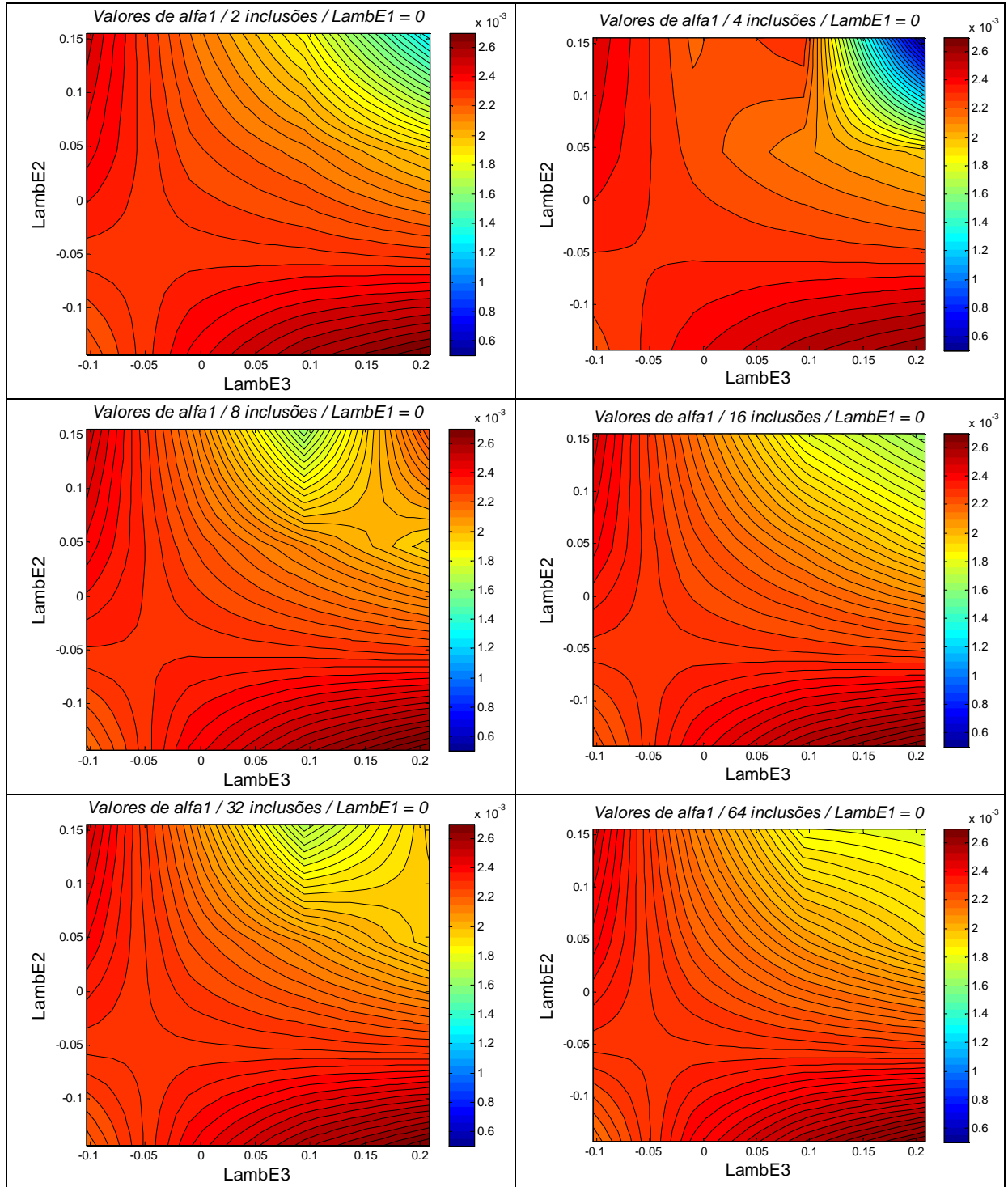


Figura A5 – Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados

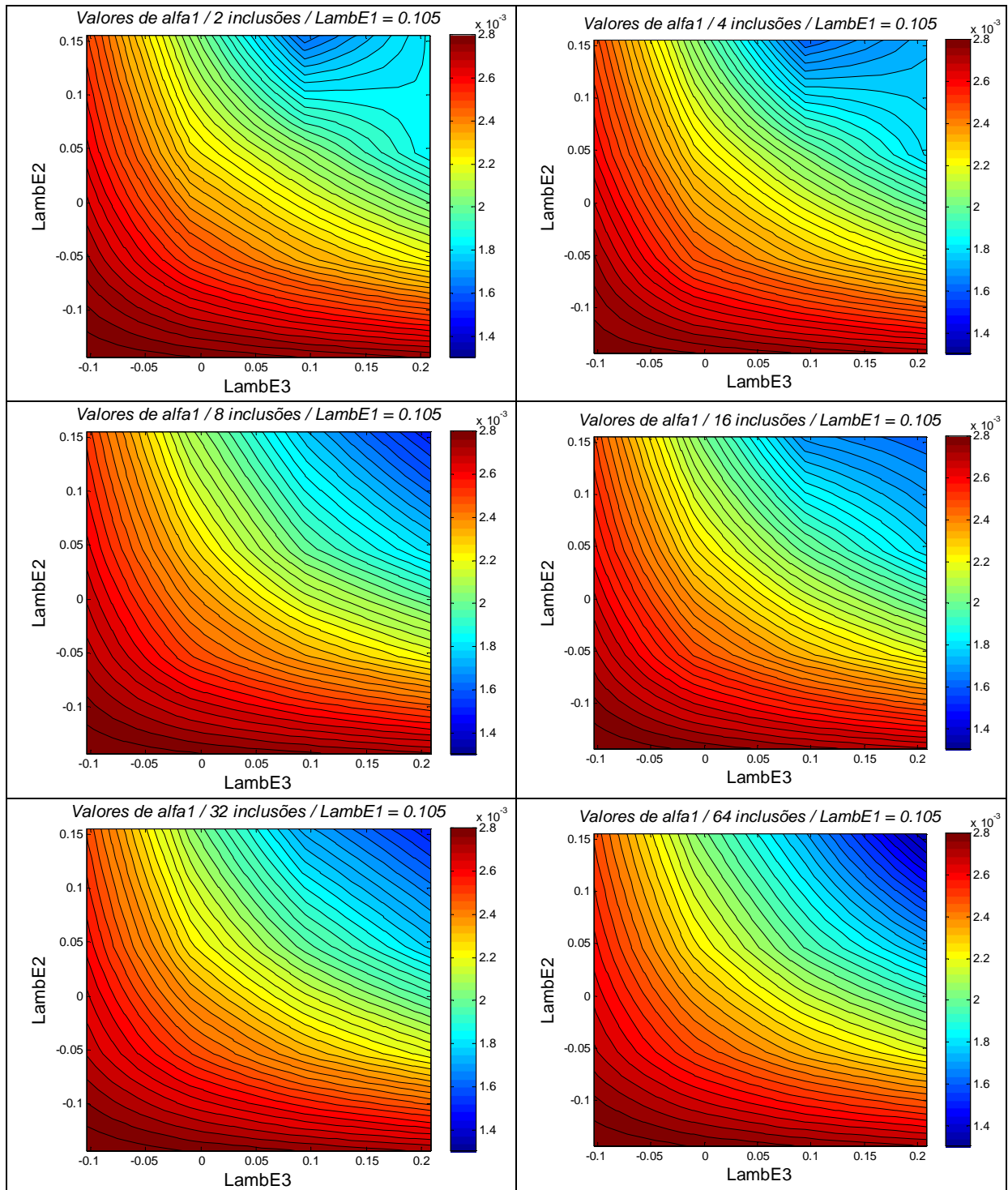


Figura A6 – Mapeamento de α_1 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados

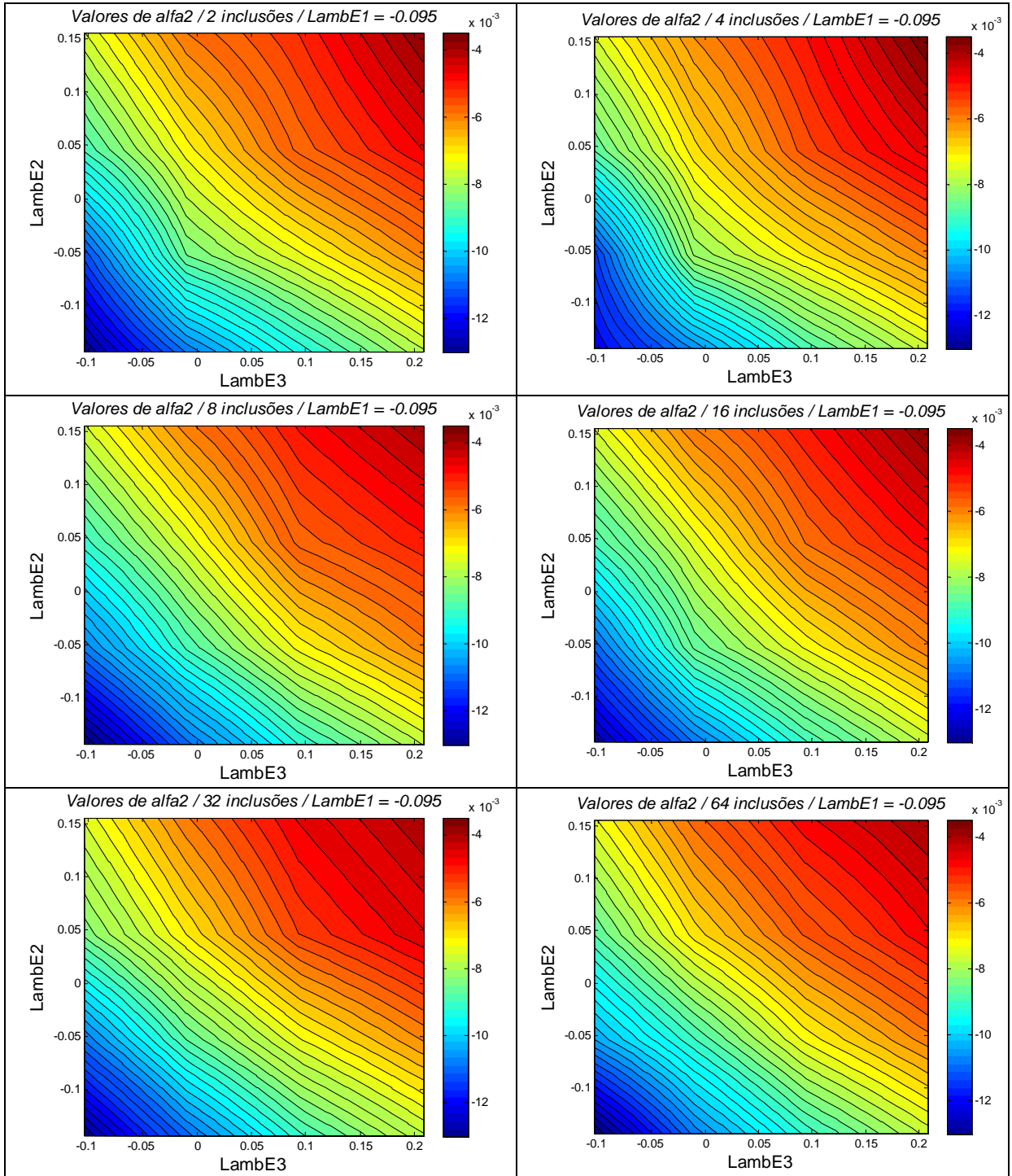


Figura A7 – Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = -0.095$ fixo, para os tamanhos de amostra analisados

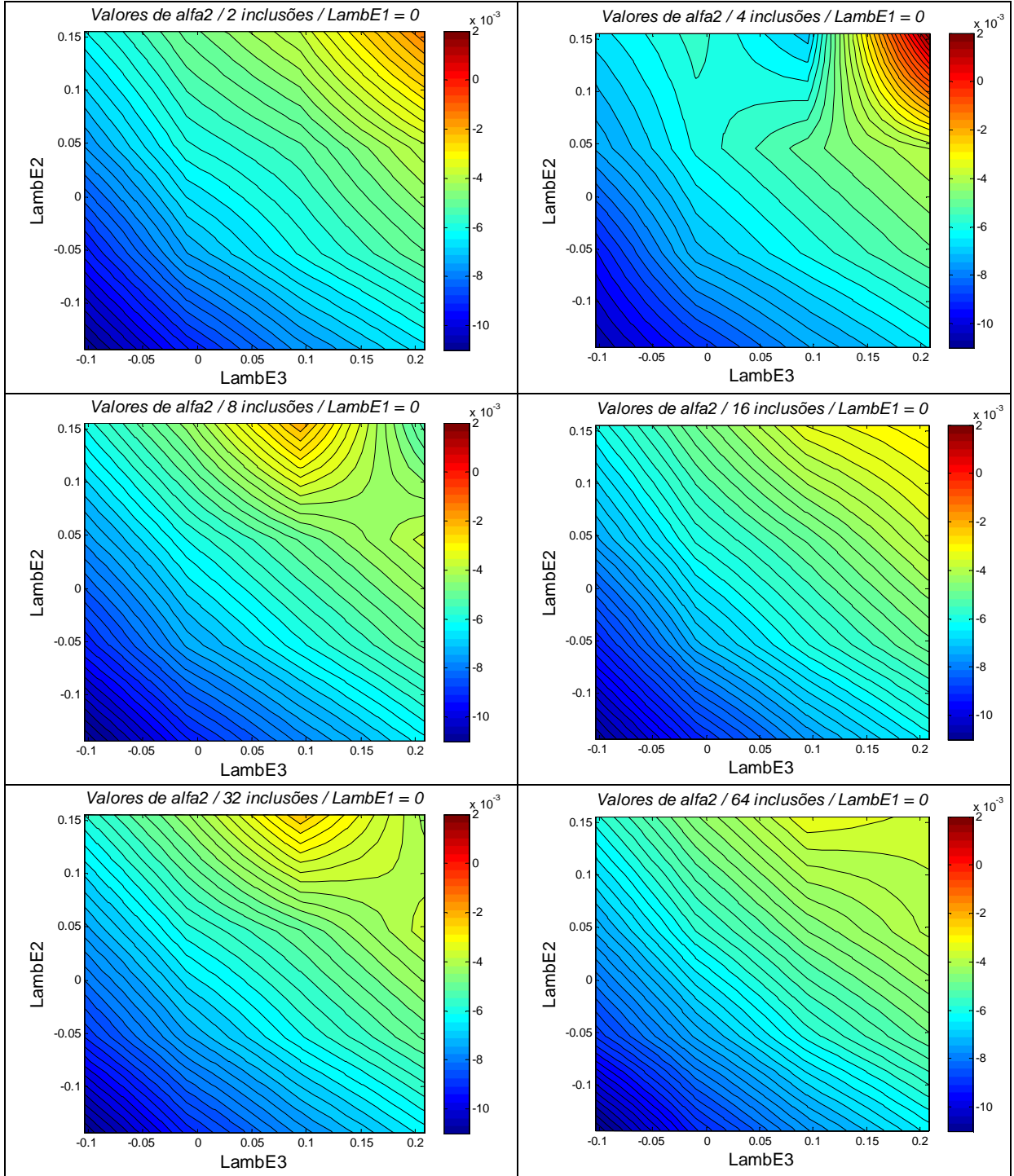


Figura A8 – Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = 0$ fixo, para os tamanhos de amostra analisados

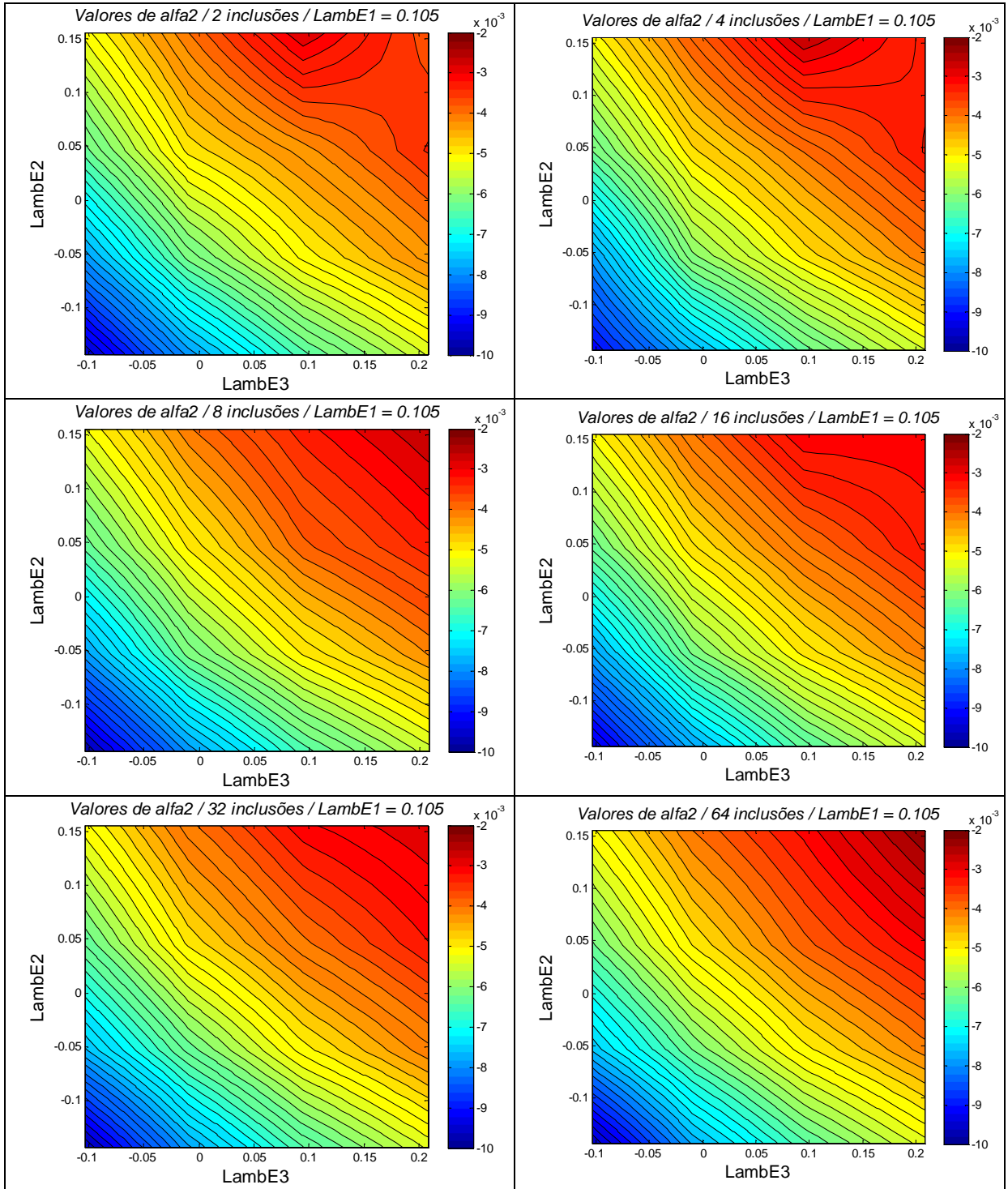


Figura A9 – Mapeamento de α_2 em função de λ_E^i com $\lambda_E^1 = 0.105$ fixo, para os tamanhos de amostra analisados

APÊNDICE B – Descrição do algoritmo utilizado para fazer o estudo de elementos finitos do caso linear elástico

Neste apêndice é apresentada a descrição detalhada dos programas desenvolvidos e utilizados na aplicação elástica linear em elementos finitos:

B.1 Descrição do algoritmo para a homogeneização elástico-linear

Nesta implementação foi utilizado o software de elementos finitos Abaqus/Standard através de análise elástica linear, para determinar as propriedades homogeneizadas do material estudado (matriz isotrópica cúbica com uma segunda fase formada por partículas esféricas com distribuição aleatória). Cabe salientar que, para isto, foi necessário automatizar o processo através da elaboração de um algoritmo, no caso, em Matlab. Este algoritmo é esquematizado através do diagrama de blocos da Figura B1 e descrito em detalhes posteriormente.

As variáveis utilizadas no diagrama da Figura B1 são identificadas a seguir:

ipot: parâmetro que vai de 1 até *npot* e permite calcular os diferentes números de nódulos de cada modelo;

npot: potência igual a 6 que permite identificar o modelo de 64 nódulos (tamanho máximo estudado);

inum_nod: parâmetro que identifica o número de nódulos do modelo que está sendo trabalhado;

iamostra: parâmetro que vai de 1 até *namostra* e permite identificar o número da amostra;

namostra: parâmetro que identifica o número máximo de amostras trabalhadas.

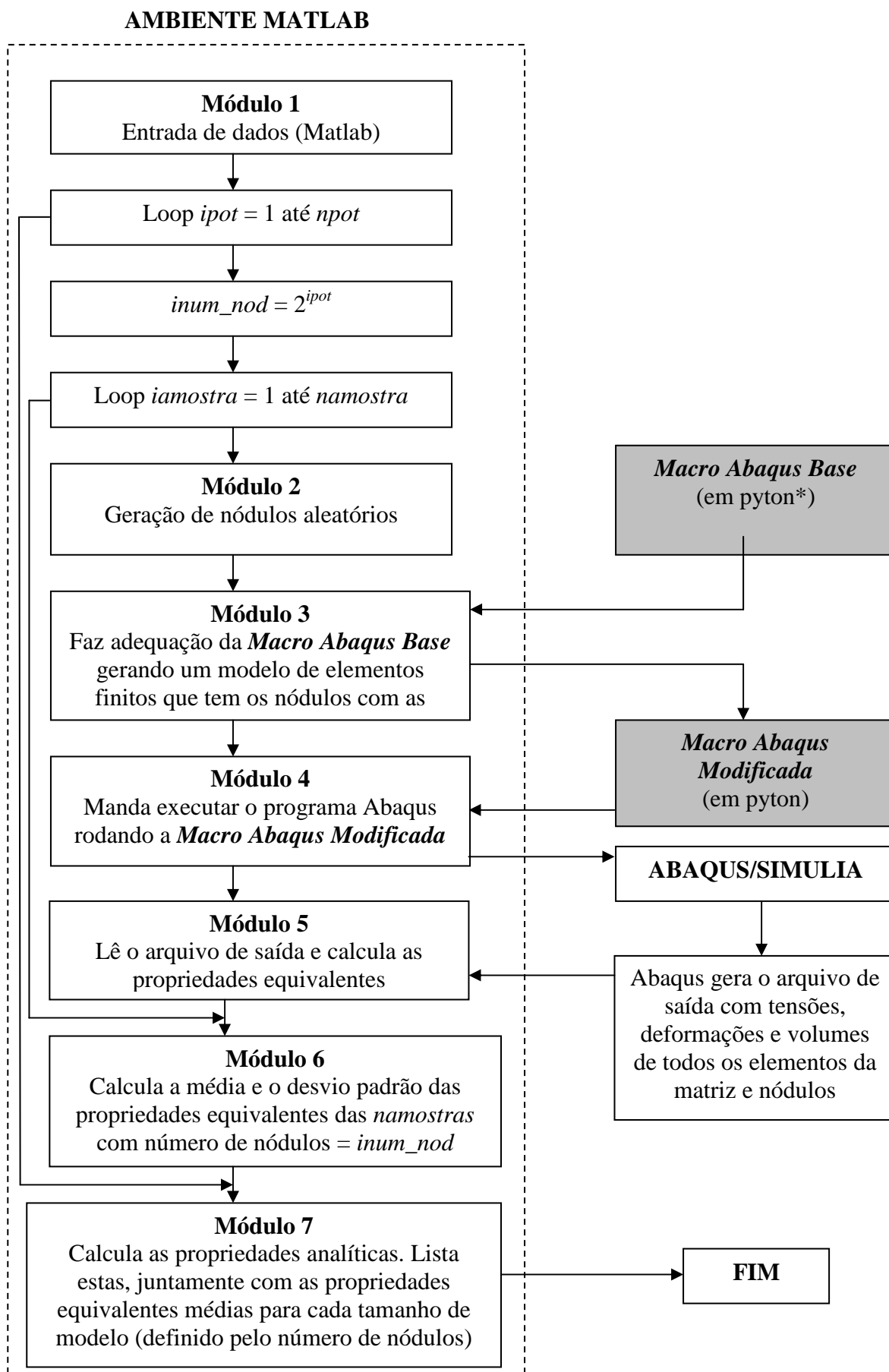


Figura B1 – Diagrama de blocos do algoritmo utilizado para calcular as propriedades efetivas equivalentes do composto elástico-linear (estudo descrito na seção 5.1)

* Python é a linguagem de comandos utilizada pelo software Abaqus/Simulia

B.2 Descrição detalhada do diagrama de blocos da Figura B1

A seguir se apresenta uma descrição detalhada dos módulos do diagrama de blocos da Figura B1, construídos dentro do ambiente Matlab. Na sequência, as seções B.2.1 a B.2.7 apresentam a listagem dos módulos programadas em Matlab. No, final, nas seções B.2.8 e B.2.9 são apresentadas as listagens da *Macro Abaqus Base* e da *Macro Abaqus Modificada* obtidas em arquivos de extensão *.dat*.

Módulo 1: Este módulo permite a introdução dos dados utilizados no algoritmo. São eles:

- Parâmetros geométricos para geração dos nódulos;
- Fração de volume (densidade) de heterogeneidades em relação ao volume total da amostra;
- Número máximo e número mínimo de nódulos na amostra;
- Número de amostras para cada tamanho de amostra;
- Módulo de elasticidade longitudinal da matriz e das heterogeneidades;
- Coeficiente de Poisson da matriz e das heterogeneidades;

A listagem deste módulo está na seção B.2.1.

Módulo 2: Este módulo gera os nódulos esféricos posicionados aleatoriamente no interior da matriz cúbica com distância mínima entre eles e entre eles e as bordas da matriz previamente estabelecidas para cada amostra. A listagem deste módulo está na seção B.2.2.

Módulo 3: Este módulo faz adequação na *Macro Abaqus Base* gerando um modelo de elementos finitos que tem os nódulos com as coordenadas geradas no módulo 2 (*Macro Abaqus Modificada*), através da sub-rotina chamada “*imprime_py*”. A listagem deste módulo está na seção B.2.3.

Módulo 4: Este módulo manda executar o programa Abaqus rodando a *Macro Abaqus Modificada* gerada no módulo 3. A listagem deste módulo está na seção B.2.4.

Módulo 5: Este módulo lê o arquivo de saída com tensões, deformações e volumes de todos os elementos da matriz e dos nódulos gerado no módulo 4 e calcula as propriedades equivalentes do composto, através da sub-rotina “*calcula_prop*”.

A seguir é descrito de forma detalhada os passos da sub-rotina “*calcula_prop*”:

- Lê o arquivo “*saída.rpt*” com as tensões, deformações e volumes de todos os elementos da matriz e das heterogeneidades;
- Calcula as propriedades equivalentes K^* e G^* , com média e desvio de cada configuração, através das expressões:

$$3K^* = \frac{\langle \frac{tr \sigma}{3} \rangle_{\Omega}}{\langle \frac{tr \varepsilon}{3} \rangle_{\Omega}} \quad e \quad 2G^* = \sqrt{\frac{\langle \sigma' \rangle_{\Omega} : \langle \sigma' \rangle_{\Omega}}{\langle \varepsilon' \rangle_{\Omega} : \langle \varepsilon' \rangle_{\Omega}}}$$

onde

$\frac{tr \sigma}{3}$ e $\frac{tr \varepsilon}{3}$ são as componentes dilatacionais do carregamento;

$\sigma' = \sigma - \frac{tr \sigma}{3} I$ e $\varepsilon' = \varepsilon - \frac{tr \varepsilon}{3} I$ são as componentes desviadoras do carregamento,

sendo I a matriz identidade;

$\langle . \rangle$ é o símbolo que indica o valor médio da magnitude indicada calculada componente a componente. Esta média é a ponderada em relação ao volume de cada elemento da matriz e dos nódulos.

A listagem deste módulo está na seção B.2.5.

Módulo 6: Este módulo calcula a média e o desvio padrão das propriedades equivalentes das *namostras* com número de nódulos = *inum_nod*. A listagem deste módulo está na seção B.2.6.

Módulo 7: Neste módulo são calculadas as propriedades analíticas, que são listadas juntamente com as propriedades equivalentes médias, obtidas para cada tamanho de modelo (definido pelo número de nódulos). A listagem deste módulo está na seção B.2.7.

B.2.1 Listagem do Módulo 1 programado em Matlab

```
a=1
b=a
c=1
n1=1
n2=1
Densidade=0.20
MaxNodulos=64
amostras=5
Nodulos=2
E1=3240.0e-06;
```

```
poisson1=0.25;
E2=40.0e-06;
poisson2=0.40;
```

B.2.2 Listagem do Módulo 2 programado em Matlab

```
cant1=18;           %quantidade de pontos por valor de z
cant2=18;           %quantidade de isocurvas z

maximo=max(max(a,b),c)

X=ones(cant2,cant1);
Y=ones(cant2,cant1);
Z=ones(cant2,cant1);
XX=ones(cant2+1,cant1+1,Nodulos);
YY=ones(cant2+1,cant1+1,Nodulos);
ZZ=ones(cant2+1,cant1+1,Nodulos);

i=1;
while i<=cant2+1;
    ii=(i-1)/cant2*pi-pi/2;
    j=1;
    while j<=cant1+1;
        jj=(j-1)/cant1*2*pi-pi;
        X(i,j)=a*abs(cos(ii)).^(n1).*abs(cos(jj)).^(n2)*sign(cos(ii))*
        sign(cos(jj));
        Z(i,j)=b*abs(cos(ii)).^(n1).*abs(sin(jj)).^(n2)*sign(cos(ii))*
        sign(sin(jj));
        Y(i,j)=c*abs(sin(ii)).^(n1)*sign(sin(ii));
        j=j+1;
    end
    i=i+1;
end

% -----Calculo do volume dos nódulos e do cubo-----
% -----obedecendo a densidade de volume estabelecida-----
r=2/n2;
t=2/n1;
F = @(x)sin(x).^(2/r-1).*cos(x).^(2/r-1);
Beta1=2*quad(F,0,pi/2);

F = @(x)sin(x).^(4/t-1).*cos(x).^(2/t-1);
Beta2=2*quad(F,0,pi/2);
Vnodulo=2/3*a*b*c*4/(r*t)*Beta1*Beta2;

while Nodulos<=MaxNodulos

Vnodulos=Nodulos*Vnodulo;    % Volume de n nodulos
```

```

Vcubo=Vnodulos/Densidade; %Volume da amostra para a densidade
desejada
Densidaderelativa=Nodulos*4/3*pi*maximo^3/Vcubo

% --Limites do ambiente onde gerar os centros dos nódulos-----

Xcub0=0.;
Xcub1=(Vcubo)^(1/3); % Dimensão do cubo
Ycub0=0.;
Ycub1=(Vcubo)^(1/3); % Dimensão do cubo
Zcub0=0.;
Zcub1=(Vcubo)^(1/3); % Dimensão do cubo
Lx=Xcub1-Xcub0;
Ly=Ycub1-Ycub0;
Lz=Zcub1-Zcub0;
if maximo>=Lx/2
    maximo
    Lx/2
fprintf('no entra en el cubo!!!\n');
off
end

rr=1
while rr<=amostras

% -----Gera os nódulos-----
i=1;
while i<=Nodulos
lim_iter(i)=350;
i=i+1;
end

ntent=150;

kk=1;
while kk<ntent
    k=1;
    while k<=Nodulos
        d=ones(cant2+1,cant1+1,Nodulos);
        D=zeros(cant2+1,cant1+1,Nodulos);
        iter=0;

        while any(any(any(d>D)))>0.5;
            iter=iter+1; % para saber quantas vezes uma
configuração boa
            D=ones(cant2+1,cant1+1,Nodulos);

% ---- Gira e movimenta a hiperelipsóide -----
cos1(k)=rand(1,1);
cos2(k)=rand(1,1);
cos3(k)=rand(1,1);

```

```

Ca=2*cos1(k)-1;
Cb=2*cos2(k)-1;
Cg=2*cos3(k)-1;

alfa(k)=acos(Ca)*180/pi;
beta(k)=acos(Cb)*180/pi;
gama(k)=acos(Cg)*180/pi;

Sa=(1-Ca^2)^(1/2);
Sb=(1-Cb^2)^(1/2);
Sg=(1-Cg^2)^(1/2);

Rx1=Ca*Cg-Sa*Cb*Sg;
Ry1=Sa*Cg+Ca*Cb*Sg;
Rz1=Sb*Sg;
Rx2=-Ca*Sg-Sa*Cb*Cg;
Ry2=-Sa*Sg+Ca*Cb*Cg;
Rz2=Sb*Cg;
Rx3=Sb*Sa;
Ry3=-Sb*Ca;
Rz3=Cb;

X1=X*Rx1+Y*Ry1+Z*Rz1;
Y1=X*Rx2+Y*Ry2+Z*Rz2;
Z1=X*Rx3+Y*Ry3+Z*Rz3;

% ---Move-se a coordenada do centro do Hiperelipsóide-----
XX(:, :, k)=X1;
YY(:, :, k)=Y1;
ZZ(:, :, k)=Z1;
coef=0.01; % Coeficiente para afastar os nódulos das bordas
do cubo. % coef=0.01 (significa 1% do Raio de afastamento
entre a % superficie do nodulo e a parede da matriz
cúbica)

while or(min(min(XX(:, :, k)))<=coef*Lx ,
max(max(XX(:, :, k)))>=Lx*(1-coef))
    x(k)=rand(1,1)*Lx;
    XX(:, :, k)=X1+x(k);
end
while or(min(min(YY(:, :, k)))<=coef*Ly ,
max(max(YY(:, :, k)))>=Ly*(1-coef))
    y(k)=rand(1,1)*Ly;
    YY(:, :, k)=Y1+y(k);
end
while or(min(min(ZZ(:, :, k)))<=coef*Lz ,
max(max(ZZ(:, :, k)))>=Lz*(1-coef))
    z(k)=rand(1,1)*Lz;

```

```

        ZZ(:, :, k) = Z1 + z(k);
    end

    if k > 1
        % ----- Calculo de distâncias -----
        j = 1;
        while j < k
            sep = 0.96; % Coeficiente para afastar os nódulos entre si

            % distancia^2 entre o centro j a cada ponto da superfície k
            D(:, :, j) = sep * ((XX(:, :, k) - x(j)).^2 + (YY(:, :, k) -
            y(j)).^2 + (ZZ(:, :, k) - z(j)).^2);
            i = 1;
            while i <= cant2 + 1;
                ii = 1;
                while ii <= cant1 + 1;
                    % matriz com produto interno
                    d(i, ii, j) = max(max((XX(:, :, j) - x(j)) * (XX(i, ii, k) -
                    x(j)) + (YY(:, :, j) - y(j)) * (YY(i, ii, k) - y(j)) + (ZZ(:, :, j) -
                    z(j)) * (ZZ(i, ii, k) - z(j)))));

                    ii = ii + 1;
                end
                i = i + 1;
            end
            j = j + 1;
        end
        end

        if iter > lim_iter(k);
            D = ones(cant2 + 1, cant1 + 1, Nodulos);
            d = ones(cant2 + 1, cant1 + 1, Nodulos);
        end

        end

        lim = lim_iter(k);
        if iter > lim;
            k = Nodulos + 1;
        end
        k = k + 1;
    end

    % critério de saída
    kk;
    if iter <= lim;
        kk = ntent + 1;
    else
        kk = kk + 1;
    end
end
end

```



```

if iter<=lim ;
    fprintf('configuracao boa:\n');
else
    fprintf('configuracao ruim:\n');
    off
end

figura2(XX,YY,ZZ,Nodulos,Lx);

minimo=min(min(a,b),c);

```

B.2.3 Listagem do Módulo 3 programado em Matlab

```

% Chamar a sub-rotina imprime_py para modificar a macro base

imprime_py(X,Y,Z,Lx,x,y,z,alfa,beta,gama,Nodulos,cant2,minimo,
E1,E2,poisson1,poisson2);

```

Sub-rotina *imprime_py*:

```

function
imprime_py(X,Y,Z,Lx,x,y,z,alfa,beta,gama,Nodulos,cant2,minimo,
E1,E2,poisson1,poisson2)

sourcepath = 'D:\Gilson\Matlab_Nódulos4'; %Pasta que contém
este programa
outputpath = 'D:\Gilson\Matlab_Nódulos4'; %Pasta onde o
arquivo de medias será gravado

arquivo1='Copia_abaqusMacros.py';
arquivo2='abaqusMacros.py';

A=importdata(arquivo1,'\n');
fid2 = fopen(arquivo2,'w');

i=1;
while i<=28;
    fprintf(fid2,'%s\n',A{i});
    i=i+1;
end
fprintf(fid2,'%g',Lx);
fprintf(fid2,',');
fprintf(fid2,'%g',Lx);
fprintf(fid2,')\n');
i=30;
while i<=33;
    fprintf(fid2,'%s\n',A{i});
    i=i+1;
end

```

```

fprintf(fid2, '%g', Lx);
fprintf(fid2, ')\n');

i=35;
while i<=46;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
%---coordenadas dos pontos-----
j=1;
while j<=cant2+1;
fprintf(fid2, '%g', X(j,1));
fprintf(fid2, ', ');
fprintf(fid2, '%g', Y(j,1));
if j<cant2+1;
    fprintf(fid2, '), (\n');
else
    fprintf(fid2, '))\n');
end
j=j+1;
end

fprintf(fid2, '    s1.Line(point1=(0.0, ');
fprintf(fid2, '%g', Y(1,1));
fprintf(fid2, '), point2=(0.0, ');
fprintf(fid2, '%g', Y(cant2+1,1));
fprintf(fid2, '))\n');

i=89;
while i<=102;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

% começa a copiar e gerar a quantidade de nódulos que se tem
k=1;
while k<=Nodulos;
fprintf(fid2, '    a1 = mdb.models[''Model-
1''].rootAssembly\n');
fprintf(fid2, '    p = mdb.models[''Model-
1''].parts[''nodulo'']\n');

fprintf(fid2, '    a1.Instance(name='''nodulo-'');
fprintf(fid2, '%u''', k);
fprintf(fid2, ', part=p, dependent=OFF)\n');
    fprintf(fid2, '    a1 = mdb.models[''Model-
1''].rootAssembly\n');

fprintf(fid2, '    a1.rotate(instanceList=('''nodulo-'');
fprintf(fid2, '%u''', k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');

```

```

fprintf(fid2, '    axisDirection=(0,0,1), angle=');
fprintf(fid2, '%g', -alfa(k));
fprintf(fid2, ')\n');

fprintf(fid2, '    a1.rotate(instanceList=('nodulo-');
fprintf(fid2, '%u'',k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');
fprintf(fid2, '    axisDirection=(1,0,0), angle=');
fprintf(fid2, '%g', -beta(k));
fprintf(fid2, ')\n');

fprintf(fid2, '    a1.rotate(instanceList=('nodulo-');
fprintf(fid2, '%u'',k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');
fprintf(fid2, '    axisDirection=(0,0,1), angle=');
fprintf(fid2, '%g', -gama(k));
fprintf(fid2, ')\n');

fprintf(fid2, '    a1.translate(instanceList=('nodulo-');
fprintf(fid2, '%u'',k);
fprintf(fid2, ', ), vector=');
fprintf(fid2, '%g,%g,%g', x(k), y(k), z(k));
fprintf(fid2, '))\n\n');
k=k+1;
end

i=164;
while i<=171;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
fprintf(fid2, '    a1.instances['nodulo-');
fprintf(fid2, '%u'',k);
k=k+1;
end

i=180;
while i<=196;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    'nodulo-%u')\n', k);
else
    fprintf(fid2, '    'nodulo-%u', \n', k);

```

```

end
k=k+1;
end

fprintf(fid2, '    session.viewports[''Viewport:
1''].view.fitView()\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');
fprintf(fid2, '    s1 = a.instances[''cubo_vazado-
1''].faces\n');
fprintf(fid2, '    sidelFaces1 = s1\n');
fprintf(fid2, '
region1=regionToolset.Region(sidelFaces=sidelFaces1)\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');

k=1;
while k<=Nodulos;
    fprintf(fid2, '    s%u = a.instances[''nodulo-
%u''].faces\n',k,k);
    fprintf(fid2, '    sidelFaces%u =
s%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n',k,k);
    k=k+1;
end

fprintf(fid2, '
region2=regionToolset.Region(sidelFaces=\n');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    sidelFaces%u)\n',k);
else
    fprintf(fid2, '    sidelFaces%u+\\\n',k);
end
k=k+1;
end

i=236;
while i<=241;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    ''nodulo-%u'')\n',k);
else
    fprintf(fid2, '    ''nodulo-%u'',\n',k);
end
k=k+1;

```

```

end

i=250;
while i<=272;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

fprintf(fid2, '    a.seedPartInstance(regions=partInstances,
size=%g, deviationFactor=0.1)\n', minimo/(3.33));

i=274;
while i<=278;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    'nodulo-%u')\n', k);
else
    fprintf(fid2, '    'nodulo-%u', \n', k);
end
k=k+1;
end

fprintf(fid2, '    session.viewports[''Viewport:
1''].view.fitView()\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');

k=1;
while k<=Nodulos;
    fprintf(fid2, '    c%u = a.instances[''nodulo-
%u''].cells\n', k, k);
    fprintf(fid2, '    cells%u =
c%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n', k, k);
    k=k+1;
end
fprintf(fid2, '    pickedRegions = ');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, 'cells%u\n', k);
else
    fprintf(fid2, 'cells%u+', k);
end
k=k+1;
end

```

```

i=306;
while i<=311;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
    fprintf(fid2, '    c%u = a.instances[''nodulo-
%u''].cells\n', k, k);
    fprintf(fid2, '    cells%u =
c%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n', k, k);
    k=k+1;
end
fprintf(fid2, '    pickedRegions = ((');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, 'cells%u), )\n', k);
else
    fprintf(fid2, 'cells%u+', k);
end
k=k+1;
end

i=329;
while i<=332;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    a.instances[''nodulo-%u''], )\n', k);
else
    fprintf(fid2, '    a.instances[''nodulo-%u''], \n', k);
end
k=k+1;
end

fprintf(fid2, '# size=minimo\n');
fprintf(fid2, '    a.seedPartInstance(regions=partInstances,
size=%g, deviationFactor=0.1)\n', minimo*1.5);
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');
fprintf(fid2, '    partInstances = (\n');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    a.instances[''nodulo-%u''], )\n', k);
else

```

```

        fprintf(fid2, '          a.instances['nodulo-%u'],\n',k);
    end
    k=k+1;
end

i=353;
while i<=368;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

i=370;
while i<=386;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end
k=1;
while k<=Nodulos;
    if k==Nodulos;
        fprintf(fid2, '          'nodulo-%u'))\n',k);
    else
        fprintf(fid2, '          'nodulo-%u',\n',k);
    end
    k=k+1;
end
i=395;
while i<=402;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

%-----Aqui se pode mudar as propriedades do material-----

fprintf(fid2, '# Cubo PMMA elastico (E[N/micrometro^2])\n');
fprintf(fid2, '      mdb.models['Model-
1'].Material(name='PMMA')\n');
fprintf(fid2, '      mdb.models['Model-
1'].materials['PMMA'].Elastic(table=((%g, %g),
))\n',E1,poisson1);
fprintf(fid2, '# Nodulos borracha elastica\n');
fprintf(fid2, '      mdb.models['Model-
1'].Material(name='borracha')\n');
fprintf(fid2, '      mdb.models['Model-
1'].materials['borracha'].Elastic(table=((%g, %g),
))\n',E2,poisson2);

i=409;
while i<=535;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

```

```

k=1;
fprintf(fid2,'      cliCommand("""elemArr = mdb.models[''Model-
1''].rootAssembly.instances[''nodulo-%u''].elements""")\n',k);
fprintf(fid2,'      cliCommand("""len(elemArr)""")\n');

fprintf(fid2,'      cliCommand("""elemArr = mdb.models[''Model-
1''].rootAssembly.instances[''cubo_vazado-
1''].elements""")\n');
fprintf(fid2,'      cliCommand("""len(elemArr)""")\n');

fprintf(fid2,'      cliCommand("""nodeArr = mdb.models[''Model-
1''].rootAssembly.instances[''nodulo-%u''].nodes""")\n',k);
fprintf(fid2,'      cliCommand("""len(nodeArr)""")\n');
fprintf(fid2,'      cliCommand("""nodeArr = mdb.models[''Model-
1''].rootAssembly.instances[''cubo_vazado-1''].nodes""")\n');
fprintf(fid2,'      cliCommand("""len(nodeArr)""")\n');

fclose(fid2);

```

B.2.4 Listagem do Módulo 4 programado em Matlab

```

%-----Executa no Abaqus a Macro Abaqus Modificada-----

istatus = dos('abaqus cae noGUI="abaqus.py"');

```

B.2.5 Listagem do Módulo 5 programado em Matlab

```

% Sub-rotina calcula_prop onde se calculam as propriedades
efetivas

```

```

calcula_prop(Nodulos,E1,E2,poisson1,poisson2,Densidade);

```

Sub-rotina *calcula_prop*:

```

function
calcula_prop(Nodulos,E1,E2,poisson1,poisson2,Densidade);

file='abaqus.rpy';
fid=fopen(file,'r');
C=textscan(fid,'%s\n',4,'HeaderLines',26,'Delimiter',' ','...
            'commentStyle','//');
Elemnodo=C{1,2}(1);
Elemcubo=C{1,2}(2);
Nodosnodo=C{1,2}(3);
Nodoscubo=C{1,2}(4);
fclose(fid);

```



```

%-----Lê o arquivo de tensões, deformações e volumes-----

file='saída.rpt';
fid=fopen(file,'r');

Rescub=textscan(fid,'%f%f%f%f%f%f%f%f%f%f%f%f%f%f',Elemcubo,
'HeaderLines',19,...
'commentStyle','//');
k=1;
while k<=Nodulos
Resnod{k,:}=textscan(fid,'%f%f%f%f%f%f%f%f%f%f%f%f%f%f',Elem
nodo,'HeaderLines',19,...
'commentStyle','//');
k=k+1;
end
Volcub=textscan(fid,'%f%f',Elemcubo,'HeaderLines',33,...
'commentStyle','//');
k=1;
while k<=Nodulos;
Volnod{k,:}=textscan(fid,'%f%f',Elemnodo,'HeaderLines',17,...
'commentStyle','//');
k=k+1;
end

fclose(fid);
delete('C:\Temp\abaqus.rpy')
delete('C:\Temp\saida.rpt')

%----Cálculo das propriedades efetivas do composto-----

tracoScubo=0;
tracoEcubo=0;
i=1;
while i<=Elemcubo
pScub(i)=(Rescub{1,10}(i)+Rescub{1,11}(i)+Rescub{1,12}(i))/3;
pEcub(i)=(Rescub{1,3}(i)+Rescub{1,4}(i)+Rescub{1,5}(i))/3;
tracoScubo=tracoScubo+pScub(i)*Volcub{1,2}(i);
tracoEcubo=tracoEcubo+pEcub(i)*Volcub{1,2}(i);
i=i+1;
end

tracoSnodulos=0;
tracoEnodulos=0;
k=1;
while k<=Nodulos;
i=1;
tracoSnodulo(k)=0;
tracoEnodulo(k)=0;
while i<=Elemnodo;
pSnod(i,k)=(Resnod{k,1}{1,10}(i)+Resnod{k,1}{1,11}(i)+Resnod{k
,1}{1,12}(i))/3;

```

```

pEnod(i,k)=(Resnod{k,1}{1,3}(i)+Resnod{k,1}{1,4}(i)+Resnod{k,1}{1,5}(i))/3;
tracoSnodulo(k)=tracoSnodulo(k)+pSnod(i,k)*Volnod{k,1}{1,2}(i)
;
tracoEnodulo(k)=tracoEnodulo(k)+pEnod(i,k)*Volnod{k,1}{1,2}(i)
;
i=i+1;
    end
tracoSnodulos=tracoSnodulos+tracoSnodulo(k);
tracoEnodulos=tracoEnodulos+tracoEnodulo(k);
    k=k+1;
end

Kequiv=1/3*(tracoSnodulos+tracoScubo)/(tracoEnodulos+tracoEcubo);

%-----TENSÕES DESVIADORAS -----
-
E11cubo=0;
E22cubo=0;
E33cubo=0;
E12cubo=0;
E13cubo=0;
E23cubo=0;
S11cubo=0;
S22cubo=0;
S33cubo=0;
S12cubo=0;
S13cubo=0;
S23cubo=0;

i=1;
while i<=Elemcubo;
E11cubo=E11cubo+(Rescub{1,3}(i)-pEcub(i))*(Volcub{1,2}(i));
E22cubo=E22cubo+(Rescub{1,4}(i)-pEcub(i))*(Volcub{1,2}(i));
E33cubo=E33cubo+(Rescub{1,5}(i)-pEcub(i))*(Volcub{1,2}(i));
E12cubo=E12cubo+Rescub{1,6}(i)/2*(Volcub{1,2}(i));
E13cubo=E13cubo+Rescub{1,7}(i)/2*(Volcub{1,2}(i));
E23cubo=E23cubo+Rescub{1,8}(i)/2*(Volcub{1,2}(i));

S11cubo=S11cubo+(Rescub{1,10}(i)-pScub(i))*(Volcub{1,2}(i));
S22cubo=S22cubo+(Rescub{1,11}(i)-pScub(i))*(Volcub{1,2}(i));
S33cubo=S33cubo+(Rescub{1,12}(i)-pScub(i))*(Volcub{1,2}(i));
S12cubo=S12cubo+Rescub{1,13}(i)*(Volcub{1,2}(i));
S13cubo=S13cubo+Rescub{1,14}(i)*(Volcub{1,2}(i));
S23cubo=S23cubo+Rescub{1,15}(i)*(Volcub{1,2}(i));

i=i+1;
end
E11nodulos=0;
E22nodulos=0;

```

```

E33nodulos=0;
E12nodulos=0;
E13nodulos=0;
E23nodulos=0;
S11nodulos=0;
S22nodulos=0;
S33nodulos=0;
S12nodulos=0;
S13nodulos=0;
S23nodulos=0;
k=1;
while k<=Nodulos;
    i=1;
    E11nodulo(k)=0;
    E22nodulo(k)=0;
    E33nodulo(k)=0;
    E12nodulo(k)=0;
    E13nodulo(k)=0;
    E23nodulo(k)=0;
    S11nodulo(k)=0;
    S22nodulo(k)=0;
    S33nodulo(k)=0;
    S12nodulo(k)=0;
    S13nodulo(k)=0;
    S23nodulo(k)=0;
    while i<=Elemnodo;
E11nodulo(k)=E11nodulo(k)+(Resnod{k,1}{1,3}(i)-
pEnod(i,k))*(Volnod{k,1}{1,2}(i));
E22nodulo(k)=E22nodulo(k)+(Resnod{k,1}{1,4}(i)-
pEnod(i,k))*(Volnod{k,1}{1,2}(i));
E33nodulo(k)=E33nodulo(k)+(Resnod{k,1}{1,5}(i)-
pEnod(i,k))*(Volnod{k,1}{1,2}(i));
E12nodulo(k)=E12nodulo(k)+Resnod{k,1}{1,6}(i)/2*(Volnod{k,1}{1,
2}(i));
E13nodulo(k)=E13nodulo(k)+Resnod{k,1}{1,7}(i)/2*(Volnod{k,1}{1,
2}(i));
E23nodulo(k)=E23nodulo(k)+Resnod{k,1}{1,8}(i)/2*(Volnod{k,1}{1,
2}(i));

S11nodulo(k)=S11nodulo(k)+(Resnod{k,1}{1,10}(i)-
pSnod(i,k))*(Volnod{k,1}{1,2}(i));
S22nodulo(k)=S22nodulo(k)+(Resnod{k,1}{1,11}(i)-
pSnod(i,k))*(Volnod{k,1}{1,2}(i));
S33nodulo(k)=S33nodulo(k)+(Resnod{k,1}{1,12}(i)-
pSnod(i,k))*(Volnod{k,1}{1,2}(i));
S12nodulo(k)=S12nodulo(k)+Resnod{k,1}{1,13}(i)*(Volnod{k,1}{1,
2}(i));
S13nodulo(k)=S13nodulo(k)+Resnod{k,1}{1,14}(i)*(Volnod{k,1}{1,
2}(i));
S23nodulo(k)=S23nodulo(k)+Resnod{k,1}{1,15}(i)*(Volnod{k,1}{1,
2}(i));

```

```

i=i+1;
    end
E11nodulos=E11nodulos+E11nodulo(k);
E22nodulos=E22nodulos+E22nodulo(k);
E33nodulos=E33nodulos+E33nodulo(k);
E12nodulos=E12nodulos+E12nodulo(k);
E13nodulos=E13nodulos+E13nodulo(k);
E23nodulos=E23nodulos+E23nodulo(k);

S11nodulos=S11nodulos+S11nodulo(k);
S22nodulos=S22nodulos+S22nodulo(k);
S33nodulos=S33nodulos+S33nodulo(k);
S12nodulos=S12nodulos+S12nodulo(k);
S13nodulos=S13nodulos+S13nodulo(k);
S23nodulos=S23nodulos+S23nodulo(k);

    k=k+1;
end

E11=E11cubo+E11nodulos;
E22=E22cubo+E22nodulos;
E33=E33cubo+E33nodulos;
E12=E12cubo+E12nodulos;
E13=E13cubo+E13nodulos;
E23=E23cubo+E23nodulos;

S11=S11cubo+S11nodulos;
S22=S22cubo+S22nodulos;
S33=S33cubo+S33nodulos;
S12=S12cubo+S12nodulos;
S13=S13cubo+S13nodulos;
S23=S23cubo+S23nodulos;

desvE=(E11^2+E22^2+E33^2+2*E12^2+2*E13^2+2*E23^2);
desvS=(S11^2+S22^2+S33^2+2*S12^2+2*S13^2+2*S23^2);

Nuequiv=1/2*((desvS)/(desvE))^(1/2);

% ----- imprime os dados -----

arquivo='data.dat';
fid = fopen(arquivo,'w');

    fprintf(fid,'%g\n',Kequiv);
    fprintf(fid,'%g\n',Nuequiv);

fclose(fid);

end

```

B.2.6 Listagem do Módulo 6 programado em Matlab

```

file='data.dat';
fid=fopen(file,'r');
Kequiv(rr)=textscan(fid,'%f',1,'commentStyle','//');
Nuequiv(rr)=textscan(fid,'%f',1,'commentStyle','//');
fclose(fid);
rr=rr+1;
end

rr=1;
somaK=0.0;
somaK2=0.0;
somaNu=0.0;
somaNu2=0.0;

while rr<=amostras
somaK=somaK+Kequiv{1,rr}(1);
somaK2=somaK2+Kequiv{1,rr}(1)*Kequiv{1,rr}(1);
somaNu=somaNu+Nuequiv{1,rr}(1);
somaNu2=somaNu2+Nuequiv{1,rr}(1)*Nuequiv{1,rr}(1);
rr=rr+1;
end
mediaK=somaK/amostras;
mediaNu=somaNu/amostras;
DesvioK=((somaK2-amostras*mediaK^2)/(amostras-1))^(1/2);
DesvioNu=((somaNu2-amostras*mediaNu^2)/(amostras-1))^(1/2);

```

B.2.7 Listagem do Módulo 7 programado em Matlab

```

%-----Cálculo dos Valores e limites analíticos-----

K1=E1/(3*(1-2*poisson1));
K2=E2/(3*(1-2*poisson2));
Nu1=E1/(2*(1+poisson1));
Nu2=E2/(2*(1+poisson2));

% Limites de Reuss
Kreuss=1/((1/(K1*(1-Densidade)))+(1/(Densidade*K2)));
Nureuss=1/((1/(Nu1*(1-Densidade)))+(1/(Densidade*Nu2)));

% Limites de Voigt
Kvoigt=K1*(1-Densidade)+K2*Densidade;
Nuvoigt=Nu1*(1-Densidade)+Nu2*Densidade;

% DILUTE
alfa=((3*K1)/((3*K1)+(4*Nu1)));
beta=(6*K1+12*Nu1)/(15*K1+20*Nu1);

```

```

Kdilute=K1+(Densidade*((K2-K1)*K1)/(K1+alfa*(K2-K1)));
Nudilute=Nu1+(Densidade*((Nu2-Nu1)*Nu1)/(Nu1+beta*(Nu2-
Nu1)));

% Mori-Tanaka
teta=K1+(4*Nu1)/3;
fi=Nu1+((9*K1*Nu1+8*Nu1*Nu1)/(6*K1+12*Nu1));
eta=(teta/(teta+(1-Densidade)*(K2-K1)));
tau=(fi/(fi+(1-Densidade)*(Nu2-Nu1)));
Kmt=K1+((K2-K1)*eta*Densidade);
Numt=Nu1+((Nu2-Nu1)*tau*Densidade);

file='resultados.dat';
fid=fopen(file,'a');
fprintf(fid,'Cantidad de Nódulos: %6.2f\n',Nodulos);
fprintf(fid,'K obtido com FEM:\n');
rr=1
while rr<=amostras
fprintf(fid,'%g ',Kequiv{1,rr}(1));
rr=rr+1;
end
fprintf(fid,'\n');
fprintf(fid,'media: %g\n',mediaK);
fprintf(fid,'desvio padrão: %g\n',DesvioK);
fprintf(fid,'\n');
fprintf(fid,'G obtido com FEM:\n');
rr=1
while rr<=amostras
fprintf(fid,'%g ',Nuequiv{1,rr}(1));
rr=rr+1;
end
fprintf(fid,'\n');
fprintf(fid,'media: %g\n',mediaNu);
fprintf(fid,'desvio padrão: %g\n',DesvioNu);
fprintf(fid,'\n');

fprintf(fid,'Reuss: %g\n', Kreuss);
fprintf(fid,' Reuss: %g\n', Nureuss);
fprintf(fid,'Voigt: %g\n', Kvoigt);
fprintf(fid,'Voigt: %g\n', Nuvoigt);
fprintf(fid,'DILUTE: %g\n', Kdilute);
fprintf(fid,'DILUTE: %g\n', Nudilute);
fprintf(fid,'Mori-Tanaka: %g\n', Kmt);
fprintf(fid,'Mori-Tanaka: %g\n', Numt);

fclose(fid);
passo=Nodulos;
Nodulos=Nodulos+passo
end

```

B.2.8 Listagem do arquivo.dat da Macro Abaqus Base (arquivo “Cópia_abaqusMacros”)

```

# Do not delete the following import lines
from abaqus import *
from abaqusConstants import *
# Abertura Macro Geral
def Assembly_cubo():
    import section
    import section
    import regionToolset
    import displayGroupMdbToolset as dgm
    import part
    import material
    import assembly
    import step
    import interaction
    import load
    import mesh
    import job
    import sketch
    import visualization
    import xyPlot
    import displayGroupOdbToolset as dgo
    import connectorBehavior
# Cria o cubo_macico
    s = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
    sheetSize=1000.0)
    g, v, d, c = s.geometry, s.vertices, s.dimensions,
s.constraints
    s.setPrimaryObject(option=STANDALONE)
    s.rectangle(point1=(0.0, 0.0), point2=(
53.5, 53.5))
    p = mdb.models['Model-1'].Part(name='cubo_macico',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
    p = mdb.models['Model-1'].parts['cubo_macico']
    p.BaseSolidExtrude(sketch=s, depth=
53.5)
    s.unsetPrimaryObject()
    p = mdb.models['Model-1'].parts['cubo_macico']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Cria o nodulo
    s1 = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
    sheetSize=1000.0)

```

```

    g, v, d, c = s1.geometry, s1.vertices, s1.dimensions,
s1.constraints
    s1.setPrimaryObject(option=STANDALONE)
    s1.ConstructionLine(point1=(0.0, -500.0), point2=(0.0,
500.0))
    s1.FixedConstraint(entity=g[2])
    s1.Spline(points=((
0.0,10),(
0.5, 6.027864045),(
1, 4.67544468),(
1.5, 3.754033308),(
2, 3.05572809),(
2.5, 2.5),(
3, 2.04554885),(
3.5, 1.667840434),(
4, 1.350889359),(
4.5, 1.083592135),(
5, 0.857864376),(
5.5, 0.667603026),(
6, 0.508066615),(
6.5, 0.375484503),(
7, 0.266799469),(
7.5, 0.179491924),(
8, 0.11145618),(
8.5, 0.060911085),(
9, 0.026334039),(
9.5, 0.00641131),(
10,0),(
9.5,-0.00641131),(
9,-0.026334039),(
8.5,-0.060911085),(
8,-0.11145618),(
7.5,-0.179491924),(
7,-0.266799469),(
6.5,-0.375484503),(
6,-0.508066615),(
5.5,-0.667603026),(
5,-0.857864376),(
4.5,-1.083592135),(
4,-1.350889359),(
3.5,-1.667840434),(
3,-2.04554885),(
2.5,-2.5),(
2,-3.05572809),(
1.5,-3.754033308),(
1,-4.67544468),(
0.5,-6.027864045),(
0,-10)))
    s1.Line(point1=(0.0,10.0), point2=(0.0,-10.0))
    s1.VerticalConstraint(entity=g[4])

```



```

    p = mdb.models['Model-1'].Part(name='nodulo',
dimensionality=THREE_D,
        type=DEFORMABLE_BODY)
    p = mdb.models['Model-1'].parts['nodulo']
    p.BaseSolidRevolve(sketch=s1, angle=360.0,
flipRevolveDirection=OFF)
    s1.unsetPrimaryObject()
    p = mdb.models['Model-1'].parts['nodulo']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Assembly cubo_macico
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['cubo_macico']
    a1.Instance(name='cubo_macico-1', part=p, dependent=OFF)
# Insere os nodulos no cubo Distribuicao aleatoria
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-1', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,1,0), angle=51.68)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=90.0)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,1,0), angle=91.14)
    a1.translate(instanceList=('nodulo-1', ),
vector=(24,10.1,18.9))
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-2', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.translate(instanceList=('nodulo-2', ),
vector=(33.2,43.4,19.3))
    a1.rotate(instanceList=('nodulo-2', ),
axisPoint=(33.2,43.4,19.3),
axisDirection=(0,1,0), angle=51.68)
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-3', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.translate(instanceList=('nodulo-3', ),
vector=(12.4,21.6,34.9))
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-4', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.translate(instanceList=('nodulo-4', ),
vector=(29.2,36.7,42.5))
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']

```

```

a1.Instance(name='nodulo-5', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-5', ),
vector=(30,16.1,35.8))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-6', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-6', ),
vector=(38.3,15.2,14.7))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-7', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-7', ),
vector=(13.8,31.7,10.2))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-8', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-8', ),
vector=(13,42,37.8))
# Cria os vazios dos nodulos no cubo( cria a part cubo_vazado)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].view.fitView()
a1 = mdb.models['Model-1'].rootAssembly
a1.PartFromBooleanCut(name='cubo_vazado',
instanceToBeCut=mdb.models['Model-
1'].rootAssembly.instances['cubo_macico-1'],
cuttingInstances=(
a1.instances['nodulo-1'],
a1.instances['nodulo-2'],
a1.instances['nodulo-3'],
a1.instances['nodulo-4'],
a1.instances['nodulo-5'],
a1.instances['nodulo-6'],
a1.instances['nodulo-7'],
a1.instances['nodulo-8'],
))
# cria a instance cubo_vazado-1 no Assembly)
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['cubo_vazado']
a1.Instance(name='cubo_vazado-1', part=p, dependent=OFF)
# Inserindo os nodulos nos vazios do cubo vazado
# Contato: tipo TIE
session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=ON,
interactions=ON, constraints=ON, connectors=ON,
engineeringFeatures=ON,

```

```

        adaptiveMeshConstraints=OFF)
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1', ))
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['cubo_vazado-1'].faces
    sidelFaces1 = s1.getSequenceFromMask(mask=('[#3fffc0 ]',
), )
    region1=regionToolset.Region(sidelFaces=sidelFaces1)
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['nodulo-1'].faces
    sidelFaces1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
    s2 = a.instances['nodulo-2'].faces
    sidelFaces2 = s2.getSequenceFromMask(mask=('[#1 ]', ), )
    s3 = a.instances['nodulo-3'].faces
    sidelFaces3 = s3.getSequenceFromMask(mask=('[#1 ]', ), )
    s4 = a.instances['nodulo-4'].faces
    sidelFaces4 = s4.getSequenceFromMask(mask=('[#1 ]', ), )
    s5 = a.instances['nodulo-5'].faces
    sidelFaces5 = s5.getSequenceFromMask(mask=('[#1 ]', ), )
    s6 = a.instances['nodulo-6'].faces
    sidelFaces6 = s6.getSequenceFromMask(mask=('[#1 ]', ), )
    s7 = a.instances['nodulo-7'].faces
    sidelFaces7 = s7.getSequenceFromMask(mask=('[#1 ]', ), )
    s8 = a.instances['nodulo-8'].faces
    sidelFaces8 = s8.getSequenceFromMask(mask=('[#1 ]', ), )
    region2=regionToolset.Region(sidelFaces=
        sidelFaces1+\
        sidelFaces2+\
        sidelFaces3+\
        sidelFaces4+\
        sidelFaces5+\
        sidelFaces6+\
        sidelFaces7+\

```

```

        sidelFaces8)
        mdb.models['Model-1'].Tie(name='tie', master=region1,
slave=region2,
        positionToleranceMethod=COMPUTED, adjust=ON,
tieRotations=ON,
        thickness=ON)
# Ajuste na tela
        session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1',
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
        session.viewports['Viewport: 1'].view.fitView()
# Criacao da malha 1 (14x14x7):
# Malha do cubo_vazado-1:
        session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1', ))
        session.viewports['Viewport: 1'].view.fitView()
        a = mdb.models['Model-1'].rootAssembly
        c1 = a.instances['cubo_vazado-1'].cells
        pickedRegions = c1.getSequenceFromMask(mask=('[#1 ]', ), )
        a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
        elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=STANDARD)
        elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=STANDARD)
        elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=STANDARD,
        distortionControl=DEFAULT)
        a = mdb.models['Model-1'].rootAssembly
        c1 = a.instances['cubo_vazado-1'].cells
        cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
        pickedRegions =(cells1, )
        a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
        elemType3))
        a = mdb.models['Model-1'].rootAssembly
        partInstances =(a.instances['cubo_vazado-1'], )
# size=Lcubo2nodulos/7
        a.seedPartInstance(regions=partInstances, size=17.96,
deviationFactor=0.1)
        a = mdb.models['Model-1'].rootAssembly
        partInstances =(a.instances['cubo_vazado-1'], )

```

```

    a.generateMesh(regions=partInstances)
# Malha dos nodulos:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    c3 = a.instances['nodulo-3'].cells
    cells3 = c3.getSequenceFromMask(mask=('[#1 ]', ), )
    c4 = a.instances['nodulo-4'].cells
    cells4 = c4.getSequenceFromMask(mask=('[#1 ]', ), )
    c5 = a.instances['nodulo-5'].cells
    cells5 = c5.getSequenceFromMask(mask=('[#1 ]', ), )
    c6 = a.instances['nodulo-6'].cells
    cells6 = c6.getSequenceFromMask(mask=('[#1 ]', ), )
    c7 = a.instances['nodulo-7'].cells
    cells7 = c7.getSequenceFromMask(mask=('[#1 ]', ), )
    c8 = a.instances['nodulo-8'].cells
    cells8 = c8.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions =
cells1+cells2+cells3+cells4+cells5+cells6+cells7+cells8
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=STANDARD)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=STANDARD)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=STANDARD,
        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    c3 = a.instances['nodulo-3'].cells
    cells3 = c3.getSequenceFromMask(mask=('[#1 ]', ), )
    c4 = a.instances['nodulo-4'].cells
    cells4 = c4.getSequenceFromMask(mask=('[#1 ]', ), )
    c5 = a.instances['nodulo-5'].cells

```

```

cells5 = c5.getSequenceFromMask(mask=('[#1 ]', ), )
c6 = a.instances['nodulo-6'].cells
cells6 = c6.getSequenceFromMask(mask=('[#1 ]', ), )
c7 = a.instances['nodulo-7'].cells
cells7 = c7.getSequenceFromMask(mask=('[#1 ]', ), )
c8 = a.instances['nodulo-8'].cells
cells8 = c8.getSequenceFromMask(mask=('[#1 ]', ), )
pickedRegions
=((cells1+cells2+cells3+cells4+cells5+cells6+cells7+cells8), )
a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
elemType3))
a = mdb.models['Model-1'].rootAssembly
partInstances = (
a.instances['nodulo-1'],
a.instances['nodulo-2'],
a.instances['nodulo-3'],
a.instances['nodulo-4'],
a.instances['nodulo-5'],
a.instances['nodulo-6'],
a.instances['nodulo-7'],
a.instances['nodulo-8'], )
# size=(2*Pi*R/2)/14
a.seedPartInstance(regions=partInstances, size=5.12,
deviationFactor=0.1)
a = mdb.models['Model-1'].rootAssembly
partInstances = (
a.instances['nodulo-1'],
a.instances['nodulo-2'],
a.instances['nodulo-3'],
a.instances['nodulo-4'],
a.instances['nodulo-5'],
a.instances['nodulo-6'],
a.instances['nodulo-7'],
a.instances['nodulo-8'], )
a.generateMesh(regions=partInstances)
# Ativa a malha do cubo_vazado-1:
a = mdb.models['Model-1'].rootAssembly
partInstances = (a.instances['cubo_vazado-1'], )
a.generateMesh(regions=partInstances)
# Ajuste na tela
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
a1 = mdb.models['Model-1'].rootAssembly
a1.regenerate()
session.viewports['Viewport:
1'].assemblyDisplay.setValues(
adaptiveMeshConstraints=OFF)
session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])

```

```

# Ativa nodulos e cubo_vazado-1
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF,
visibleInstances=('cubo_vazado-1',
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
# Supress cubo macico
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a = mdb.models['Model-1'].rootAssembly
    a.features['cubo_macico-1'].suppress()
# Cria materiais PMMA e Borracha
    session.viewports['Viewport:
1'].partDisplay.setValues(sectionAssignments=ON,
        engineeringFeatures=ON)
# Cubo PMMA elastico (E[N/micrometro^2])
    mdb.models['Model-1'].Material(name='PMMA')
    mdb.models['Model-
1'].materials['PMMA'].Elastic(table=((3240.0e-06, 0.25), ))
# Nodulos borracha elastica
    mdb.models['Model-1'].Material(name='borracha')
    mdb.models['Model-
1'].materials['borracha'].Elastic(table=((40.0e-06, 0.40), ))
# Cria secoes cubo e nodulo
    mdb.models['Model-1'].HomogeneousSolidSection(name='cubo',
material='PMMA',
        thickness=1.0)
    mdb.models['Model-
1'].HomogeneousSolidSection(name='nodulo',
        material='borracha', thickness=1.0)
# Assigna secao nodulo a nodulo
    p = mdb.models['Model-1'].parts['nodulo']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['nodulo']
    p.SectionAssignment(region=region, sectionName='nodulo',
offset=0.0)
# Assigna secao cubo a cubo vazado
    p = mdb.models['Model-1'].parts['cubo_vazado']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    p = mdb.models['Model-1'].parts['cubo_vazado']

```

```

c = p.cells
cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
region = regionToolset.Region(cells=cells)
p = mdb.models['Model-1'].parts['cubo_vazado']
p.SectionAssignment(region=region, sectionName='cubo',
offset=0.0)
# Cria o Step
session.viewports['Viewport:
1'].assemblyDisplay.setValues(loads=OFF, bcs=OFF,
predefinedFields=OFF, connectors=OFF,
adaptiveMeshConstraints=ON)
mdb.models['Model-1'].StaticStep(name='final',
previous='Initial')
session.viewports['Viewport:
1'].assemblyDisplay.setValues(step='final')
# Inclui efeito nao linear de grandes deformacoes e
deslocamentos no step final para rodar hiperelastico
a = mdb.models['Model-1'].rootAssembly
a.regenerate()
session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=OFF)
session.viewports['Viewport:
1'].assemblyDisplay.meshOptions.setValues(
meshTechnique=OFF)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
mdb.models['Model-1'].steps['final'].setValues(nlgeom=ON)
# Cria BC
# Deslocamento prescrito em y
session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
session.viewports['Viewport: 1'].view.fitView()
session.viewports['Viewport:
1'].view.setValues(session.views['Top'])
mdb.models['Model-
1'].ExpressionField(name='AnalyticalField-1', localCsys=None,
description='', expression='0.001*(X+Y+Z)')
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#2 ]', ), )
region = regionToolset.Region(faces=faces1)
mdb.models['Model-1'].DisplacementBC(name='BC-3',
createStepName='final',
region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
amplitude=UNSET, fixed=OFF, distributionType=FIELD,
fieldName='AnalyticalField-1', localCsys=None)
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#4 ]', ), )

```



```

    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-7',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Bottom'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#8 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-4',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#10 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-5',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#20 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-6',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-8',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)

```

```

# Tela branca

session.graphicsOptions.setValues(backgroundColor='#FFFFFF')
# Ajustes na tela
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
# opcoes .odb
    mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(variables=(
    'S', 'PE', 'PEEQ', 'PEMAG', 'LE', 'U', 'RF', 'CF',
    'CSTRESS', 'CDISP',
    'EVOL'))
# RUN
    mdb.Job(name='gill', model='Model-1', type=ANALYSIS,
explicitPrecision=SINGLE,
        nodalOutputPrecision=SINGLE, description='',
        parallelizationMethodExplicit=DOMAIN,
multiprocessingMode=DEFAULT,
        numDomains=1, userSubroutine='', numCpus=1,
preMemory=1536.0,
        standardMemory=1536.0, standardMemoryPolicy=MODERATE,
scratch='',
        echoPrint=OFF, modelPrint=OFF, contactPrint=OFF,
historyPrint=OFF)
    mdb.jobs['gill'].submit(consistencyChecking=OFF)
    mdb.jobs['gill'].waitForCompletion()
# saida de tensoes e deformacoes e inp
    o3 = session.openOdb(name='C:/Temp/gill.odb')
    session.viewports['Viewport:
1'].setValues(displayedObject=o3)
    session.viewports['Viewport:
1'].odbDisplay.display.setValues(plotState=(
    CONTOURS_ON_DEF, ))
    odb = session.odbs['C:/Temp/gill.odb']
    session.writeFieldReport(fileName='saída.rpt', append=ON,
        sortItem='Element Label', odb=odb, step=0, frame=1,
        outputPosition=INTEGRATION_POINT, variable=(( 'LE',
INTEGRATION_POINT, (
        (COMPONENT, 'LE11'), (COMPONENT, 'LE22'), (COMPONENT,
'LE33'), (
        COMPONENT, 'LE12'), (COMPONENT, 'LE13'), (COMPONENT,
'LE23'), )), ('S',
        INTEGRATION_POINT, ((INVARIANT, 'Mises'), (COMPONENT,
'S11'), (
        COMPONENT, 'S22'), (COMPONENT, 'S33'), (COMPONENT,
'S12'), (COMPONENT,
        'S13'), (COMPONENT, 'S23'), )), ))
    session.writeFieldReport(fileName='saída.rpt', append=ON,
        sortItem='Element Label', odb=odb, step=0, frame=0,

```

```

        outputPosition=WHOLE_ELEMENT, variable=(('EVOL',
        WHOLE_ELEMENT), ))

```

B.2.9 Listagem do arquivo.dat da Macro Abaqus Modificada (arquivo “abaqusMacros”)

```

# Do not delete the following import lines
from abaqus import *
from abaqusConstants import *
# Abertura Macro Geral
def Assembly_cubo():
    import section
    import section
    import regionToolset
    import displayGroupMdbToolset as dgm
    import part
    import material
    import assembly
    import step
    import interaction
    import load
    import mesh
    import job
    import sketch
    import visualization
    import xyPlot
    import displayGroupOdbToolset as dgo
    import connectorBehavior
# Cria o cubo_macico
    s = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
        sheetSize=1000.0)
        g, v, d, c = s.geometry, s.vertices, s.dimensions,
s.constraints
        s.setPrimaryObject(option=STANDALONE)
        s.rectangle(point1=(0.0, 0.0), point2=(
3.47293, 3.47293))
        p = mdb.models['Model-1'].Part(name='cubo_macico',
dimensionality=THREE_D,
        type=DEFORMABLE_BODY)
        p = mdb.models['Model-1'].parts['cubo_macico']
        p.BaseSolidExtrude(sketch=s, depth=
3.47293)
        s.unsetPrimaryObject()
        p = mdb.models['Model-1'].parts['cubo_macico']
        session.viewports['Viewport:
1'].setValues(displayedObject=p)
        del mdb.models['Model-1'].sketches['__profile__']
# Cria o nodulo

```

```

s1 = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
    sheetSize=1000.0)
    g, v, d, c = s1.geometry, s1.vertices, s1.dimensions,
s1.constraints
    s1.setPrimaryObject(option=STANDALONE)
    s1.ConstructionLine(point1=(0.0, -500.0), point2=(0.0,
500.0))
    s1.FixedConstraint(entity=g[2])
    s1.Spline(points=((
-6.12323e-017,-1),(
-0.173648,-0.984808),(
-0.34202,-0.939693),(
-0.5,-0.866025),(
-0.642788,-0.766044),(
-0.766044,-0.642788),(
-0.866025,-0.5),(
-0.939693,-0.34202),(
-0.984808,-0.173648),(
-1,0),(
-0.984808,0.173648),(
-0.939693,0.34202),(
-0.866025,0.5),(
-0.766044,0.642788),(
-0.642788,0.766044),(
-0.5,0.866025),(
-0.34202,0.939693),(
-0.173648,0.984808),(
-6.12323e-017,1)))
    s1.Line(point1=(0.0,-1), point2=(0.0,1))
    s1.VerticalConstraint(entity=g[4])
    p = mdb.models['Model-1'].Part(name='nodulo',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
    p = mdb.models['Model-1'].parts['nodulo']
    p.BaseSolidRevolve(sketch=s1, angle=360.0,
flipRevolveDirection=OFF)
    s1.unsetPrimaryObject()
    p = mdb.models['Model-1'].parts['nodulo']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Assembly cubo_macico
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['cubo_macico']
    a1.Instance(name='cubo_macico-1', part=p, dependent=OFF)
# Insere os nodulos no cubo Distribuicao aleatoria
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-1', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly

```

```

a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-50.0135)
a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(1,0,0), angle=-162.705)
a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-112.686)
a1.translate(instanceList=('nodulo-1', ),
vector=(2.43926,1.2038,2.39437))

a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-2', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-28.2452)
a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(1,0,0), angle=-63.2696)
a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-124.23)
a1.translate(instanceList=('nodulo-2', ),
vector=(1.17148,2.3115,1.0641))

# Cria os vazios dos nodulos no cubo( cria a part cubo_vazado)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].view.fitView()
a1 = mdb.models['Model-1'].rootAssembly
a1.PartFromBooleanCut(name='cubo_vazado',
instanceToBeCut=mdb.models['Model-
1'].rootAssembly.instances['cubo_macico-1'],
cuttingInstances=(
a1.instances['nodulo-1'],
a1.instances['nodulo-2'],
))
# cria a instance cubo_vazado-1 no Assembly)
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['cubo_vazado']
a1.Instance(name='cubo_vazado-1', part=p, dependent=OFF)
# Inserindo os nodulos nos vazios do cubo vazado
# Contato: tipo TIE
session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=ON,
interactions=ON, constraints=ON, connectors=ON,
engineeringFeatures=ON,
adaptiveMeshConstraints=OFF)
session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
'cubo_vazado-1', ))
session.viewports['Viewport: 1'].view.fitView()

```

```

    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'nodulo-1',
        'nodulo-2'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['cubo_vazado-1'].faces
    sidelFaces1 = s1
    region1=regionToolset.Region(sidelFaces=sidelFaces1)
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['nodulo-1'].faces
    sidelFaces1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
    s2 = a.instances['nodulo-2'].faces
    sidelFaces2 = s2.getSequenceFromMask(mask=('[#1 ]', ), )
    region2=regionToolset.Region(sidelFaces=
        sidelFaces1+\
        sidelFaces2)
    mdb.models['Model-1'].Tie(name='tie', master=region1,
slave=region2,
        positionToleranceMethod=COMPUTED, adjust=ON,
tieRotations=ON,
        thickness=ON)
# Ajuste na tela
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1',
        'nodulo-1',
        'nodulo-2'))
    session.viewports['Viewport: 1'].view.fitView()
# Criacao da malha 1 (14x14x7):
# Malha do cubo_vazado-1:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1', ))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    pickedRegions = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=STANDARD)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=STANDARD)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=STANDARD,

```

```

        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = (cells1, )
    a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
            elemType3))
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (a.instances['cubo_vazado-1'], )
# size=Lcubo2nodulos/7
    a.seedPartInstance(regions=partInstances, size=0.3003,
deviationFactor=0.1)
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (a.instances['cubo_vazado-1'], )
    a.generateMesh(regions=partInstances)
# Malha dos nodulos:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'nodulo-1',
        'nodulo-2'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = cells1+cells2
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=STANDARD)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=STANDARD)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=STANDARD,
        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = ((cells1+cells2), )
    a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
            elemType3))
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (
        a.instances['nodulo-1'],
        a.instances['nodulo-2'], )
# size=minimo

```

```

    a.seedPartInstance(regions=partInstances, size=1.5,
deviationFactor=0.1)
    a = mdb.models['Model-1'].rootAssembly
    partInstances =(
        a.instances['nodulo-1'],
        a.instances['nodulo-2'], )
    a.generateMesh(regions=partInstances)
# Ativa a malha do cubo_vazado-1:
    a = mdb.models['Model-1'].rootAssembly
    partInstances =(a.instances['cubo_vazado-1'], )
    a.generateMesh(regions=partInstances)
# Ajuste na tela
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.regenerate()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
# Ativa nodulos e cubo_vazado-1
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF,
visibleInstances=('cubo_vazado-1',
        'nodulo-1',
        'nodulo-2'))
# Supress cubo macico
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a = mdb.models['Model-1'].rootAssembly
    a.features['cubo_macico-1'].suppress()
# Cria materiais PMMA e Borracha
    session.viewports['Viewport:
1'].partDisplay.setValues(sectionAssignments=ON,
        engineeringFeatures=ON)
# Cubo PMMA elastico (E[N/micrometro^2])
    mdb.models['Model-1'].Material(name='PMMA')
    mdb.models['Model-
1'].materials['PMMA'].Elastic(table=((0.00324, 0.25), ))
# Nodulos borracha elastica
    mdb.models['Model-1'].Material(name='borracha')
    mdb.models['Model-
1'].materials['borracha'].Elastic(table=((4e-005, 0.4), ))
# Cria secoes cubo e nodulo
    mdb.models['Model-1'].HomogeneousSolidSection(name='cubo',
material='PMMA',
        thickness=1.0)

```



```

        mdb.models['Model-1'].HomogeneousSolidSection(name='nodulo',
            material='borracha', thickness=1.0)
# Assigna secao nodulo a nodulo
    p = mdb.models['Model-1'].parts['nodulo']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['nodulo']
    p.SectionAssignment(region=region, sectionName='nodulo',
offset=0.0)
# Assigna secao cubo a cubo vazado
    p = mdb.models['Model-1'].parts['cubo_vazado']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    p.SectionAssignment(region=region, sectionName='cubo',
offset=0.0)
# Cria o Step
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(loads=OFF, bcs=OFF,
        predefinedFields=OFF, connectors=OFF,
adaptiveMeshConstraints=ON)
    mdb.models['Model-1'].StaticStep(name='final',
previous='Initial')
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(step='final')
# Inclui efeito nao linear de grandes deformacoes e
deslocamentos no step final para rodar hiperelastico
    a = mdb.models['Model-1'].rootAssembly
    a.regenerate()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=OFF)
    session.viewports['Viewport:
1'].assemblyDisplay.meshOptions.setValues(
        meshTechnique=OFF)
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    mdb.models['Model-1'].steps['final'].setValues(nlgeom=ON)
# Cria BC
# Deslocamento prescrito em y
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].view.setValues(session.views['Top'])

```

```

mdb.models['Model-1'].ExpressionField(name='AnalyticalField-1', localCsys=None,
description='', expression='0.001*(X+Y+Z)')
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#2 ]', ), )
region = regionToolset.Region(faces=faces1)
mdb.models['Model-1'].DisplacementBC(name='BC-3',
createStepName='final',
region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
amplitude=UNSET, fixed=OFF, distributionType=FIELD,
fieldName='AnalyticalField-1', localCsys=None)
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#4 ]', ), )
region = regionToolset.Region(faces=faces1)
mdb.models['Model-1'].DisplacementBC(name='BC-7',
createStepName='final',
region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
amplitude=UNSET, fixed=OFF, distributionType=FIELD,
fieldName='AnalyticalField-1', localCsys=None)
session.viewports['Viewport:
1'].view.setValues(session.views['Bottom'])
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#8 ]', ), )
region = regionToolset.Region(faces=faces1)
mdb.models['Model-1'].DisplacementBC(name='BC-4',
createStepName='final',
region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
amplitude=UNSET, fixed=OFF, distributionType=FIELD,
fieldName='AnalyticalField-1', localCsys=None)
session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#10 ]', ), )
region = regionToolset.Region(faces=faces1)
mdb.models['Model-1'].DisplacementBC(name='BC-5',
createStepName='final',
region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
amplitude=UNSET, fixed=OFF, distributionType=FIELD,
fieldName='AnalyticalField-1', localCsys=None)
a = mdb.models['Model-1'].rootAssembly
f1 = a.instances['cubo_vazado-1'].faces
faces1 = f1.getSequenceFromMask(mask=('[#20 ]', ), )
region = regionToolset.Region(faces=faces1)

```

```

        mdb.models['Model-1'].DisplacementBC(name='BC-6',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
        a = mdb.models['Model-1'].rootAssembly
        f1 = a.instances['cubo_vazado-1'].faces
        faces1 = f1.getSequenceFromMask(mask=('[#1 ]', ), )
        region = regionToolset.Region(faces=faces1)
        mdb.models['Model-1'].DisplacementBC(name='BC-8',
createStepName='final',
        region=region, u1=1.0, u2=1.0, u3=1.0, ur1=UNSET,
ur2=UNSET, ur3=UNSET,
        amplitude=UNSET, fixed=OFF, distributionType=FIELD,
        fieldName='AnalyticalField-1', localCsys=None)
# Tela branca

session.graphicsOptions.setValues(backgroundColor='#FFFFFF')
# Ajustes na tela
        session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
        session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
# opcoes .odb
        mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(variables=(
        'S', 'PE', 'PEEQ', 'PEMAG', 'LE', 'U', 'RF', 'CF',
'CSTRESS', 'CDISP',
        'EVOL'))
# RUN
        mdb.Job(name='gill1', model='Model-1', type=ANALYSIS,
explicitPrecision=SINGLE,
        nodalOutputPrecision=SINGLE, description='',
        parallelizationMethodExplicit=DOMAIN,
multiprocessingMode=DEFAULT,
        numDomains=1, userSubroutine='', numCpus=1,
preMemory=1536.0,
        standardMemory=1536.0, standardMemoryPolicy=MODERATE,
scratch='',
        echoPrint=OFF, modelPrint=OFF, contactPrint=OFF,
historyPrint=OFF)
        mdb.jobs['gill1'].submit(consistencyChecking=OFF)
        mdb.jobs['gill1'].waitForCompletion()
# saida de tensoes e deformacoes e inp
        o3 = session.openOdb(name='C:/Temp/gill1.odb')
        session.viewports['Viewport:
1'].setValues(displayedObject=o3)
        session.viewports['Viewport:
1'].odbDisplay.display.setValues(plotState=(
        CONTOURS_ON_DEF, ))

```

```

odb = session.odbs['C:/Temp/gill1.odb']
session.writeFieldReport(fileName='saída.rpt', append=ON,
    sortItem='Element Label', odb=odb, step=0, frame=1,
    outputPosition=INTEGRATION_POINT, variable=(('LE',
INTEGRATION_POINT, (
    (COMPONENT, 'LE11'), (COMPONENT, 'LE22'), (COMPONENT,
'LE33'), (
    COMPONENT, 'LE12'), (COMPONENT, 'LE13'), (COMPONENT,
'LE23'), )), ('S',
    INTEGRATION_POINT, ((INVARIANT, 'Mises'), (COMPONENT,
'S11'), (
    COMPONENT, 'S22'), (COMPONENT, 'S33'), (COMPONENT,
'S12'), (COMPONENT,
    'S13'), (COMPONENT, 'S23'), )), ))
session.writeFieldReport(fileName='saída.rpt', append=ON,
    sortItem='Element Label', odb=odb, step=0, frame=0,
outputPosition=WHOLE_ELEMENT, variable=(('EVOL',
WHOLE_ELEMENT), ))
cliCommand("""elemArr = mdb.models['Model-
1'].rootAssembly.instances['nodulo-1'].elements""")
cliCommand("""len(elemArr)""")
cliCommand("""elemArr = mdb.models['Model-
1'].rootAssembly.instances['cubo_vazado-1'].elements""")
cliCommand("""len(elemArr)""")
cliCommand("""nodeArr = mdb.models['Model-
1'].rootAssembly.instances['nodulo-1'].nodes""")
cliCommand("""len(nodeArr)""")
cliCommand("""nodeArr = mdb.models['Model-
1'].rootAssembly.instances['cubo_vazado-1'].nodes""")
cliCommand("""len(nodeArr)""")

```

APÊNDICE C – Descrição do algoritmo utilizado para fazer o estudo de elementos finitos do caso elástico não-linear

No presente apêndice se apresenta a descrição detalhada dos programas desenvolvidos e utilizados na aplicação elástica não-linear em elementos finitos:

C.1 Descrição do algoritmo para a homogeneização elástica não-linear

Nesta implementação foi utilizado o software de elementos finitos Abaqus/Explicit através de análise elástica não-linear, para proceder a aplicação da técnica de homogeneização apresentada por Temizer e Zohdi, 2007, em que o comportamento do material é caracterizado por um estado de tensão-deformação e também pelo conceito de *material map*. O material estudado foi uma matriz cúbica isotrópica com uma segunda fase formada por partículas esféricas hiperelásticas com distribuição aleatória. Nesse caso, também devido ao fato de serem analisadas várias amostras de cada modelo, foi necessário automatizar o processo através da elaboração de um algoritmo, no caso, em Matlab. Este algoritmo é esquematizado através do diagrama de blocos da Figura C1 e descrito em detalhes posteriormente.

As variáveis utilizadas no diagrama da Figura C1 são identificadas a seguir:

ipot: parâmetro que vai de 1 até *npot* e permite calcular os diferentes números de nódulos de cada modelo;

npot: potência igual a 6 que permite identificar o modelo de 64 nódulos (tamanho máximo estudado);

inum_nod: parâmetro que identifica o número de nódulos do modelo que está sendo trabalhado;

iamostra: parâmetro que vai de 1 até *namostra* e permite identificar o número da amostra;

namostra: parâmetro que identifica o número máximo de amostras trabalhadas.

jesadot-deformação: Parâmetro que identifica o número de estados de carga executados para cada amostra;

N: número de estados de deformação, apresentado na seção 5.2.2.

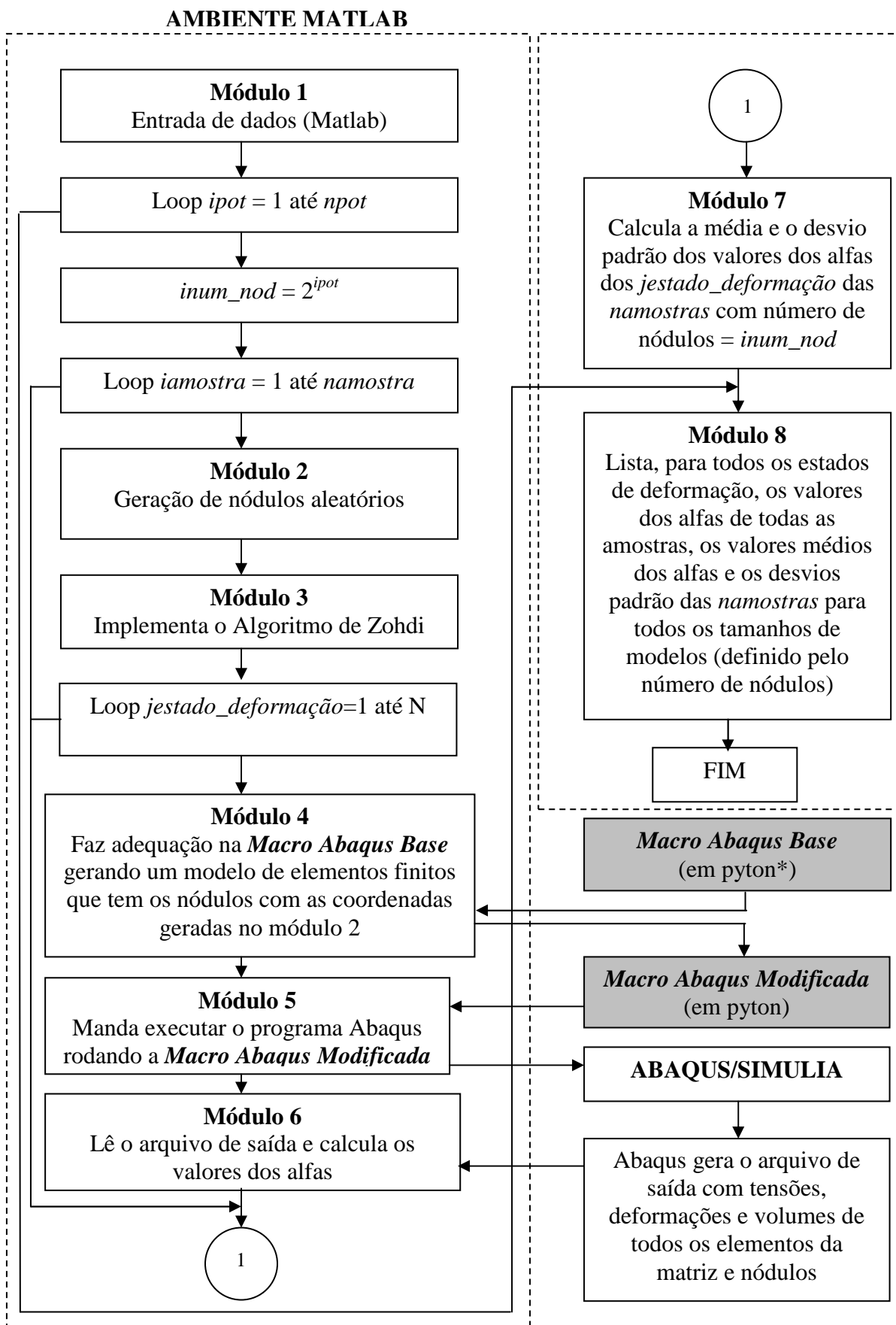


Figura C1 – Diagrama de blocos do algoritmo utilizado para realizar a homogeneização do composto elástico não-linear (estudo descrito na seção 5.2)

* Python é a linguagem de comandos utilizada pelo software Abaqus/Simulia

C.2 Descrição detalhada do diagrama de blocos da Figura C1

Apresenta-se a seguir uma descrição detalhada dos módulos do diagrama de blocos da Figura C1, construídos dentro do ambiente Matlab. Após, nas seções C.2.1 a C.2.8, são apresentadas as listagens dos módulos programadas em Matlab. No, final, nas seções C.2.9 e C.2.10 são apresentadas as listagens da *Macro Abaqus Base* e da *Macro Abaqus Modificada* obtidas em arquivos de extensão *.dat*.

Módulo 1: Este módulo permite a introdução dos dados utilizados no algoritmo. São eles:

- Parâmetros geométricos para geração dos nódulos;
- Fração de volume de heterogeneidades em relação ao volume total da amostra;
- Número máximo e número mínimo de nódulos na amostra;
- Número de amostras para cada tamanho de amostra;
- Módulo de elasticidade longitudinal da matriz e das heterogeneidades;
- Coeficiente de Poisson da matriz e das heterogeneidades;
- Densidade da matriz e das heterogeneidades;
- Número de incrementos (*ninc*);
- Dimensões de matrizes;

A listagem deste módulo está na seção C.2.1.

Módulo 2: Este módulo gera os nódulos esféricos posicionados aleatoriamente no interior da matriz cúbica com distância mínima entre eles e entre eles e as bordas da matriz previamente estabelecidas para cada amostra. A listagem deste módulo está na seção C.2.2.

Módulo 3: Este módulo implementa o algoritmo de Zohdi para cada estado de deformação aplicado a cada amostra de cada tamanho de modelo. Para isso, a seguinte sequência é apresentada:

- Estabelece os limites dos estiramentos do modelo $\lambda_F^i \in [\lambda_F^{\min}, \lambda_F^{\max}]$;
- Estabelece as defasagens *a* e *b* e os valores dos incrementos λ_F^{inc} de λ_F^i ;
- Para cada estado de deformação (através do comando *while*):

- Monta a matriz $F = \begin{pmatrix} \lambda_F^1 & & \\ & \lambda_F^2 & \\ & & \lambda_F^3 \end{pmatrix}$ de estiramentos;

- Calcula as condições de contorno a serem aplicadas no modelo, na forma:

$$CC = \begin{pmatrix} \frac{L_0(\lambda_F^1 - 1)}{2} & & \\ & \frac{L_0(\lambda_F^2 - 1)}{2} & \\ & & \frac{L_0(\lambda_F^3 - 1)}{2} \end{pmatrix}$$

- Calcula a matriz inversa de F;

- Calcula a matriz transposta de F;

- Calcula a matriz de deformações

$$\lambda_E = \begin{pmatrix} \frac{[(\lambda_F^1)^2 - 1]}{2} & & \\ & \frac{[(\lambda_F^2)^2 - 1]}{2} & \\ & & \frac{[(\lambda_F^3)^2 - 1]}{2} \end{pmatrix};$$

- Monta a matriz

$$EE = \begin{pmatrix} 1 & \lambda_E^1 & (\lambda_E^1)^2 \\ 1 & \lambda_E^2 & (\lambda_E^2)^2 \\ 1 & \lambda_E^3 & (\lambda_E^3)^2 \end{pmatrix};$$

- Calcula $J = \det(F)$;

- Calcula a matriz inversa de EE;

A listagem deste módulo está na seção C.2.3.

Módulo 4: Este módulo faz a adequação da *Macro Abaqus Base* gerando um modelo de elementos finitos que tem os nódulos com as coordenadas geradas no módulo 2 (*Macro Abaqus Modificada*), através da sub-rotina chamada “*imprime_py*”. A listagem deste módulo está na seção C.2.4.

Módulo 5: Este módulo manda executar o programa Abaqus rodando a *Macro Abaqus Modificada* gerada no módulo 4. A listagem deste módulo está na seção C.2.5.

Módulo 6: Este módulo lê o arquivo de saída com tensões, deformações e volumes de todos os elementos da matriz e dos nódulos gerado no módulo 5 e calcula os valores dos coeficientes alfas do composto, através da sub-rotina “*calcula_alfas*”.

A seguir é descrito de forma detalhada os passos da sub-rotina “*calcula_alfas*”:

- Lê o arquivo “*saída.rpt*” com as tensões, deformações e volumes de todos os elementos da matriz e da heterogeneidades;
- Calcula as tensões de Piola-Kirchhoff 2 a partir das tensões de Cauchy através da expressão (Equação 2.56)

$$\mathbf{T}_2 = \mathbf{PK}^{2esp} = \det(\mathbf{F})\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-T}$$

- Calcula os coeficientes α_0 , α_1 e α_2 através da expressão

$$\begin{pmatrix} 1 & \lambda_E^1 & (\lambda_E^1)^2 \\ 1 & \lambda_E^2 & (\lambda_E^2)^2 \\ 1 & \lambda_E^3 & (\lambda_E^3)^2 \end{pmatrix} \begin{Bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{Bmatrix} = \begin{Bmatrix} \lambda_S^1 \\ \lambda_S^2 \\ \lambda_S^3 \end{Bmatrix}$$

A listagem deste módulo está na seção C.2.6.

Módulo 7: Este módulo calcula a média e o desvio padrão dos valores dos alfas das *namostras* com número de nódulos = *inum_nod*, para todos os estados de deformação aplicados. A listagem deste módulo está na seção C.2.7.

Módulo 8: Neste módulo ocorre a listagem, para todos os estados de deformação, dos valores dos alfas de todas as amostras, os valores médios dos alfas e os desvios padrão das *namostras* para todos os tamanhos de modelos (definido pelo número de nódulos). A listagem deste módulo está na seção C.2.8.

C.2.1 Listagem do Módulo 1 programado em Matlab

```
a=1
b=a
c=1
n1=1
n2=1
Densidade=0.10
MaxNodulos=32
amostras=4
Nodulos=32
E1=3240.0e-06;
poisson1=0.25;
```

```

E2=40.0e-06;
poisson2=0.49;
densidade1=1200e-24;      %PMMA
densidade2=950e-24;      %Elastômero
ninc=3;
Nnoddif=1;
alpha=zeros(3,(1+ninc)^3,amostras,Nnoddif);
LambE=zeros(3,(1+ninc)^3,Nnoddif);
mediaalpha=zeros(3,(1+ninc)^3,Nnoddif);
somaalpha=zeros(3,(1+ninc)^3,Nnoddif);
desvioalpha=zeros(3,(1+ninc)^3,Nnoddif);

```

C.2.2 Listagem do Módulo 2 programado em Matlab

```

cant1=18;          %quantidade de pontos por valor de z
cant2=18;          %quantidade de isocurvas z

maximo=max(max(a,b),c)

X=ones(cant2,cant1);
Y=ones(cant2,cant1);
Z=ones(cant2,cant1);
XX=ones(cant2+1,cant1+1,Nodulos);
YY=ones(cant2+1,cant1+1,Nodulos);
ZZ=ones(cant2+1,cant1+1,Nodulos);

i=1;
while i<=cant2+1;
    ii=(i-1)/cant2*pi-pi/2;
    j=1;
    while j<=cant1+1;
        jj=(j-1)/cant1*2*pi-pi;
        X(i,j)=a*abs(cos(ii)).^(n1).*abs(cos(jj)).^(n2)*sign(cos(ii))*
        sign(cos(jj));
        Z(i,j)=b*abs(cos(ii)).^(n1).*abs(sin(jj)).^(n2)*sign(cos(ii))*
        sign(sin(jj));
        Y(i,j)=c*abs(sin(ii)).^(n1)*sign(sin(ii));
        j=j+1;
    end
    i=i+1;
end

% Calculo do volume dos nódulos e do cubo obedecendo a
% densidade de volume estabelecida

r=2/n2;
t=2/n1;
F = @(x)sin(x).^(2/r-1).*cos(x).^(2/r-1);
Beta1=2*quad(F,0,pi/2);

```

```

F = @(x)sin(x).^(4/t-1).*cos(x).^(2/t-1);
Beta2=2*quad(F,0,pi/2);
Vnodulo=2/3*a*b*c^4/(r*t)*Beta1*Beta2;

while Nodulos<=MaxNodulos

Vnodulos=Nodulos*Vnodulo;    % Volume de n nodulos
Vcubo=Vnodulos/Densidade;    % Volume da amostra para a
densidade desejada
Densidaderelativa=Nodulos*4/3*pi*maximo^3/Vcubo
% Limites do ambiente onde gerar os centros dos nódulos
Xcub0=0.;
Xcub1=(Vcubo)^(1/3);        % Dimensão do cubo
Ycub0=0.;
Ycub1=(Vcubo)^(1/3);        % Dimensão do cubo
Zcub0=0.;
Zcub1=(Vcubo)^(1/3);        % Dimensão do cubo
Lx=Xcub1-Xcub0;
Ly=Ycub1-Ycub0;
Lz=Zcub1-Zcub0;
    if maximo>=Lx/2
        maximo
        Lx/2
    fprintf('no entra en el cubo!!!\n');
    off
end

rr=1
while rr<=amostras

% -----Gera os nódulos-----

i=1;
while i<=Nodulos
lim_iter(i)=350;
i=i+1;
end

ntent=150;

kk=1;
while kk<ntent
    k=1;
    while k<=Nodulos
        d=ones(cant2+1,cant1+1,Nodulos);
        D=zeros(cant2+1,cant1+1,Nodulos);
        iter=0;

        while any(any(any(d>D)))>0.5;

```

```

        iter=iter+1;           % para saber quantas vezes uma
configuração boa
        D=ones(cant2+1,cant1+1,Nodulos);

% ----- Gira e movimenta a hiperelipsóide -----

cos1(k)=rand(1,1);
cos2(k)=rand(1,1);
cos3(k)=rand(1,1);

Ca=2*cos1(k)-1;
Cb=2*cos2(k)-1;
Cg=2*cos3(k)-1;

alfa(k)=acos(Ca)*180/pi;
beta(k)=acos(Cb)*180/pi;
gama(k)=acos(Cg)*180/pi;

Sa=(1-Ca^2)^(1/2);
Sb=(1-Cb^2)^(1/2);
Sg=(1-Cg^2)^(1/2);
    Rx1=Ca*Cg-Sa*Cb*Sg;
Ry1=Sa*Cg+Ca*Cb*Sg;
Rz1=Sb*Sg;
Rx2=-Ca*Sg-Sa*Cb*Cg;
Ry2=-Sa*Sg+Ca*Cb*Cg;
Rz2=Sb*Cg;
Rx3=Sb*Sa;
Ry3=-Sb*Ca;
Rz3=Cb;

X1=X*Rx1+Y*Ry1+Z*Rz1;
Y1=X*Rx2+Y*Ry2+Z*Rz2;
Z1=X*Rx3+Y*Ry3+Z*Rz3;

% -----Move-se a coordenada do centro do Hiperelipsóide-----

XX(:, :, k)=X1;
YY(:, :, k)=Y1;
ZZ(:, :, k)=Z1;
coef=0.01; % Coeficiente para afastar os nódulos das bordas
do cubo.
           % coef=0.01 (significa 1% do Raio de afastamento
entre a
           % superficie do nódulo e a parede da matriz
cúbica)

    while or(min(min(XX(:, :, k)))<=coef*Lx ,
max(max(XX(:, :, k)))>=Lx*(1-coef))
        x(k)=rand(1,1)*Lx;

```

```

        XX(:, :, k) = X1 + x(k);
    end
    while or(min(min(YY(:, :, k))) <= coef * Ly ,
max(max(YY(:, :, k))) >= Ly * (1 - coef))
        y(k) = rand(1, 1) * Ly;
        YY(:, :, k) = Y1 + y(k);
    end
    while or(min(min(ZZ(:, :, k))) <= coef * Lz ,
max(max(ZZ(:, :, k))) >= Lz * (1 - coef))
        z(k) = rand(1, 1) * Lz;
        ZZ(:, :, k) = Z1 + z(k);
    end

if k > 1
% ----- Calculo de distâncias -----

    j = 1;
    while j < k
        sep = 0.96; %Coeficiente para afastar os nódulos entre si
        % distancia^2 entre o centro j a cada ponto da superf. K

        D(:, :, j) = sep * ((XX(:, :, k) - x(j)).^2 + (YY(:, :, k) -
y(j)).^2 + (ZZ(:, :, k) - z(j)).^2);
        i = 1;
        while i <= cant2 + 1;
            ii = 1;
            while ii <= cant1 + 1;
% matriz com produto interno
d(i, ii, j) = max(max((XX(:, :, j) - x(j)) * (XX(i, ii, k) -
x(j)) + (YY(:, :, j) - y(j)) * (YY(i, ii, k) - y(j)) + (ZZ(:, :, j) -
z(j)) * (ZZ(i, ii, k) - z(j))));

            ii = ii + 1;
        end
        i = i + 1;
    end
    j = j + 1;
end
end

    if iter > lim_iter(k);
        D = ones(cant2 + 1, cant1 + 1, Nodulos);
        d = ones(cant2 + 1, cant1 + 1, Nodulos);
    end

    end

    lim = lim_iter(k);
    if iter > lim;
        k = Nodulos + 1;
    end
    k = k + 1;

```

```

end

% critério de saída
kk;
if iter<=lim;
kk=ntent+1;
else
kk=kk+1;
end
end

if iter<=lim ;
fprintf('configuracao boa:\n');
else
fprintf('configuracao ruim:\n');
off
end

figura2(XX,YY,ZZ,Nodulos,Lx);

minimo=min(min(a,b),c);

```

C.2.3 Listagem do Módulo 3 programado em Matlab

```

%-----Implementação do algoritmo de ZOHD-----

L0=Lx;
Lambf1_min=0.80;

Lf_a=0.045;           % defasagem a
Lf_b=0.045;           % defasagem b

Lambf2_min=Lambf1_min+Lf_a;
Lambf3_min=Lambf1_min+Lf_a+Lf_b;

Lambf1_inc=0.10;      %incrementos de deformação
Lambf2_inc=0.10;
Lambf3_inc=0.10;

Lambf1_max=(Lambf1_min)+(ninc*Lambf1_inc);
Lambf2_max=(Lambf2_min)+(ninc*Lambf2_inc);
Lambf3_max=(Lambf3_min)+(ninc*Lambf3_inc);

Lambf1=Lambf1_min;
Lambf2=Lambf1_min+Lf_a;
Lambf3=Lambf1_min+Lf_a+Lf_b;
aa=Lambf1;

```

```

bb=Lambf2;
cc=Lambf3;
inc=0;

i1=1;
i2=1;
i3=1;
ia=0;

while (aa)<=(Lambf1_max)
    while (bb)<=(Lambf2_max)
        while (cc)<=(Lambf3_max)

            F=[aa,0,0;0,bb,0;0,0,cc]           % matriz F
            bcaa=(L0*(aa-1))/2;
            bccb=(L0*(bb-1))/2;
            bccc=(L0*(cc-1))/2;
            BC=[bcaa,0,0;0,bccb,0;0,0,bccc] % matriz com as
                                                condições de
                                                %contorno nas três direções
            Fi=inv(F);                        % Inversa de F
            Fti=Fi';                          % Transposta da inversa de F
            LambE1=0.5*((aa*aa)-1);
            LambE2=0.5*((bb*bb)-1);
            LambE3=0.5*((cc*cc)-1);
            LambE123=[LambE1,LambE2,LambE3] % matriz com as
deformações

EE=[1,LambE1,LambE1*LambE1;1,LambE2,LambE2*LambE2;1,LambE3,LambE3*LambE3];
    J=det(F);
    Ei=inv(EE);                               % Inversa de E

%----- ORGANIZA A MACRO PARA RODAR NO ABAQUS -----

imprime_py(X,Y,Z,Lx,x,y,z,alfa,beta,gama,Nodulos,cant2,minimo,
E1,E2,poisson1,poisson2,densidade1,densidade2,aa,bb,cc,bcaa,bc
bb,bccc,BC);

%-----RODA A MACRO NO ABAQUS-----

istatus = dos('abaqus cae noGUI="abaqus.py"');

%CALCULA AS TENSÕES E RESOLVE O SISTEMA PARA ACHAR OS ALPHAS

    ia=ia+1;
    [alpha]=calcula_alphas(alpha,Nodulos,rr,Ei,J,Fi,Fti,Lx,ia,
contador);
    LambE(1,ia,contador)=LambE123(1);
    LambE(2,ia,contador)=LambE123(2);

```

```

LambE(3,ia,contador)=LambE123(3);
cc=cc+Lambf3_inc;
i3=i3+1;

end
bb=bb+Lambf2_inc;
cc=Lambf3;
i2=i2+1;

end
aa=aa+Lambf1_inc;
bb=Lambf2;
cc=Lambf3;
i1=i1+1;
end

```

C.2.4 Listagem do Módulo 4 programado em Matlab

```
%----- ORGANIZA A MACRO PARA RODAR NO ABAQUS -----
```

```

imprime_py(X,Y,Z,Lx,x,y,z,alfa,beta,gama,Nodulos,cant2,minimo,
E1,E2,poisson1,poisson2,densidade1,densidade2,aa,bb,cc,bcaa,bc
bb,bccc,BC);

```

Sub-rotina *imprime_py*:

```

function
imprime_py(X,Y,Z,Lx,x,y,z,alfa,beta,gama,Nodulos,cant2,minimo,
E1,E2,poisson1,poisson2,densidade1,densidade2,aa,bb,cc,bcaa,bc
bb,bccc,BC)

arquivo1='Copia_abaqusMacros.py';
arquivo2='abaqusMacros.py';
A=importdata(arquivo1,'\n');
fid2 = fopen(arquivo2,'w');

i=1;
while i<=28;
    fprintf(fid2,'%s\n',A{i});
    i=i+1;
end
fprintf(fid2,'%g',Lx);
fprintf(fid2,',');
fprintf(fid2,'%g',Lx);
fprintf(fid2,')\n');

i=30;
while i<=33;
    fprintf(fid2,'%s\n',A{i});
    i=i+1;
end

```



```

fprintf(fid2, '%g', Lx);
fprintf(fid2, ')\n');

i=35;
while i<=46;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
% coordenadas dos pontos
j=1;
while j<=cant2+1;
    fprintf(fid2, '%g', X(j,1));
    fprintf(fid2, ', ');
    fprintf(fid2, '%g', Y(j,1));
    if j<cant2+1;
        fprintf(fid2, '), (\n');
    else
        fprintf(fid2, '))\n');
    end
    j=j+1;
end

fprintf(fid2, '    s1.Line(point1=(0.0, ');
fprintf(fid2, '%g', Y(1,1));
fprintf(fid2, '), point2=(0.0, ');
fprintf(fid2, '%g', Y(cant2+1,1));
fprintf(fid2, '))\n');

i=89;
while i<=102;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

% começa a copiar a quantidade de nódulos que se tem

k=1;
while k<=Nodulos;
    fprintf(fid2, '    a1 = mdb.models[''Model-
1''].rootAssembly\n');
    fprintf(fid2, '    p = mdb.models[''Model-
1''].parts[''nodulo'']\n');

    fprintf(fid2, '    a1.Instance(name='''nodulo-'');
    fprintf(fid2, '%u''', k);
    fprintf(fid2, ', part=p, dependent=OFF)\n');

    fprintf(fid2, '    a1 = mdb.models[''Model-
1''].rootAssembly\n');

    fprintf(fid2, '    a1.rotate(instanceList=(''nodulo-'');

```

```

fprintf(fid2, '%u' ,k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');
fprintf(fid2, '      axisDirection=(0,0,1), angle=');
fprintf(fid2, '%g' ,-alfa(k));
fprintf(fid2, ')\n');

fprintf(fid2, '      a1.rotate(instanceList=('nodulo-');
fprintf(fid2, '%u' ,k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');
fprintf(fid2, '      axisDirection=(1,0,0), angle=');
fprintf(fid2, '%g' ,-beta(k));
fprintf(fid2, ')\n');

fprintf(fid2, '      a1.rotate(instanceList=('nodulo-');
fprintf(fid2, '%u' ,k);
fprintf(fid2, ', ), axisPoint=(0,0,0),\n');
fprintf(fid2, '      axisDirection=(0,0,1), angle=');
fprintf(fid2, '%g' ,-gama(k));
fprintf(fid2, ')\n');

fprintf(fid2, '      a1.translate(instanceList=('nodulo-');
fprintf(fid2, '%u' ,k);
fprintf(fid2, ', ), vector=');
fprintf(fid2, '%g,%g,%g' ,x(k),y(k),z(k));
fprintf(fid2, '))\n\n');
k=k+1;
end

i=164;
while i<=171;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
fprintf(fid2, '      a1.instances['nodulo-');
fprintf(fid2, '%u' ],\n',k);
k=k+1;
end

i=180;
while i<=196;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '      ('nodulo-%u'))\n',k);

```

```

else
    fprintf(fid2, '          'nodulo-%u', \n', k);
end
k=k+1;
end

fprintf(fid2, '    session.viewports[''Viewport:
1''].view.fitView()\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');
fprintf(fid2, '    s1 = a.instances[''cubo_vazado-
1''].faces\n');
fprintf(fid2, '    sidelFaces1 = s1\n');
fprintf(fid2, '
region1=regionToolset.Region(sidelFaces=sidelFaces1)\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');

k=1;
while k<=Nodulos;
    fprintf(fid2, '    s%u = a.instances[''nodulo-
%u''].faces\n', k, k);
    fprintf(fid2, '    sidelFaces%u =
s%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n', k, k);
    k=k+1;
end

fprintf(fid2, '
region2=regionToolset.Region(sidelFaces=\n');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    sidelFaces%u)\n', k);
else
    fprintf(fid2, '    sidelFaces%u+\\ \n', k);
end
k=k+1;
end

i=236;
while i<=241;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '          'nodulo-%u'')\n', k);
else
    fprintf(fid2, '          'nodulo-%u', \n', k);

```

```

end
k=k+1;
end

i=250;
while i<=272;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

fprintf(fid2, '    a.seedPartInstance(regions=partInstances,
size=%g, deviationFactor=0.1)\n', minimo/(3.33));

i=274;
while i<=278;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    'nodulo-%u')\n', k);
else
    fprintf(fid2, '    'nodulo-%u', \n', k);
end
k=k+1;
end

fprintf(fid2, '    session.viewports[''Viewport:
1''].view.fitView()\n');
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');

k=1;
while k<=Nodulos;
    fprintf(fid2, '    c%u = a.instances[''nodulo-
%u''].cells\n', k, k);
    fprintf(fid2, '    cells%u =
c%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n', k, k);
    k=k+1;
end
fprintf(fid2, '    pickedRegions = ');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, 'cells%u\n', k);
else
    fprintf(fid2, 'cells%u+', k);
end
k=k+1;

```

```

end

i=306;
while i<=311;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
while k<=Nodulos;
    fprintf(fid2, '    c%u = a.instances[''nodulo-
%u''].cells\n', k, k);
    fprintf(fid2, '    cells%u =
c%u.getSequenceFromMask(mask=('[#1 ]'', ), )\n', k, k);
    k=k+1;
end
fprintf(fid2, '    pickedRegions = ((');
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, 'cells%u), )\n', k);
else
    fprintf(fid2, 'cells%u+', k);
end
k=k+1;
end

i=329;
while i<=332;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
k=1;
while k<=Nodulos;
if k==Nodulos;
    fprintf(fid2, '    a.instances[''nodulo-%u''], )\n', k);
else
    fprintf(fid2, '    a.instances[''nodulo-%u''], \n', k);
end
k=k+1;
end

fprintf(fid2, '# size=minimo\n');
fprintf(fid2, '    a.seedPartInstance(regions=partInstances,
size=%g, deviationFactor=0.1)\n', minimo*1.5);
fprintf(fid2, '    a = mdb.models[''Model-
1''].rootAssembly\n');
fprintf(fid2, '    partInstances = (\n');
k=1;
while k<=Nodulos;
if k==Nodulos;

```

```

        fprintf(fid2, '          a.instances[''nodulo-%u''], )\n',k);
    else
        fprintf(fid2, '          a.instances[''nodulo-%u''],\n',k);
    end
    k=k+1;
end

i=353;
while i<=369;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

i=370;
while i<=386;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end
k=1;
while k<=Nodulos;
    if k==Nodulos;
        fprintf(fid2, '          ''nodulo-%u'')\n',k);
    else
        fprintf(fid2, '          ''nodulo-%u'',\n',k);
    end
    k=k+1;
end
i=395;
while i<=402;
    fprintf(fid2, '%s\n',A{i});
    i=i+1;
end

% Aqui se pode mudar as propriedades do material

fprintf(fid2, '# Cubo PMMA elastico (E[N/micrometro^2])\n');
fprintf(fid2, '      mdb.models[''Model-
1''].Material(name='''PMMA''')\n');
fprintf(fid2, '      mdb.models[''Model-
1''].materials[''PMMA''].Elastic(table=(%g, %g),
)\n',E1,poisson1);
fprintf(fid2, '      mdb.models[''Model-
1''].materials[''PMMA''].Density(table=(%g, ),
)\n',densidade1);
fprintf(fid2, '# Nodulos borracha elastica\n');
fprintf(fid2, '      mdb.models[''Model-
1''].Material(name='''borracha''')\n');
fprintf(fid2, '      mdb.models[''Model-
1''].materials[''borracha''].Hyperelastic(type=YEOH,\n');
fprintf(fid2, '          volumetricResponse=POISSON_RATIO,
poissonRatio=%g, table=())\n',poisson2);

```

```

i=411;
while i<=447;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
fprintf(fid2, '      mdb.models[''Model-
1''].materials[''borracha''].Density(table=((%g, ),
))\n', densidade2);

i=449;
while i<=475;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
fprintf(fid2, '      smooth=SOLVER_DEFAULT, data=((0.0, 0.0),
(1e-8, %g))\n', bcaa);

i=477;
while i<=477;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
fprintf(fid2, '      smooth=SOLVER_DEFAULT, data=((0.0, 0.0),
(1e-8, %g))\n', bcbb);

i=479;
while i<=479;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end
fprintf(fid2, '      smooth=SOLVER_DEFAULT, data=((0.0, 0.0),
(1e-8, %g))\n', bccc);

i=481;
while i<=591;
    fprintf(fid2, '%s\n', A{i});
    i=i+1;
end

k=1;
fprintf(fid2, '      cliCommand("""elemArr = mdb.models[''Model-
1''].rootAssembly.instances[''nodulo-%u''].elements""")\n', k);
fprintf(fid2, '      cliCommand("""len(elemArr)""")\n');
fprintf(fid2, '      cliCommand("""elemArr = mdb.models[''Model-
1''].rootAssembly.instances[''cubo_vazado-
1''].elements""")\n');
fprintf(fid2, '      cliCommand("""len(elemArr)""")\n');
fprintf(fid2, '      cliCommand("""nodeArr = mdb.models[''Model-
1''].rootAssembly.instances[''nodulo-%u''].nodes""")\n', k);
fprintf(fid2, '      cliCommand("""len(nodeArr)""")\n');

```

```
fprintf(fid2, '      cliCommand("""nodeArr = mdb.models[''Model-1''].rootAssembly.instances[''cubo_vazado-1''].nodes""")\n');
fprintf(fid2, '      cliCommand("""len(nodeArr)""")\n');

fclose(fid2);
```

C.2.5 Listagem do Módulo 5 programado em Matlab

```
%-----Executa no Abaqus a Macro Abaqus Modificada-----

istatus = dos('abaqus cae noGUI="abaqus.py"');
```

C.2.6 Listagem do Módulo 6 programado em Matlab

```
% Sub-rotina calcula_alfas onde se calculam os coeficientes
% alfas

ia=ia+1;
alpha]=calcula_alfas(alpha,Nodulos,rr,Ei,J,Fi,Fti,Lx,ia,contador);
```

Sub-rotina *calcula_alfas*:

```
function
[alpha]=calcula_alfas(alpha,Nodulos,rr,Ei,J,Fi,Fti,Lx,ia,contador);

file='abaqus.rpy';
fid=fopen(file,'r');
C=textscan(fid,'%s\n',4,'HeaderLines',27,'Delimiter',' ','...
            'commentStyle','//');
Elemnodo=C{1,2}(1);
Elemcubo=C{1,2}(2);
Nodosnodo=C{1,2}(3);
Nodoscubo=C{1,2}(4);
fclose(fid);

% Lê o arquivo de tensões, deformações e volumes

file='saída.rpt';
fid=fopen(file,'r');
Rescub=textscan(fid,'%f%f%f%f%f%f%f%f%f%f%f%f',Elemcubo,
                'HeaderLines',19,...
                'commentStyle','//');
k=1;
while k<=Nodulos
```



```

Resnod{k,:}=textscan(fid,'%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f',Elem
nodo,'HeaderLines',19,...
                    'commentStyle','//');
k=k+1;
end

Volcub=textscan(fid,'%f%f',Elemcubo,'HeaderLines',33,...
                'commentStyle','//');

CC=textscan(fid,'%s%n',1,'HeaderLines',8,...
            'commentStyle','//');

VolcubT=CC{1,2}(1);

k=1;
while k<=Nodulos;
Volnod{k,:}=textscan(fid,'%f%f',Elemnodo,'HeaderLines',8,...
                    'commentStyle','//');
CC=textscan(fid,'%s%n',1,'HeaderLines',8,...
            'commentStyle','//');
VolnodT(k)=CC{1,2}(1);

k=k+1;
end

fclose(fid);
delete('C:\Temp\abaqus.rpy')
delete('C:\Temp\saída.rpt')

%Até aqui leu o arquivo de tensões, deformações e volumes

Sigcub1=0;
Sigcub2=0;
Sigcub3=0;
i=1;
while i<=Elemcubo

Sigcub1=Sigcub1+(Rescub{1,10}(i)*Volcub{1,2}(i));
Sigcub2=Sigcub2+(Rescub{1,11}(i)*Volcub{1,2}(i));
Sigcub3=Sigcub3+(Rescub{1,12}(i)*Volcub{1,2}(i));

i=i+1;
end

Signod1=0;
Signod2=0;
Signod3=0;
k=1;
while k<=Nodulos;
    i=1;

```

```

    while i<=Elemnodo;

    Signod1=Signod1+(Resnod{k,1}{1,10}(i)*Volnod{k,1}{1,2}(i));
    Signod2=Signod2+(Resnod{k,1}{1,11}(i)*Volnod{k,1}{1,2}(i));
    Signod3=Signod3+(Resnod{k,1}{1,12}(i)*Volnod{k,1}{1,2}(i));

    i=i+1;
    end
    k=k+1;
end

Voltot=0;
ii=1;
while ii<=Nodulos
    Voltot=Voltot+VolnodT(ii);
    ii=ii+1;
end
Voltot=Voltot+VolcubT;

Sig1=(Sigcub1+Signod1)/Voltot;
Sig2=(Sigcub2+Signod2)/Voltot;
Sig3=(Sigcub3+Signod3)/Voltot;

Sig123=[Sig1,0,0;0,Sig2,0;0,0,Sig3];    %(Tensões de Cauchy)
Xis=J*Fi;
Ze=Sig123*Fti;
S1=Xis*Ze;
S=[S1(1,1),S1(2,2),S1(3,3)];          %Tensões Piola-
Kirchhoff 2

    alpha012=EI*S';

    alpha(1,ia,rr,contador)=alpha012(1);
    alpha(2,ia,rr,contador)=alpha012(2);
    alpha(3,ia,rr,contador)=alpha012(3)

end

```

C.2.7 Listagem do Módulo 7 programado em Matlab

```

rr=rr+1;
end

rr=1;
while rr<=amostras
    tt=1;
    while tt<=((1+ninc)^3)

```

```

        ttt=1;
        while ttt<=3
mediaalpha(ttt,tt,contador)=mediaalpha(ttt,tt,contador)+alpha(
ttt,tt,rr,contador)/amostras;
        ttt=ttt+1;
        end
        tt=tt+1;
end
rr=rr+1;
end

rr=1;
while rr<=amostras
tt=1;
while tt<=((1+ninc)^3)
        ttt=1;
        while ttt<=3
somaalpha(ttt,tt,contador)=somaalpha(ttt,tt,contador)+(alpha(t
tt,tt,rr,contador)-mediaalpha(ttt,tt,contador))^2;
        ttt=ttt+1;
        end
        tt=tt+1;
end
rr=rr+1;
end

tt=1;
while tt<=(1+ninc)^3
        ttt=1;
        while ttt<=3
desvioalpha(ttt,tt,contador)=(somaalpha(ttt,tt,contador)/(amos
tras-1))^(0.5);

        ttt=ttt+1;
        end
        tt=tt+1;
end
end

```

C.2.8 Listagem do Módulo 8 programado em Matlab

```

file='resultados.dat';
fid=fopen(file,'a');
fprintf(fid,'Cantidad de Nódulos: %6.2f\n',Nodulos);
fprintf(fid,'Valores de LambE:\n');

ttt=1;
while ttt<=3
        tt=1;
while tt<=(1+ninc)^3

```

```

fprintf(fid, '%g    ', LambE(ttt, tt, contador));
tt=tt+1;
end
fprintf(fid, '\n');
ttt=ttt+1;
end
rr=1;
while rr<=amostras
fprintf(fid, 'Amostra numero: %6.2f\n', rr);

fprintf(fid, 'alpha:\n');

ttt=1;
while ttt<=3
    tt=1;
    while tt<=(1+ninc)^3
fprintf(fid, '%g    ', alpha(ttt, tt, rr, contador));
tt=tt+1;
    end
    fprintf(fid, '\n');
    ttt=ttt+1;
end
rr=rr+1;
end

fprintf(fid, '\n');
fprintf(fid, 'Valores médios de alpha:\n');
ttt=1;
while ttt<=3
    tt=1;
    while tt<=(1+ninc)^3
fprintf(fid, '%g    ', mediaalpha(ttt, tt, contador));
tt=tt+1;
    end
    fprintf(fid, '\n');
    ttt=ttt+1;
end
fprintf(fid, 'Desvios de alpha:\n');
    ttt=1;
    while ttt<=3
        tt=1;
        while tt<=(1+ninc)^3
fprintf(fid, '%g    ', desvioalpha(ttt, tt, contador));
tt=tt+1;
        end
        fprintf(fid, '\n');
        ttt=ttt+1;
    end

fprintf(fid, '\n');

```

```

fclose(fid);
passo=Nodulos;
Nodulos=Nodulos+passo
end
fprintf('acabou');

```

C.2.9 Listagem do arquivo.dat da Macro Abaqus Base (arquivo “Cópia_abaqusMacros”)

```

# Do not delete the following import lines
from abaqus import *
from abaqusConstants import *
# Abertura Macro Geral
def Assembly_cubo():
    import section
    import section
    import regionToolset
    import displayGroupMdbToolset as dgm
    import part
    import material
    import assembly
    import step
    import interaction
    import load
    import mesh
    import job
    import sketch
    import visualization
    import xyPlot
    import displayGroupOdbToolset as dgo
    import connectorBehavior
# Cria o cubo_macico
    s = mdb.models['Model-1'].ConstrainedSketch(name='__profile__',
        sheetSize=1000.0)
        g, v, d, c = s.geometry, s.vertices, s.dimensions,
s.constraints
        s.setPrimaryObject(option=STANDALONE)
        s.rectangle(point1=(0.0, 0.0), point2=(
53.5, 53.5))
        p = mdb.models['Model-1'].Part(name='cubo_macico',
dimensionality=THREE_D,
        type=DEFORMABLE_BODY)
        p = mdb.models['Model-1'].parts['cubo_macico']
        p.BaseSolidExtrude(sketch=s, depth=
53.5)
        s.unsetPrimaryObject()
        p = mdb.models['Model-1'].parts['cubo_macico']

```

```

    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Cria o nodulo
    s1 = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
        sheetSize=1000.0)
    g, v, d, c = s1.geometry, s1.vertices, s1.dimensions,
s1.constraints
    s1.setPrimaryObject(option=STANDALONE)
    s1.ConstructionLine(point1=(0.0, -500.0), point2=(0.0,
500.0))
    s1.FixedConstraint(entity=g[2])
    s1.Spline(points=((
0.0, 10),(
0.5, 6.027864045),(
1, 4.67544468),(
1.5, 3.754033308),(
2, 3.05572809),(
2.5, 2.5),(
3, 2.04554885),(
3.5, 1.667840434),(
4, 1.350889359),(
4.5, 1.083592135),(
5, 0.857864376),(
5.5, 0.667603026),(
6, 0.508066615),(
6.5, 0.375484503),(
7, 0.266799469),(
7.5, 0.179491924),(
8, 0.11145618),(
8.5, 0.060911085),(
9, 0.026334039),(
9.5, 0.00641131),(
10,0),(
9.5,-0.00641131),(
9,-0.026334039),(
8.5,-0.060911085),(
8,-0.11145618),(
7.5,-0.179491924),(
7,-0.266799469),(
6.5,-0.375484503),(
6,-0.508066615),(
5.5,-0.667603026),(
5,-0.857864376),(
4.5,-1.083592135),(
4,-1.350889359),(
3.5,-1.667840434),(
3,-2.04554885),(
2.5,-2.5),(
2,-3.05572809),(

```

```

1.5,-3.754033308),(
1,-4.67544468),(
0.5,-6.027864045),(
0,-10))
    s1.Line(point1=(0.0,10.0), point2=(0.0,-10.0))
    s1.VerticalConstraint(entity=g[4])
    p = mdb.models['Model-1'].Part(name='nodulo',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
    p = mdb.models['Model-1'].parts['nodulo']
    p.BaseSolidRevolve(sketch=s1, angle=360.0,
flipRevolveDirection=OFF)
    s1.unsetPrimaryObject()
    p = mdb.models['Model-1'].parts['nodulo']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Assembly cubo_macico
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['cubo_macico']
    a1.Instance(name='cubo_macico-1', part=p, dependent=OFF)
# Insere os nodulos no cubo Distribuicao aleatoria
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-1', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,1,0), angle=51.68)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=90.0)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,1,0), angle=91.14)
    a1.translate(instanceList=('nodulo-1', ),
vector=(24,10.1,18.9))
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-2', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.translate(instanceList=('nodulo-2', ),
vector=(33.2,43.4,19.3))
    a1.rotate(instanceList=('nodulo-2', ),
axisPoint=(33.2,43.4,19.3),
axisDirection=(0,1,0), angle=51.68)
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-3', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.translate(instanceList=('nodulo-3', ),
vector=(12.4,21.6,34.9))
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']

```

```

a1.Instance(name='nodulo-4', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-4', ),
vector=(29.2,36.7,42.5))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-5', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-5', ),
vector=(30,16.1,35.8))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-6', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-6', ),
vector=(38.3,15.2,14.7))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-7', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-7', ),
vector=(13.8,31.7,10.2))
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['nodulo']
a1.Instance(name='nodulo-8', part=p, dependent=OFF)
a1 = mdb.models['Model-1'].rootAssembly
a1.translate(instanceList=('nodulo-8', ),
vector=(13,42,37.8))
# Cria os vazios dos nodulos no cubo( cria a part cubo_vazado)
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].view.fitView()
a1 = mdb.models['Model-1'].rootAssembly
a1.PartFromBooleanCut(name='cubo_vazado',
instanceToBeCut=mdb.models['Model-
1'].rootAssembly.instances['cubo_macico-1'],
cuttingInstances=(
a1.instances['nodulo-1'],
a1.instances['nodulo-2'],
a1.instances['nodulo-3'],
a1.instances['nodulo-4'],
a1.instances['nodulo-5'],
a1.instances['nodulo-6'],
a1.instances['nodulo-7'],
a1.instances['nodulo-8'],
))
# cria a instance cubo_vazado-1 no Assembly)
a1 = mdb.models['Model-1'].rootAssembly
p = mdb.models['Model-1'].parts['cubo_vazado']
a1.Instance(name='cubo_vazado-1', part=p, dependent=OFF)

```



```

# Inserindo os nodulos nos vazios do cubo vazado
# Contato: tipo TIE
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=ON,
    interactions=ON, constraints=ON, connectors=ON,
engineeringFeatures=ON,
    adaptiveMeshConstraints=OFF)
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
    'cubo_vazado-1', ))
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
    'nodulo-1',
    'nodulo-2',
    'nodulo-3',
    'nodulo-4',
    'nodulo-5',
    'nodulo-6',
    'nodulo-7',
    'nodulo-8'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['cubo_vazado-1'].faces
    sidelFaces1 = s1.getSequenceFromMask(mask=('[#3fffc0 ]',
), )
    region1=regionToolset.Region(sidelFaces=sidelFaces1)
    a = mdb.models['Model-1'].rootAssembly
    s1 = a.instances['nodulo-1'].faces
    sidelFaces1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
    s2 = a.instances['nodulo-2'].faces
    sidelFaces2 = s2.getSequenceFromMask(mask=('[#1 ]', ), )
    s3 = a.instances['nodulo-3'].faces
    sidelFaces3 = s3.getSequenceFromMask(mask=('[#1 ]', ), )
    s4 = a.instances['nodulo-4'].faces
    sidelFaces4 = s4.getSequenceFromMask(mask=('[#1 ]', ), )
    s5 = a.instances['nodulo-5'].faces
    sidelFaces5 = s5.getSequenceFromMask(mask=('[#1 ]', ), )
    s6 = a.instances['nodulo-6'].faces
    sidelFaces6 = s6.getSequenceFromMask(mask=('[#1 ]', ), )
    s7 = a.instances['nodulo-7'].faces
    sidelFaces7 = s7.getSequenceFromMask(mask=('[#1 ]', ), )
    s8 = a.instances['nodulo-8'].faces
    sidelFaces8 = s8.getSequenceFromMask(mask=('[#1 ]', ), )
    region2=regionToolset.Region(sidelFaces=
    sidelFaces1+\

```

```

        sidelFaces2+\
        sidelFaces3+\
        sidelFaces4+\
        sidelFaces5+\
        sidelFaces6+\
        sidelFaces7+\
        sidelFaces8)
    mdb.models['Model-1'].Tie(name='tie', master=region1,
slave=region2,
        positionToleranceMethod=COMPUTED, adjust=ON,
tieRotations=ON,
        thickness=ON)
# Ajuste na tela
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1',
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
    session.viewports['Viewport: 1'].view.fitView()
# Criacao da malha 1 (14x14x7):
# Malha do cubo_vazado-1:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'cubo_vazado-1', ))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    pickedRegions = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=EXPLICIT)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=EXPLICIT)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=EXPLICIT,
        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = (cells1, )
    a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
        elemType3))
    a = mdb.models['Model-1'].rootAssembly

```

```

    partInstances =(a.instances['cubo_vazado-1'], )
# size=Lcubo2nodulos/7
    a.seedPartInstance(regions=partInstances, size=17.96,
deviationFactor=0.1)
    a = mdb.models['Model-1'].rootAssembly
    partInstances =(a.instances['cubo_vazado-1'], )
    a.generateMesh(regions=partInstances)
# Malha dos nodulos:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    c3 = a.instances['nodulo-3'].cells
    cells3 = c3.getSequenceFromMask(mask=('[#1 ]', ), )
    c4 = a.instances['nodulo-4'].cells
    cells4 = c4.getSequenceFromMask(mask=('[#1 ]', ), )
    c5 = a.instances['nodulo-5'].cells
    cells5 = c5.getSequenceFromMask(mask=('[#1 ]', ), )
    c6 = a.instances['nodulo-6'].cells
    cells6 = c6.getSequenceFromMask(mask=('[#1 ]', ), )
    c7 = a.instances['nodulo-7'].cells
    cells7 = c7.getSequenceFromMask(mask=('[#1 ]', ), )
    c8 = a.instances['nodulo-8'].cells
    cells8 = c8.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions =
cells1+cells2+cells3+cells4+cells5+cells6+cells7+cells8
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=EXPLICIT)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=EXPLICIT)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=EXPLICIT,
        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells

```

```

cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
c3 = a.instances['nodulo-3'].cells
cells3 = c3.getSequenceFromMask(mask=('[#1 ]', ), )
c4 = a.instances['nodulo-4'].cells
cells4 = c4.getSequenceFromMask(mask=('[#1 ]', ), )
c5 = a.instances['nodulo-5'].cells
cells5 = c5.getSequenceFromMask(mask=('[#1 ]', ), )
c6 = a.instances['nodulo-6'].cells
cells6 = c6.getSequenceFromMask(mask=('[#1 ]', ), )
c7 = a.instances['nodulo-7'].cells
cells7 = c7.getSequenceFromMask(mask=('[#1 ]', ), )
c8 = a.instances['nodulo-8'].cells
cells8 = c8.getSequenceFromMask(mask=('[#1 ]', ), )
pickedRegions
=((cells1+cells2+cells3+cells4+cells5+cells6+cells7+cells8), )
a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
elemType3))
a = mdb.models['Model-1'].rootAssembly
partInstances = (
a.instances['nodulo-1'],
a.instances['nodulo-2'],
a.instances['nodulo-3'],
a.instances['nodulo-4'],
a.instances['nodulo-5'],
a.instances['nodulo-6'],
a.instances['nodulo-7'],
a.instances['nodulo-8'], )
# size=(2*Pi*R/2)/14
a.seedPartInstance(regions=partInstances, size=5.12,
deviationFactor=0.1)
a = mdb.models['Model-1'].rootAssembly
partInstances = (
a.instances['nodulo-1'],
a.instances['nodulo-2'],
a.instances['nodulo-3'],
a.instances['nodulo-4'],
a.instances['nodulo-5'],
a.instances['nodulo-6'],
a.instances['nodulo-7'],
a.instances['nodulo-8'], )
a.generateMesh(regions=partInstances)
# Ativa a malha do cubo_vazado-1:
a = mdb.models['Model-1'].rootAssembly
partInstances = (a.instances['cubo_vazado-1'], )
a.generateMesh(regions=partInstances)
# Ajuste na tela
a = mdb.models['Model-1'].rootAssembly
session.viewports['Viewport:
1'].setValues(displayedObject=a)
a1 = mdb.models['Model-1'].rootAssembly

```

```

    a1.regenerate()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
# Ativa nodulos e cubo_vazado-1
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF,
visibleInstances=('cubo_vazado-1',
        'nodulo-1',
        'nodulo-2',
        'nodulo-3',
        'nodulo-4',
        'nodulo-5',
        'nodulo-6',
        'nodulo-7',
        'nodulo-8'))
# Supress cubo macico
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a = mdb.models['Model-1'].rootAssembly
    a.features['cubo_macico-1'].suppress()
# Cria materiais PMMA e Borracha
    session.viewports['Viewport:
1'].partDisplay.setValues(sectionAssignments=ON,
        engineeringFeatures=ON)
# Cubo PMMA elastico (E[N/micrometro^2])
    mdb.models['Model-1'].Material(name='PMMA')
    mdb.models['Model-
1'].materials['PMMA'].Elastic(table=((3240.0e-06, 0.25), ))
    mdb.models['Model-
1'].materials['PMMA'].Density(table=((0.1, ), ))
# Nodulos borracha hiperelastica
    mdb.models['Model-1'].Material(name='borracha')
    mdb.models['Model-
1'].materials['borracha'].Hyperelastic(type=YEOH,
        volumetricResponse=POISSON_RATIO, poissonRatio=0.49,
table=())
    mdb.models['Model-
1'].materials['borracha'].hyperelastic.UniaxialTestData(
        table=((0.0e-06, 0.0), (0.067493e-06, 0.013173),
(0.122799e-06, 0.022706), (
        0.170091e-06, 0.031698), (0.215172e-06, 0.042221),
(0.256204e-06, 0.050924), (
        0.294346e-06, 0.061872), (0.331216e-06, 0.07209),
(0.365922e-06, 0.082178), (
        0.398546e-06, 0.09223), (0.429188e-06, 0.103614),
(0.460153e-06, 0.113874), (

```

0.488826e-06, 0.1232), (0.516538e-06, 0.135243),
(0.543559e-06, 0.147037), (
0.570611e-06, 0.158219), (0.596299e-06, 0.169079),
(0.620352e-06, 0.179255), (
0.643838e-06, 0.190456), (0.667422e-06, 0.2014),
(0.689879e-06, 0.211714), (
0.71129e-06, 0.222107), (0.733281e-06, 0.231915),
(0.753824e-06, 0.242115), (
0.775483e-06, 0.253571), (0.794915e-06, 0.264952),
(0.814219e-06, 0.277722), (
0.834128e-06, 0.287577), (0.852405e-06, 0.298812),
(0.871407e-06, 0.308476), (
0.890574e-06, 0.318789), (0.908795e-06, 0.329664),
(0.926346e-06, 0.340022), (
0.944297e-06, 0.352691), (0.962587e-06, 0.363463),
(0.978529e-06, 0.373858), (
0.99624e-06, 0.384217), (1.013976e-06, 0.395852),
(1.031494e-06, 0.405303), (
1.047365e-06, 0.416323), (1.065122e-06, 0.427394),
(1.08113e-06, 0.43914), (
1.098552e-06, 0.451832), (1.11521e-06, 0.462129),
(1.131931e-06, 0.473704), (
1.148226e-06, 0.483077), (1.165724e-06, 0.492492),
(1.180919e-06, 0.501916), (
1.19806e-06, 0.513476), (1.214544e-06, 0.52601),
(1.231202e-06, 0.536938), (
1.247524e-06, 0.547675), (1.265773e-06, 0.560064),
(1.281563e-06, 0.569819), (
1.299028e-06, 0.580185), (1.316158e-06, 0.591409),
(1.332376e-06, 0.602582), (
1.349576e-06, 0.614422), (1.367571e-06, 0.623176),
(1.383819e-06, 0.635066), (
1.402139e-06, 0.645278), (1.419228e-06, 0.654957),
(1.438199e-06, 0.665256), (
1.45591e-06, 0.676374), (1.473884e-06, 0.68719),
(1.492489e-06, 0.69775), (
1.510456e-06, 0.706527), (1.528951e-06, 0.716974),
(1.5482e-06, 0.726045), (
1.566719e-06, 0.736834), (1.586524e-06, 0.746179),
(1.606595e-06, 0.755446), (
1.626638e-06, 0.766364), (1.648778e-06, 0.776606),
(1.668463e-06, 0.786105), (
1.689116e-06, 0.795061), (1.711186e-06, 0.806769),
(1.733663e-06, 0.81744), (
1.755887e-06, 0.827993), (1.778442e-06, 0.83807),
(1.804289e-06, 0.848048), (
1.828315e-06, 0.856081), (1.853119e-06, 0.864174),
(1.880331e-06, 0.872358), (
1.906031e-06, 0.881806), (1.933201e-06, 0.892394),
(1.961445e-06, 0.901593), (

```

        1.990647e-06, 0.910831), (2.020123e-06, 0.918884),
(2.051192e-06, 0.927358), (
        2.082824e-06, 0.937684), (2.115919e-06, 0.944883),
(2.149674e-06, 0.952773), (
        2.184837e-06, 0.962062), (2.220966e-06, 0.971009),
(2.257541e-06, 0.978642), (
        2.295553e-06, 0.987007), (2.336457e-06, 0.996596),
(2.376337e-06, 1.004143), (
        2.419025e-06, 1.011558), (2.462917e-06, 1.020487),
(2.508152e-06, 1.026605), (
        2.555243e-06, 1.035424), (2.603459e-06, 1.042),
(2.654023e-06, 1.048794), (
        2.704187e-06, 1.057041), (2.758468e-06, 1.065288)))
    mdb.models['Model-
1'].materials['borracha'].Density(table=((0.1, ), ))
# Cria secos cubo e nodulo
    mdb.models['Model-1'].HomogeneousSolidSection(name='cubo',
material='PMMA',
        thickness=1.0)
    mdb.models['Model-
1'].HomogeneousSolidSection(name='nodulo',
        material='borracha', thickness=1.0)
# Assigna secas nodulo a nodulo
    p = mdb.models['Model-1'].parts['nodulo']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['nodulo']
    p.SectionAssignment(region=region, sectionName='nodulo',
offset=0.0)
# Assigna secas cubo a cubo vazado
    p = mdb.models['Model-1'].parts['cubo_vazado']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    p.SectionAssignment(region=region, sectionName='cubo',
offset=0.0)
# Cria o Step
    mdb.models['Model-1'].ExplicitDynamicsStep(name='final',
previous='Initial',
        timePeriod=1e-08)
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(step='final')
# Cria Amplitude para rodar explícito
    mdb.models['Model-1'].TabularAmplitude(name='Amp-1',
timeSpan=STEP,
        smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8, 1.0)))

```

```

        mdb.models['Model-1'].TabularAmplitude(name='Amp-2',
timeSpan=STEP,
            smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8, 1.0)))
        mdb.models['Model-1'].TabularAmplitude(name='Amp-3',
timeSpan=STEP,
            smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8, 1.0)))
# Cria BC
# Deslocamento prescrito
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#10 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-1',
createStepName='final',
        region=region, u1=UNSET, u2=UNSET, u3=1.0, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-3', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Back'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#20 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-2',
createStepName='final',
        region=region, u1=UNSET, u2=UNSET, u3=-1.0, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-3', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Top'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#2 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-3',
createStepName='final',
        region=region, u1=UNSET, u2=1.0, u3=UNSET, url=UNSET,
ur2=UNSET,

```



```

        ur3=UNSET, amplitude='Amp-2', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Bottom'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#8 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-4',
createStepName='final',
        region=region, u1=UNSET, u2=-1.0, u3=UNSET, ur1=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-2', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Left'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-5',
createStepName='final',
        region=region, u1=-1.0, u2=UNSET, u3=UNSET, ur1=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-1', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Right'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#4 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-6',
createStepName='final',
        region=region, u1=1.0, u2=UNSET, u3=UNSET, ur1=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-1', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
    session.viewports['Viewport: 1'].view.fitView()
# Tela branca

session.graphicsOptions.setValues(background='FFFFFF')
# Ajustes na tela

```

```

    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
# opcoes .odb
    mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(variables=(
    'S', 'PE', 'PEEQ', 'LE', 'U', 'RF','CSTRESS',
    'EVOL'))
# Ajustes no numero de intervalosField e History output
request e no Tie antes de rodar
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(interactions=OFF,
    constraints=OFF, connectors=OFF,
engineeringFeatures=OFF,
    adaptiveMeshConstraints=ON)
    mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(
    numIntervals=10)
    mdb.models['Model-1'].historyOutputRequests['H-Output-
1'].setValues(
    numIntervals=10)
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(interactions=ON,
    constraints=ON, connectors=ON, engineeringFeatures=ON,
    adaptiveMeshConstraints=OFF)
    mdb.models['Model-1'].constraints['tie'].setValues(
    constraintEnforcement=SURFACE_TO_SURFACE)
# RUN
    mdb.Job(name='gill', model='Model-1', type=ANALYSIS,
explicitPrecision=SINGLE,
    nodalOutputPrecision=SINGLE, description='',
    parallelizationMethodExplicit=DOMAIN,
multiprocessingMode=DEFAULT,
    numDomains=1, userSubroutine='', numCpus=1,
preMemory=2024.0,
    standardMemory=1536.0, standardMemoryPolicy=MODERATE,
scratch='',
    echoPrint=OFF, modelPrint=OFF, contactPrint=OFF,
historyPrint=OFF)
    mdb.jobs['gill'].submit(consistencyChecking=OFF)
    mdb.jobs['gill'].waitForCompletion()
# saida de tensoes e deformacoes e inp
    o3 = session.openOdb(name='C:/Temp/gill.odb')
    session.viewports['Viewport:
1'].setValues(displayedObject=o3)
    session.viewports['Viewport:
1'].odbDisplay.display.setValues(plotState=(
    CONTOURS_ON_DEF, ))
    odb = session.odbs['C:/Temp/gill.odb']
    session.writeFieldReport(fileName='saída.rpt', append=ON,

```

```

        sortItem='Element Label', odb=odb, step=0, frame=10,
        outputPosition=INTEGRATION_POINT, variable= (('LE',
INTEGRATION_POINT, (
        (COMPONENT, 'LE11'), (COMPONENT, 'LE22'), (COMPONENT,
'LE33'), (
        COMPONENT, 'LE12'), (COMPONENT, 'LE13'), (COMPONENT,
'LE23'), )), ('S',
        INTEGRATION_POINT, ((INVARIANT, 'Mises'), (COMPONENT,
'S11'), (
        COMPONENT, 'S22'), (COMPONENT, 'S33'), (COMPONENT,
'S12'), (COMPONENT,
        'S13'), (COMPONENT, 'S23'), )), ))
    session.writeFieldReport(fileName='saída.rpt', append=ON,
        sortItem='Element Label', odb=odb, step=0, frame=10,
        outputPosition=WHOLE_ELEMENT, variable= (('EVOL',
WHOLE_ELEMENT), ))

```

C.2.10 Listagem do arquivo.dat da Macro Abaqus Modificada (arquivo “abaqusMacros”)

```

# Do not delete the following import lines
from abaqus import *
from abaqusConstants import *
# Abertura Macro Geral
def Assembly_cubo():
    import section
    import section
    import regionToolset
    import displayGroupMdbToolset as dgm
    import part
    import material
    import assembly
    import step
    import interaction
    import load
    import mesh
    import job
    import sketch
    import visualization
    import xyPlot
    import displayGroupOdbToolset as dgo
    import connectorBehavior
# Cria o cubo_macico
    s = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
        sheetSize=1000.0)
        g, v, d, c = s.geometry, s.vertices, s.dimensions,
s.constraints

```

```

s.setPrimaryObject(option=STANDALONE)
s.rectangle(point1=(0.0, 0.0), point2=(
4.37562,4.37562))
p = mdb.models['Model-1'].Part(name='cubo_macico',
dimensionality=THREE_D,
type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['cubo_macico']
p.BaseSolidExtrude(sketch=s, depth=
4.37562)
s.unsetPrimaryObject()
p = mdb.models['Model-1'].parts['cubo_macico']
session.viewports['Viewport:
1'].setValues(displayedObject=p)
del mdb.models['Model-1'].sketches['__profile__']
# Cria o nodulo
s1 = mdb.models['Model-
1'].ConstrainedSketch(name='__profile__',
sheetSize=1000.0)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions,
s1.constraints
s1.setPrimaryObject(option=STANDALONE)
s1.ConstructionLine(point1=(0.0, -500.0), point2=(0.0,
500.0))
s1.FixedConstraint(entity=g[2])
s1.Spline(points=((
-6.12323e-017,-1),(
-0.173648,-0.984808),(
-0.34202,-0.939693),(
-0.5,-0.866025),(
-0.642788,-0.766044),(
-0.766044,-0.642788),(
-0.866025,-0.5),(
-0.939693,-0.34202),(
-0.984808,-0.173648),(
-1,0),(
-0.984808,0.173648),(
-0.939693,0.34202),(
-0.866025,0.5),(
-0.766044,0.642788),(
-0.642788,0.766044),(
-0.5,0.866025),(
-0.34202,0.939693),(
-0.173648,0.984808),(
-6.12323e-017,1)))
s1.Line(point1=(0.0,-1), point2=(0.0,1))
s1.VerticalConstraint(entity=g[4])
p = mdb.models['Model-1'].Part(name='nodulo',
dimensionality=THREE_D,
type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['nodulo']

```

```

    p.BaseSolidRevolve(sketch=s1, angle=360.0,
flipRevolveDirection=OFF)
    s1.unsetPrimaryObject()
    p = mdb.models['Model-1'].parts['nodulo']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    del mdb.models['Model-1'].sketches['__profile__']
# Assembly cubo_macico
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['cubo_macico']
    a1.Instance(name='cubo_macico-1', part=p, dependent=OFF)
# Insere os nodulos no cubo Distribuicao aleatoria
    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-1', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-63.6722)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(1,0,0), angle=-40.9223)
    a1.rotate(instanceList=('nodulo-1', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-80.5106)
    a1.translate(instanceList=('nodulo-1', ),
vector=(1.25119,2.37886,3.13153))

    a1 = mdb.models['Model-1'].rootAssembly
    p = mdb.models['Model-1'].parts['nodulo']
    a1.Instance(name='nodulo-2', part=p, dependent=OFF)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-39.8224)
    a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(1,0,0), angle=-27.4438)
    a1.rotate(instanceList=('nodulo-2', ), axisPoint=(0,0,0),
axisDirection=(0,0,1), angle=-84.3578)
    a1.translate(instanceList=('nodulo-2', ),
vector=(3.18714,2.52367,1.95385))

# Cria os vazios dos nodulos no cubo( cria a part cubo_vazado)
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    session.viewports['Viewport: 1'].view.fitView()
    a1 = mdb.models['Model-1'].rootAssembly
    a1.PartFromBooleanCut(name='cubo_vazado',
instanceToBeCut=mdb.models['Model-
1'].rootAssembly.instances['cubo_macico-1'],
cuttingInstances=(
    a1.instances['nodulo-1'],
    a1.instances['nodulo-2'],
))

```

```

# cria a instance cubo_vazado-1 no Assembly)
  a1 = mdb.models['Model-1'].rootAssembly
  p = mdb.models['Model-1'].parts['cubo_vazado']
  a1.Instance(name='cubo_vazado-1', part=p, dependent=OFF)
# Inserindo os nodulos nos vazios do cubo vazado
# Contato: tipo TIE
  session.viewports['Viewport:
1'].assemblyDisplay.setValues(mesh=ON,
  interactions=ON, constraints=ON, connectors=ON,
engineeringFeatures=ON,
  adaptiveMeshConstraints=OFF)
  session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
  'cubo_vazado-1', ))
  session.viewports['Viewport: 1'].view.fitView()
  session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
  session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
  session.viewports['Viewport: 1'].view.fitView()
  session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
  'nodulo-1',
  'nodulo-2'))
  session.viewports['Viewport: 1'].view.fitView()
  a = mdb.models['Model-1'].rootAssembly
  s1 = a.instances['cubo_vazado-1'].faces
  sidelFaces1 = s1
  region1=regionToolset.Region(sidelFaces=sidelFaces1)
  a = mdb.models['Model-1'].rootAssembly
  s1 = a.instances['nodulo-1'].faces
  sidelFaces1 = s1.getSequenceFromMask(mask=('[#1 ]', ), )
  s2 = a.instances['nodulo-2'].faces
  sidelFaces2 = s2.getSequenceFromMask(mask=('[#1 ]', ), )
  region2=regionToolset.Region(sidelFaces=
  sidelFaces1+\
  sidelFaces2)
  mdb.models['Model-1'].Tie(name='tie', master=region1,
slave=region2,
  positionToleranceMethod=COMPUTED, adjust=ON,
tieRotations=ON,
  thickness=ON)
# Ajuste na tela
  session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
  'cubo_vazado-1',
  'nodulo-1',
  'nodulo-2'))
  session.viewports['Viewport: 1'].view.fitView()
# Criacao da malha 1 (14x14x7):
# Malha do cubo_vazado-1:

```

```

    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
    'cubo_vazado-1', ))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    pickedRegions = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=EXPLICIT)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=EXPLICIT)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=EXPLICIT,
    distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['cubo_vazado-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = (cells1, )
    a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
    elemType3))
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (a.instances['cubo_vazado-1'], )
# size=Lcubo2nodulos/7
    a.seedPartInstance(regions=partInstances, size=0.3003,
deviationFactor=0.1)
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (a.instances['cubo_vazado-1'], )
    a.generateMesh(regions=partInstances)
# Malha dos nodulos:
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(visibleInstances=(
    'nodulo-1',
    'nodulo-2'))
    session.viewports['Viewport: 1'].view.fitView()
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = cells1+cells2
    a.setMeshControls(regions=pickedRegions, elemShape=TET,
technique=FREE)
    elemType1 = mesh.ElemType(elemCode=C3D8R,
elemLibrary=EXPLICIT)
    elemType2 = mesh.ElemType(elemCode=C3D6,
elemLibrary=EXPLICIT)
    elemType3 = mesh.ElemType(elemCode=C3D4,
elemLibrary=EXPLICIT,

```

```

        distortionControl=DEFAULT)
    a = mdb.models['Model-1'].rootAssembly
    c1 = a.instances['nodulo-1'].cells
    cells1 = c1.getSequenceFromMask(mask=('[#1 ]', ), )
    c2 = a.instances['nodulo-2'].cells
    cells2 = c2.getSequenceFromMask(mask=('[#1 ]', ), )
    pickedRegions = ((cells1+cells2), )
    a.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2,
            elemType3))
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (
        a.instances['nodulo-1'],
        a.instances['nodulo-2'], )
# size=minimo
    a.seedPartInstance(regions=partInstances, size=1.5,
deviationFactor=0.1)
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (
        a.instances['nodulo-1'],
        a.instances['nodulo-2'], )
    a.generateMesh(regions=partInstances)
# Ativa a malha do cubo_vazado-1:
    a = mdb.models['Model-1'].rootAssembly
    partInstances = (a.instances['cubo_vazado-1'], )
    a.generateMesh(regions=partInstances)
# Ajuste na tela
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a1 = mdb.models['Model-1'].rootAssembly
    a1.regenerate()
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
# Ativa nodulos e cubo_vazado-1
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(
        adaptiveMeshConstraints=OFF,
visibleInstances=('cubo_vazado-1',
        'nodulo-1',
        'nodulo-2'))
# Supress cubo macico
    a = mdb.models['Model-1'].rootAssembly
    session.viewports['Viewport:
1'].setValues(displayedObject=a)
    a = mdb.models['Model-1'].rootAssembly
    a.features['cubo_macico-1'].suppress()
# Cria materiais PMMA e Borracha

```



```

    session.viewports['Viewport:
1'].partDisplay.setValues(sectionAssignments=ON,
    engineeringFeatures=ON)
# Cubo PMMA elastico (E[N/micrometro^2])
    mdb.models['Model-1'].Material(name='PMMA')
    mdb.models['Model-
1'].materials['PMMA'].Elastic(table=((0.00324, 0.25), ))
    mdb.models['Model-
1'].materials['PMMA'].Density(table=((1.2e-021, ), ))
# Nodulos borracha hiperelástica
    mdb.models['Model-1'].Material(name='borracha')
    mdb.models['Model-
1'].materials['borracha'].Hyperelastic(type=YEOH,
    volumetricResponse=POISSON_RATIO, poissonRatio=0.49,
table=())
    mdb.models['Model-
1'].materials['borracha'].hyperelastic.UniaxialTestData(
    table=((0.0e-06, 0.0), (0.067493e-06, 0.013173),
(0.122799e-06, 0.022706), (
    0.170091e-06, 0.031698), (0.215172e-06, 0.042221),
(0.256204e-06, 0.050924), (
    0.294346e-06, 0.061872), (0.331216e-06, 0.07209),
(0.365922e-06, 0.082178), (
    0.398546e-06, 0.09223), (0.429188e-06, 0.103614),
(0.460153e-06, 0.113874), (
    0.488826e-06, 0.1232), (0.516538e-06, 0.135243),
(0.543559e-06, 0.147037), (
    0.570611e-06, 0.158219), (0.596299e-06, 0.169079),
(0.620352e-06, 0.179255), (
    0.643838e-06, 0.190456), (0.667422e-06, 0.2014),
(0.689879e-06, 0.211714), (
    0.71129e-06, 0.222107), (0.733281e-06, 0.231915),
(0.753824e-06, 0.242115), (
    0.775483e-06, 0.253571), (0.794915e-06, 0.264952),
(0.814219e-06, 0.277722), (
    0.834128e-06, 0.287577), (0.852405e-06, 0.298812),
(0.871407e-06, 0.308476), (
    0.890574e-06, 0.318789), (0.908795e-06, 0.329664),
(0.926346e-06, 0.340022), (
    0.944297e-06, 0.352691), (0.962587e-06, 0.363463),
(0.978529e-06, 0.373858), (
    0.99624e-06, 0.384217), (1.013976e-06, 0.395852),
(1.031494e-06, 0.405303), (
    1.047365e-06, 0.416323), (1.065122e-06, 0.427394),
(1.08113e-06, 0.43914), (
    1.098552e-06, 0.451832), (1.11521e-06, 0.462129),
(1.131931e-06, 0.473704), (
    1.148226e-06, 0.483077), (1.165724e-06, 0.492492),
(1.180919e-06, 0.501916), (
    1.19806e-06, 0.513476), (1.214544e-06, 0.52601),
(1.231202e-06, 0.536938), (

```

```

1.247524e-06, 0.547675), (1.265773e-06, 0.560064),
(1.281563e-06, 0.569819), (
1.299028e-06, 0.580185), (1.316158e-06, 0.591409),
(1.332376e-06, 0.602582), (
1.349576e-06, 0.614422), (1.367571e-06, 0.623176),
(1.383819e-06, 0.635066), (
1.402139e-06, 0.645278), (1.419228e-06, 0.654957),
(1.438199e-06, 0.665256), (
1.45591e-06, 0.676374), (1.473884e-06, 0.68719),
(1.492489e-06, 0.69775), (
1.510456e-06, 0.706527), (1.528951e-06, 0.716974),
(1.5482e-06, 0.726045), (
1.566719e-06, 0.736834), (1.586524e-06, 0.746179),
(1.606595e-06, 0.755446), (
1.626638e-06, 0.766364), (1.648778e-06, 0.776606),
(1.668463e-06, 0.786105), (
1.689116e-06, 0.795061), (1.711186e-06, 0.806769),
(1.733663e-06, 0.81744), (
1.755887e-06, 0.827993), (1.778442e-06, 0.83807),
(1.804289e-06, 0.848048), (
1.828315e-06, 0.856081), (1.853119e-06, 0.864174),
(1.880331e-06, 0.872358), (
1.906031e-06, 0.881806), (1.933201e-06, 0.892394),
(1.961445e-06, 0.901593), (
1.990647e-06, 0.910831), (2.020123e-06, 0.918884),
(2.051192e-06, 0.927358), (
2.082824e-06, 0.937684), (2.115919e-06, 0.944883),
(2.149674e-06, 0.952773), (
2.184837e-06, 0.962062), (2.220966e-06, 0.971009),
(2.257541e-06, 0.978642), (
2.295553e-06, 0.987007), (2.336457e-06, 0.996596),
(2.376337e-06, 1.004143), (
2.419025e-06, 1.011558), (2.462917e-06, 1.020487),
(2.508152e-06, 1.026605), (
2.555243e-06, 1.035424), (2.603459e-06, 1.042),
(2.654023e-06, 1.048794), (
2.704187e-06, 1.057041), (2.758468e-06, 1.065288)))
mdb.models['Model-
1'].materials['borracha'].Density(table=((9.5e-022, ), ))
# Cria secoes cubo e nodulo
mdb.models['Model-1'].HomogeneousSolidSection(name='cubo',
material='PMMA',
thickness=1.0)
mdb.models['Model-
1'].HomogeneousSolidSection(name='nodulo',
material='borracha', thickness=1.0)
# Assigna secao nodulo a nodulo
p = mdb.models['Model-1'].parts['nodulo']
c = p.cells
cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
region = regionToolset.Region(cells=cells)

```

```

    p = mdb.models['Model-1'].parts['nodulo']
    p.SectionAssignment(region=region, sectionName='nodulo',
offset=0.0)
# Assigna secao cubo a cubo vazado
    p = mdb.models['Model-1'].parts['cubo_vazado']
    session.viewports['Viewport:
1'].setValues(displayedObject=p)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    c = p.cells
    cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(cells=cells)
    p = mdb.models['Model-1'].parts['cubo_vazado']
    p.SectionAssignment(region=region, sectionName='cubo',
offset=0.0)
# Cria o Step
    mdb.models['Model-1'].ExplicitDynamicsStep(name='final',
previous='Initial',
        timePeriod=1e-08)
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(step='final')
# Cria Amplitude para rodar explícito
    mdb.models['Model-1'].TabularAmplitude(name='Amp-1',
timeSpan=STEP,
        smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8,
0.437562)))
    mdb.models['Model-1'].TabularAmplitude(name='Amp-2',
timeSpan=STEP,
        smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8,
0.43975)))
    mdb.models['Model-1'].TabularAmplitude(name='Amp-3',
timeSpan=STEP,
        smooth=SOLVER_DEFAULT, data=((0.0, 0.0), (1e-8,
0.441938)))
# Cria BC
# Deslocamento prescrito
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PARALLEL)
    session.viewports['Viewport: 1'].view.fitView()
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    session.viewports['Viewport:
1'].view.setValues(session.views['Front'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#10 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-1',
createStepName='final',

```

```

        region=region, u1=UNSET, u2=UNSET, u3=1.0, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-3', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Back'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#20 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-2',
createStepName='final',
        region=region, u1=UNSET, u2=UNSET, u3=-1.0, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-3', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Top'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#2 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-3',
createStepName='final',
        region=region, u1=UNSET, u2=1.0, u3=UNSET, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-2', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Bottom'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#8 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-4',
createStepName='final',
        region=region, u1=UNSET, u2=-1.0, u3=UNSET, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-2', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Left'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#1 ]', ), )
    region = regionToolset.Region(faces=faces1)

```

```

        mdb.models['Model-1'].DisplacementBC(name='BC-5',
createStepName='final',
        region=region, u1=-1.0, u2=UNSET, u3=UNSET, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-1', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Right'])
    a = mdb.models['Model-1'].rootAssembly
    f1 = a.instances['cubo_vazado-1'].faces
    faces1 = f1.getSequenceFromMask(mask=('[#4 ]', ), )
    region = regionToolset.Region(faces=faces1)
    mdb.models['Model-1'].DisplacementBC(name='BC-6',
createStepName='final',
        region=region, u1=1.0, u2=UNSET, u3=UNSET, url=UNSET,
ur2=UNSET,
        ur3=UNSET, amplitude='Amp-1', fixed=OFF,
distributionType=UNIFORM,
        fieldName='', localCsys=None)
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
    session.viewports['Viewport: 1'].view.fitView()
# Tela branca

session.graphicsOptions.setValues(backgroundColor='#FFFFFF')
# Ajustes na tela
    session.viewports['Viewport:
1'].view.setValues(session.views['Iso'])
    session.viewports['Viewport:
1'].view.setProjection(projection=PERSPECTIVE)
# opciones .odb
    mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(variables=(
        'S', 'PE', 'PEEQ', 'LE', 'U', 'RF', 'CSTRESS',
        'EVOL'))
# Ajustes no numero de intervalosField e History output
request e no Tie antes de rodar
    session.viewports['Viewport:
1'].assemblyDisplay.setValues(interactions=OFF,
        constraints=OFF, connectors=OFF,
engineeringFeatures=OFF,
        adaptiveMeshConstraints=ON)
    mdb.models['Model-1'].fieldOutputRequests['F-Output-
1'].setValues(
        numIntervals=10)
    mdb.models['Model-1'].historyOutputRequests['H-Output-
1'].setValues(
        numIntervals=10)

```

```

    session.viewports['Viewport:
1'].assemblyDisplay.setValues(interactions=ON,
    constraints=ON, connectors=ON, engineeringFeatures=ON,
    adaptiveMeshConstraints=OFF)
    mdb.models['Model-1'].constraints['tie'].setValues(
        constraintEnforcement=SURFACE_TO_SURFACE)
# RUN
    mdb.Job(name='gill', model='Model-1', type=ANALYSIS,
explicitPrecision=SINGLE,
        nodalOutputPrecision=SINGLE, description='',
        parallelizationMethodExplicit=DOMAIN,
multiprocessingMode=DEFAULT,
        numDomains=1, userSubroutine='', numCpus=1,
preMemory=2024.0,
        standardMemory=1536.0, standardMemoryPolicy=MODERATE,
scratch='',
        echoPrint=OFF, modelPrint=OFF, contactPrint=OFF,
historyPrint=OFF)
    mdb.jobs['gill'].submit(consistencyChecking=OFF)
    mdb.jobs['gill'].waitForCompletion()
# saida de tensoes e deformacoes e inp
    o3 = session.openOdb(name='C:/Temp/gill.odb')
    session.viewports['Viewport:
1'].setValues(displayedObject=o3)
    session.viewports['Viewport:
1'].odbDisplay.display.setValues(plotState=(
        CONTOURS_ON_DEF, ))
    odb = session.odbs['C:/Temp/gill.odb']
    session.writeFieldReport(fileName='saída.rpt', append=ON,
        sortItem='Element Label', odb=odb, step=0, frame=1,
        outputPosition=INTEGRATION_POINT, variable=(( 'LE',
INTEGRATION_POINT, (
        (COMPONENT, 'LE11'), (COMPONENT, 'LE22'), (COMPONENT,
'LE33'), (
        COMPONENT, 'LE12'), (COMPONENT, 'LE13'), (COMPONENT,
'LE23'), )), ('S',
        INTEGRATION_POINT, ((INVARIANT, 'Mises'), (COMPONENT,
'S11'), (
        COMPONENT, 'S22'), (COMPONENT, 'S33'), (COMPONENT,
'S12'), (COMPONENT,
        'S13'), (COMPONENT, 'S23'), )), ))
    session.writeFieldReport(fileName='saída.rpt', append=ON,
        sortItem='Element Label', odb=odb, step=0, frame=0,
outputPosition=WHOLE_ELEMENT, variable=(( 'EVOL',
WHOLE_ELEMENT), ))
    cliCommand("""elemArr = mdb.models['Model-
1'].rootAssembly.instances['nodulo-1'].elements""")
    cliCommand("""len(elemArr)""")
    cliCommand("""elemArr = mdb.models['Model-
1'].rootAssembly.instances['cubo_vazado-1'].elements""")
    cliCommand("""len(elemArr)""")

```

```
cliCommand("""nodeArr = mdb.models['Model-  
1'].rootAssembly.instances['nodulo-1'].nodes""")  
cliCommand("""len(nodeArr)""")  
cliCommand("""nodeArr = mdb.models['Model-  
1'].rootAssembly.instances['cubo_vazado-1'].nodes""")  
cliCommand("""len(nodeArr)""")
```