

3

Algoritmos e metodologia

3.1. Construção de ensembles virtuais

Nesta seção vão ser descritos com certo nível de detalhes os algoritmos computacionais que, implementando o método RSA, foram desenvolvidos e utilizados para gerar ensembles virtuais com conjuntos de partículas de diferentes características, e com diferentes tipos de distribuição espacial, variando:

- 1) Forma:
 - a) circular
 - b) elíptica
- 2) Distribuição de tamanho:
 - a) constante
 - b) aleatória
- 3) Distribuição de orientação:
 - a) constante
 - b) aleatória
- 4) Distribuição espacial
 - a) Homogênea randômica (RHE)
 - b) Heterogênea:
 - i) clusters
 - ii) zonas de exclusão

O que dá um total de 18 tipos de ensembles diferentes para os quais foram programadas rotinas particulares (Apêndice 1) usando o software Wolfram Mathematica 7.0 [66].

A estrutura macroscópica do algoritmo de programação é a mesma para todos os casos de ensembles (Figura 15).

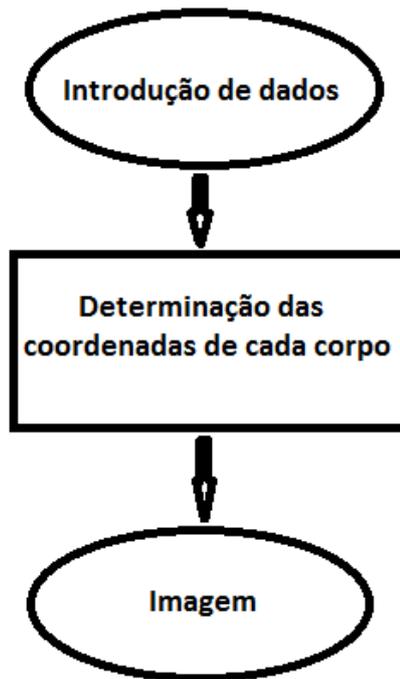


Figura 15 – Estrutura macroscópica dos algoritmos usados

Em uma primeira célula do programa são definidos os valores de cada parâmetro que define o ensemble:

- 1- Número de partículas
- 2- Tamanho da imagem em pixels
- 3- Valor do tamanho ou faixa de variação dos dois raios da partícula elíptica (dependendo do caso)
- 4- Ângulo de orientação (se é o caso em que é constante)

Para distribuições espaciais heterogêneas com clusters se adiciona:

- 5- Número de clusters
- 6- Valor do tamanho dos eixos principais dos clusters (clusters elípticos)
- 7- Posição dos clusters

Para distribuições espaciais heterogêneas com zonas de exclusão se adiciona:

5' - Número de zonas de exclusão

6' - Valor do tamanho dos eixos principais das zonas de exclusão (zonas de exclusão elípticas)

7' - Posição das zonas de exclusão

Uma segunda célula do programa calcula as coordenadas do centro de cada partícula, ao mesmo tempo em que é definido o tamanho e a orientação de cada corpo para os casos que estas quantidades não sejam constantes.

Finalmente, na terceira célula é construída a imagem que define o ensemble usando os parâmetros fornecidos pela primeira e pela segunda célula.

A determinação das coordenadas de cada partícula, que se realiza na segunda célula, é o processo chave do método RSA. A seguir vão ser expostos os algoritmos genéricos desta etapa para cada caso de distribuição espacial. Também se recomenda consultar o Apêndice 1 da tese onde estão os 18 códigos de programação íntegros e comentados, usados para cada caso particular.

3.1.1. Determinação das coordenadas de cada partícula num RHE

A Figura 16 mostra o fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula num RHE.

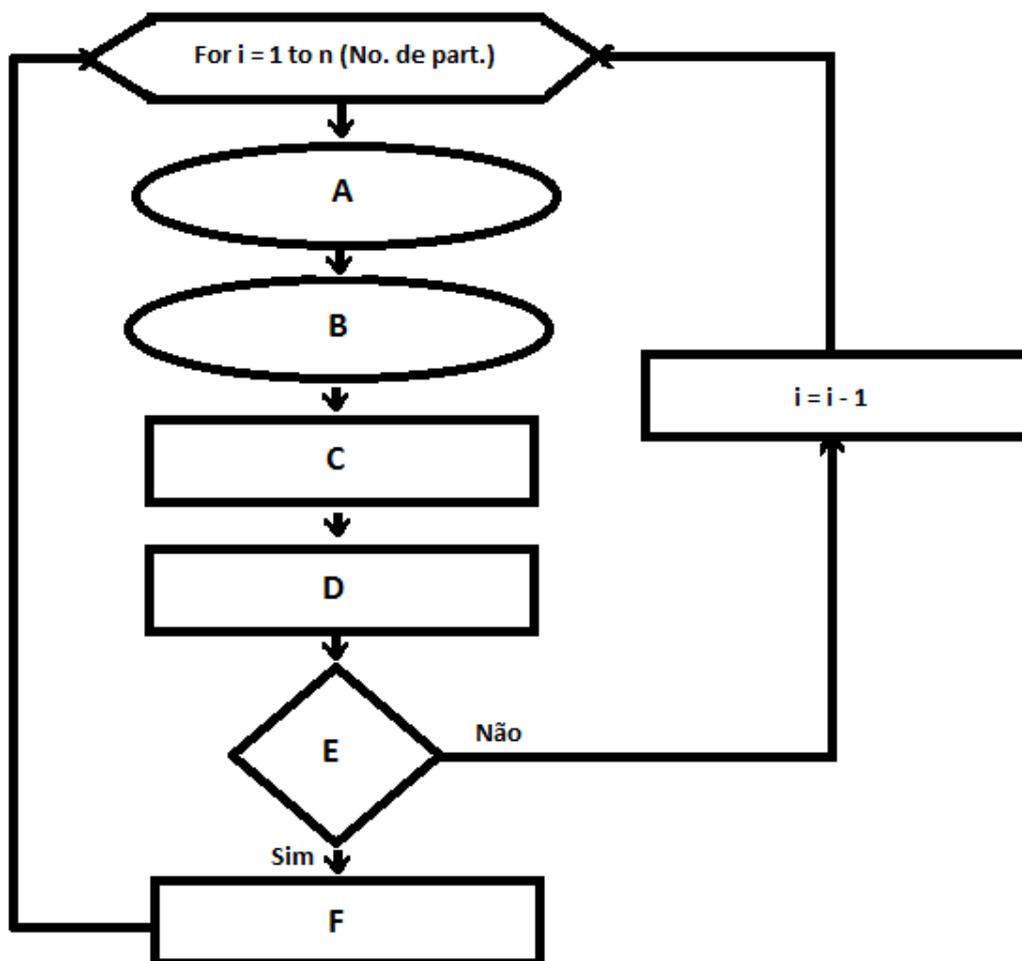


Figura 16 – Fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula num RHE. A descrição dos elementos A-F é fornecida no texto do corpo da tese.

No fluxograma:

- A) Propõe, para o corpo P_i , uma coordenada randômica qualquer a_i dentro do campo da imagem limitado por uma “área proibida”, que no começo ($i=1$) é só uma estreita margem nas bordas para evitar corpos cortados nas fronteiras, mas que cresce ao final de cada ciclo no passo “F”.
- B)
- É dado um valor randômico, dentro da faixa definida na primeira célula, para o tamanho dos eixos da elipse do corpo (só no caso de distribuição de tamanho randômica)
 - É dado um valor randômico para o ângulo de orientação (só para o caso de distribuição de orientação randômica)
- C) Constrói uma imagem auxiliar Img_i com o corpo P_i e com os restantes corpos aceitos ($P_1 \dots - P_{i-1}$)
- D) Determina o número de corpos (n_i) em Img_i .
- E) Condicional com teste: “ $n_i = i$ “
- F) O corpo é aceito na posição proposta: Se acrescenta a “área proibida” adicionando a área ocupada por P_i .

3.1.2. Determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea (clusters)

A Figura 17 amostra o fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea formando clusters.

No fluxograma:

- A) É escolhida uma semente de coordenada (um ponto) associada a cada cluster, obedecendo a uma lei de probabilidade normal centrada no ponto central do cluster, e com $\sigma(x)$ e $\sigma(y)$ correspondentes aos valores dos eixos principais dos clusters definidos na primeira célula.
- B) São eliminadas aquelas sementes que estejam na “área proibida” (mesmo conceito do algoritmo anterior). O novo conjunto são as “sementes maduras”.
- C) Condicional testando se existe alguma semente madura.
- D) É proposta para o corpo P_i , uma coordenada randômica qualquer a_i para seu centro, escolhida entre as sementes maduras
- E)
- É dado um valor randômico, dentro da faixa definida na primeira célula, para o tamanho dos eixos da elipse do corpo (só no caso de distribuição de tamanho randômica)

- É dado um valor randômico ao ângulo de orientação (só para o caso de distribuição de orientação randômica)

F) Constrói uma imagem auxiliar Img_i com o corpo P_i e com os restantes corpos aceitos ($P_1 \dots - P_{i-1}$)

G) Determina o número de corpos (n_i) em Img_i .

H) Condicional com teste: “ $n_i = i$ “

O corpo é aceito na posição proposta: Se acrescenta a área proibida adicionando a área ocupada por P_i .

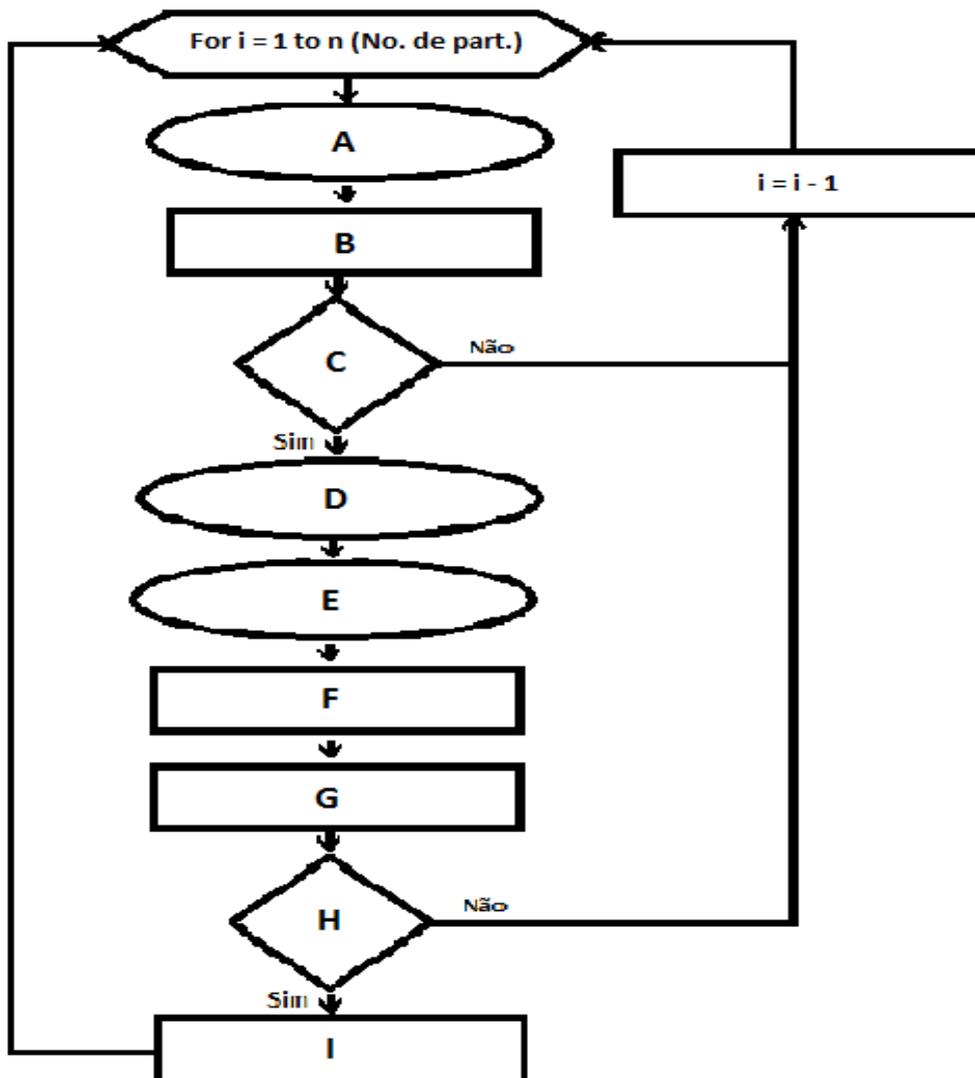


Figura 17 – Fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea formando clusters. A descrição dos elementos A - I é fornecida no texto do corpo da tese.

3.1.3. Determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea (áreas de exclusão)

A Figura 18 mostra o fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea com áreas de exclusão.

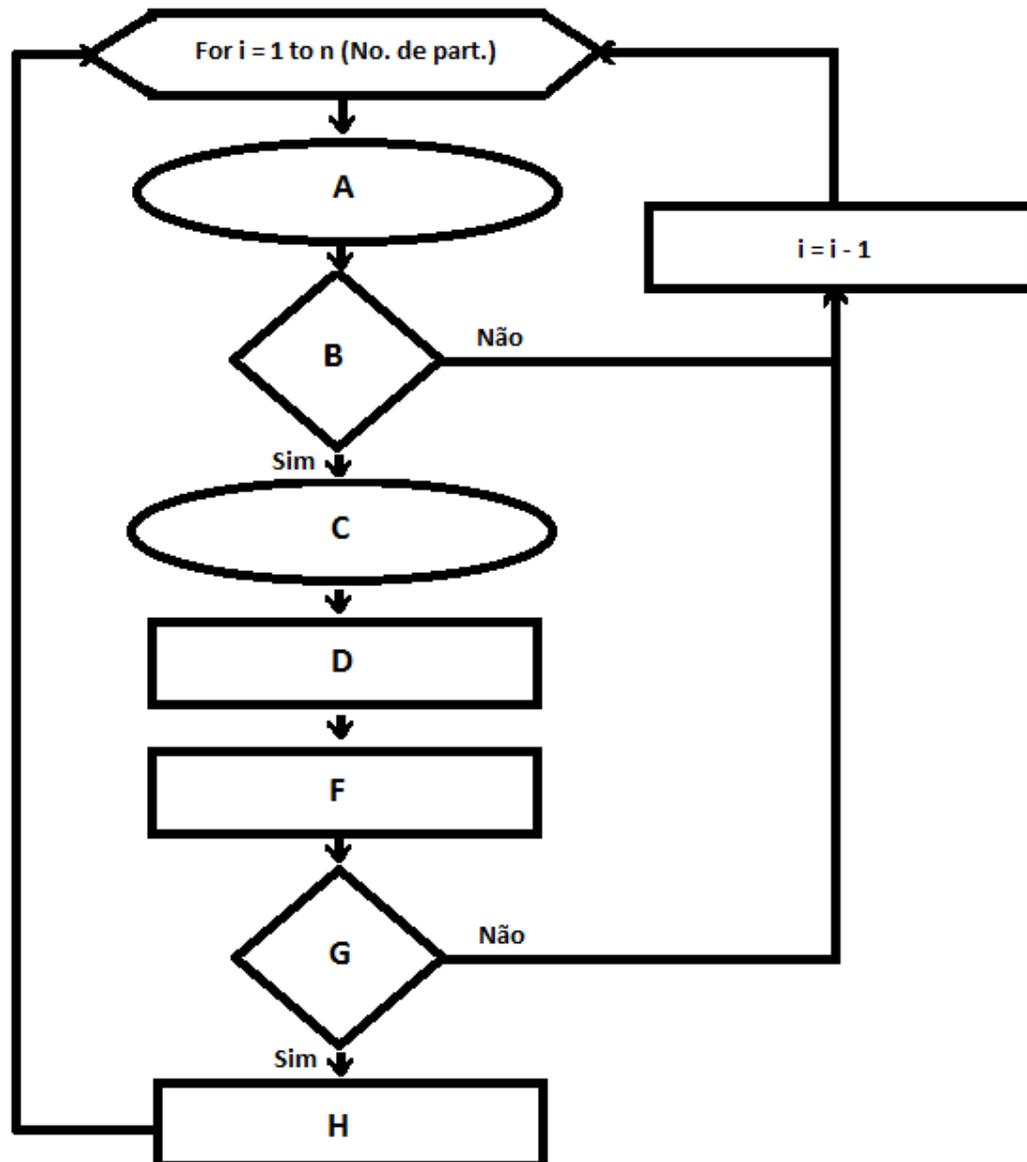


Figura 18 – Fluxograma do algoritmo genérico para a determinação das coordenadas de cada partícula em ensembles com distribuição espacial heterogênea com zonas de exclusão. A descrição dos elementos A - H é fornecida no texto do corpo da tese.

No fluxograma:

- A) Propõe, para o corpo P_i , uma coordenada randômica qualquer a_i de seu centro, dentro do campo da imagem limitado por uma área proibida, que no começo ($i=1$) é só uma estreita margem nas bordas para evitar corpos cortados nas fronteiras, mas que cresce ao final de cada ciclo no passo “H”.
- B) Condicional que testa se a_i está fora das zonas de exclusão definidas.
- C)
- É dado um valor randômico, dentro da faixa definida na primeira célula, para o tamanho dos eixos da elipse do corpo (só no caso de distribuição de tamanho randômica)
 - É dado um valor randômico ao ângulo de orientação (só para o caso de distribuição de orientação randômica)
- D) Constrói uma imagem auxiliar Img_i com o corpo P_i e com os restantes corpos aceitos ($P_1 \dots P_{i-1}$)
- E) Determina o número de corpos (n_i) em Img_i .
- F) Condicional com teste: “ $n_i = i$ “
- G) O corpo é aceito na posição proposta: Se acrescenta a área proibida adicionando a área ocupada por P_i .

3.2. Determinação do Cov_{dmean} de ensembles reais e virtuais (imagens 2D)

No Apêndice 2 está o código comentado usado para determinar o Cov_{dmean} de ensembles reais e virtuais bi-dimensionais (imagens 2D). O código foi desenvolvido no software Wolfram Mathematica 7.0. Nesta seção se pretende fazer uma descrição detalhada do algoritmo implícito nesse código.

Metodologicamente, para a descrição do algoritmo, digamos que ele tem duas etapas (Figura 19). Numa primeira é aplicada a operação de “tessellation” de Voronoi na imagem previamente segmentada (Img_0), com respeito aos corpos cuja distribuição espacial se deseja estudar. Em uma segunda etapa, usando o resultado da primeira, se determina o valor do Cov_{dmean} .

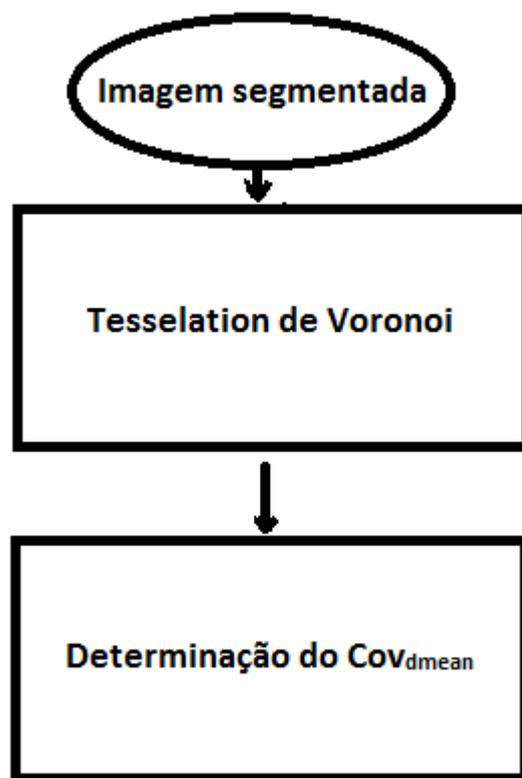


Figura 19 – Esquema geral do algoritmo para a obtenção do Cov_{dmean}

A operação de “tessellation” de Voronoi se aplica em apenas duas linhas do código e dá como resultado outra imagem (Img_{Tess}) que não é mais que, basicamente, o mapa de distâncias euclidianas ou transformada de distância² (1ª linha), adequadamente segmentado, (2ª linha) da imagem inicial (Figura 20 e Figura 21). Nesta operação o limiar de segmentação não é necessariamente o mesmo para toda Img_0 , de forma que é necessária uma terceira linha para controlar o processo. Nesta última linha se determina o número de corpos (células) de Img_{Tess} . O número de corpos de Img_{Tess} deve ser igual ao número de corpos de Img_0 (seção 1.4.2). Esta condição é suficiente para controlar o valor adequado do limiar de segmentação. Este passo, ou processo de controle, por enquanto programado para ser manual, pode ser facilmente automatizado dependendo das características do conjunto de imagens a analisar.

² Operação mediante a qual o tom de cinza de cada pixel é substituído por um valor proporcional à distância até o pixel mais próximo da outra cor.

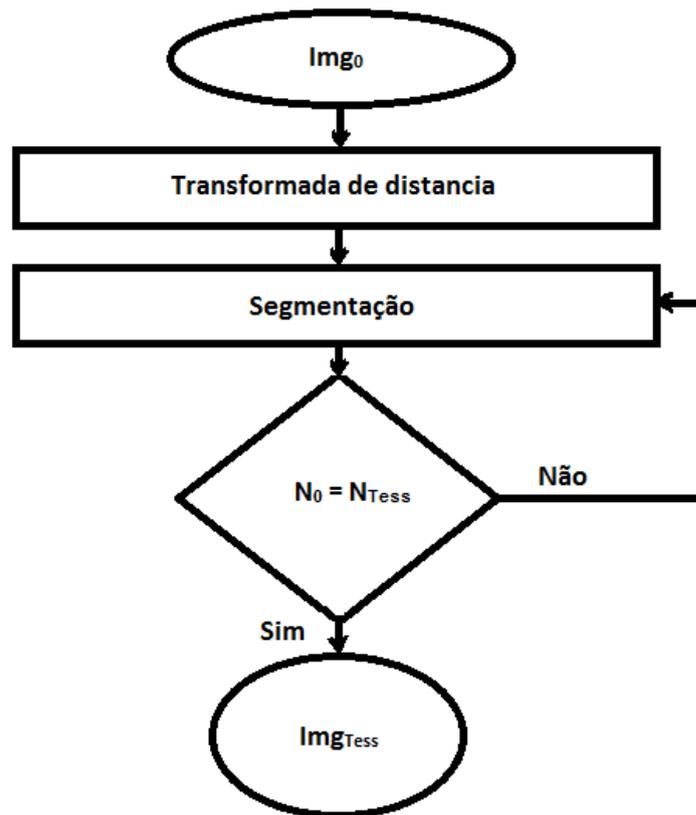


Figura 20 – Algoritmo básico usado para aplicar a operação de “tessellation” deVoronoi.

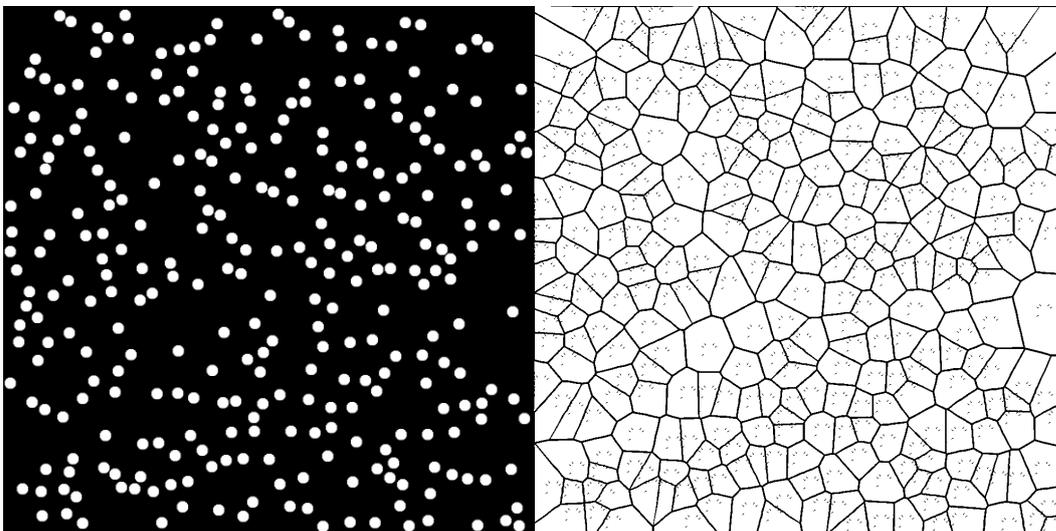


Figura 21 – Exemplo de aplicação da “tessellation” deVoronoi: Img_0 (esquerda), Img_{Tess} (direita)

Uma vez aplicado o processo de “tessellation” é possível passar à segunda etapa do programa. A Figura 22 mostra o algoritmo usado nesta fase e, logo a seguir, a descrição dos passos marcados no fluxograma.

- A) Identifica os elementos que têm vizinhança com a fronteira da imagem. *Os parâmetros locais associados a estes corpos não devem ser considerados no cálculo devido a que sua vizinhança está incompleta. Entretanto, sua condição de vizinho de outros corpos do interior da imagem é considerada.*
- B) Filtra os elementos que têm vizinhança com a fronteira.
- C) Obtém as coordenadas da célula i (ou célula(i)).
- D) Cria uma imagem em que só aparece a célula(i).
- E) Cria uma imagem da célula(i) dilatada.
- F) Cria uma imagem da célula(i) e das células vizinhas (aquelas que fazem fronteira com a célula(i)).
- G) Cria uma imagem com os corpos correspondentes às células da imagem anterior.
- H) Cria uma imagem só com os pixels da fronteira de cada corpo da imagem anterior.
- I) Cria uma imagem com os pixels da fronteira do corpo i (ou corpo(i)).
- J) Obtém as coordenadas dos pixels da fronteira do corpo(i).
- K) Obtém as coordenadas dos pixels da fronteira de cada um dos corpos vizinhos do corpo(i).
- L) A partir dos dados obtidos em K e em J é determinada a distância média aos vizinhos mais próximos (d_{mean}) para o corpo(i). *Este passo, programado numa linha só de código, é um exemplo clássico do paradigma de programação funcional com o qual o Wolfram Mathematica se identifica. Esta linha de código sintetiza um procedimento relativamente complexo, no qual se toma o mínimo das distâncias medidas desde cada ponto fronteira do corpo(i) até cada ponto da fronteira de cada vizinho, como a distância entre os dois corpos; que é determinada para cada um dos vizinhos e calculada a média do conjunto (d_{mean}).*
- M) Calcula o coeficiente de variação de d_{mean} .

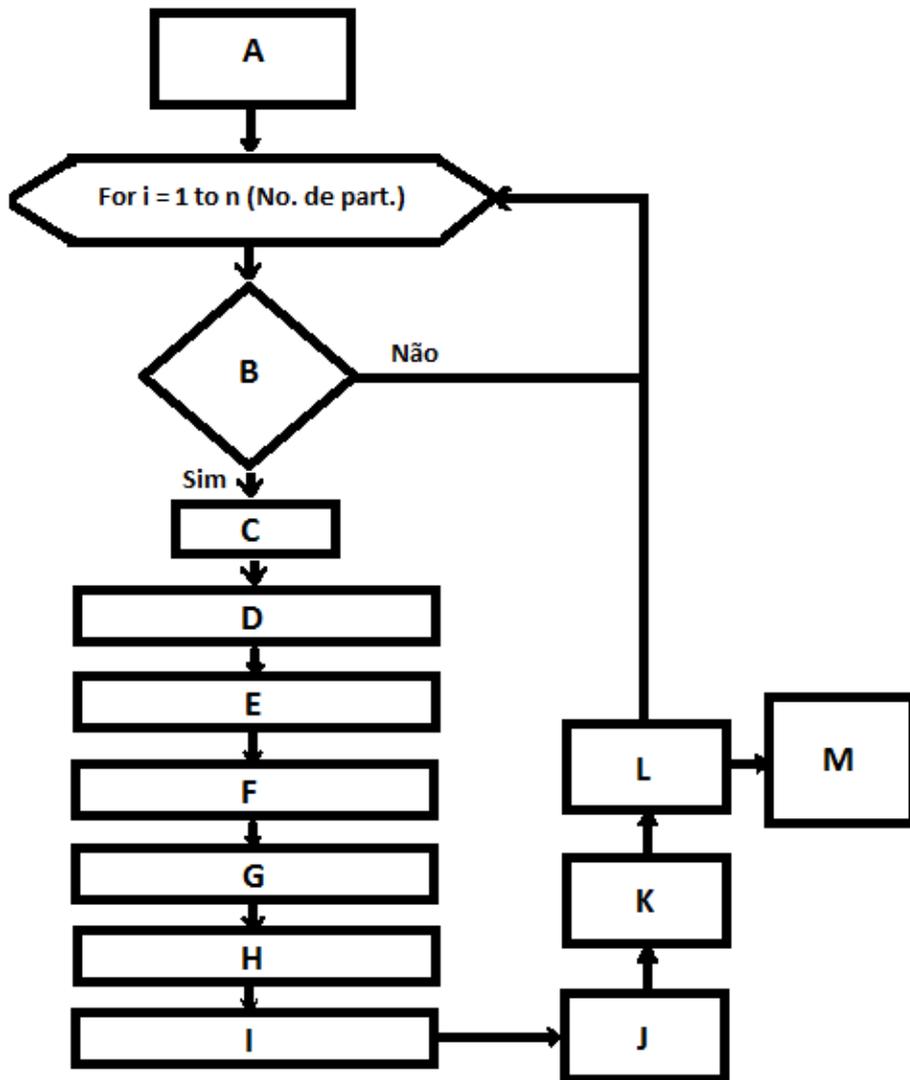


Figura 22 – Algoritmo da segunda etapa: determinação do Cov_{dmean} .

3.3.Cálculo do Cov_{dmean} local e geração de mapas de Cov_{dmean}

Para a determinação do Cov_{dmean} local o usuário deve fornecer as coordenadas do centro do campo de interesse na imagem e seu tamanho. O algoritmo da Figura 23 mostra como é realizado o processo.



Figura 23 – Rotina para a determinação do Cov_{dmean} local.

Por outro lado, para gerar um mapa de Cov_{dmean} associado a uma imagem, esta é dividida em campos locais adjacentes ou quadrantes. É o usuário que fornece o número de quadrantes em que vai ser dividida a imagem na forma $Q_f \times Q_c$ (Q_f : número de quadrantes na linha, Q_c : na coluna). A rotina para gerar o mapa é mostrada na Figura 24.

Os valores de Q_f e Q_c definem o tamanho dos quadrantes. Um dos resultados da presente tese (capítulo 4) é precisamente estabelecer um critério para o tamanho mínimo dos campos locais, tanto independentes como dos que conformam um mapa, de forma que o mapa possa ser refinado com eficiência.

Para programação das duas rotinas expostas nesta seção foi usado o software Wolfram Mathematica 7.0. Os códigos comentados estão no Apêndice 3.

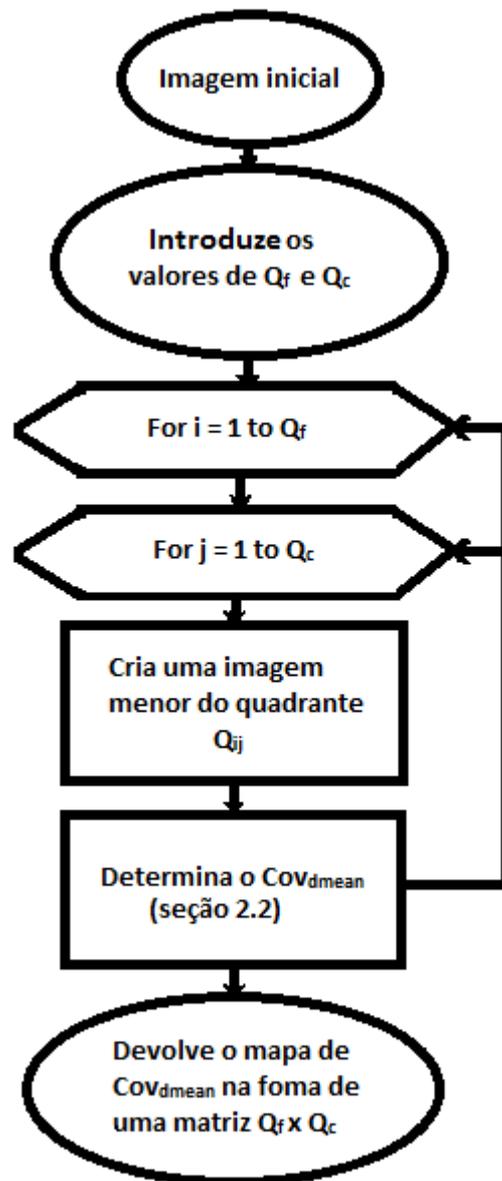


Figura 24 – Rotina para a construção do mapa de Cov_{dmean} .

3.4. Metodologia

Nesta seção será exposta a metodologia usada para cumprir com os três primeiros objetivos da tese.

3.4.1. Corroborando os resultados de N. Yang e co-autores [33]

Com o fim de reproduzir de forma genérica os resultados de [33] foram construídos ensembles com 1000 partículas em imagens de 1000 x 1000 pixels,

com as características apresentadas na Tabela 1 (unidades de comprimento em pixels).

Para cada configuração foi determinado o Cov_{dmean} médio e o desvio padrão. Para realizar esta análise, os dois procedimentos envolvidos - construção do ensemble (seção 2.1) e determinação do Cov_{dmean} (seção 2.2) - foram programados em série e dentro de um ciclo, o que massifica e agiliza o processo. A Figura 25 descreve o algoritmo deste artifício.

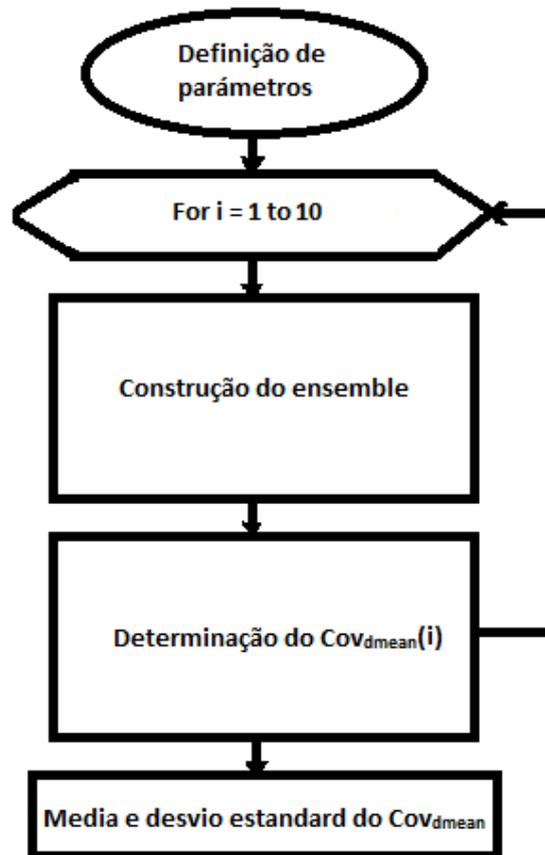


Figura 25 – Algoritmo usado para a construção e caracterização de ensembles

Tabela 1 – Configurações dos ensembles virtuais construídos

Configuração	Distribuição espacial	Distribuição de tamanho e forma	Distribuição de orientação	Posição dos centros de heterogeneidade	Tamanho das heterogeneidades	No. de ensembles
A	RHE	$R=11$	-	-	-	10
B	RHE	$6 < R < 13$	-	-	-	10
C	RHE	$R_1=6, R_2=13$	$\theta = 0$	-	-	10
D	RHE	$R_1=6, R_2=13$	randômica	-	-	10
E	RHE	$5 < R_1 < 8$ $8 < R_2 < 14$	$\theta = 0$	-	-	10
F	RHE	$5 < R_1 < 8$ $8 < R_2 < 14$	randômica	-	-	10
G	Clusters	$R=11$	-	$(250,250); (500,500); (250,750);$ $(750,250);(750,750)$	$r_1 = 60$ $r_2 = 100$	10
H	Clusters	$5 < R_1 < 8$ $8 < R_2 < 14$	randômica	$(250,250); (500,500); (250,750);$ $(750,250);(750,750)$	$r_1 = 100$ $r_2 = 150$	10
I	Zonas de exclusão	$R=11$	-	$(250,250); (500,500); (250,750);$ $(750,250);(750,750)$	$r_1 = 60$ $r_2 = 100$	10
J	Zonas de exclusão	$5 < R_1 < 8$ $8 < R_2 < 14$	randômica	$(250,250); (500,500); (250,750);$ $(750,250);(750,750)$	$r_1 = 100$ $r_2 = 150$	10

3.4.2. Estendendo a validade dos resultados de N. Yang e co-autores [33]

Para ampliar a área de validade estudada por N. Yang e co-autores foram construídos e caracterizados ensembles espaciais randômicos e homogêneos (RHE) com número de partículas abaixo de 878, e medido o valor do Cov_{dmean} . Caminhando nas direções indicadas do gráfico da Figura 26, que são:

- Mantendo a fração de área de partículas constante (15%, 25%)
- Mantendo tamanho de partículas constante

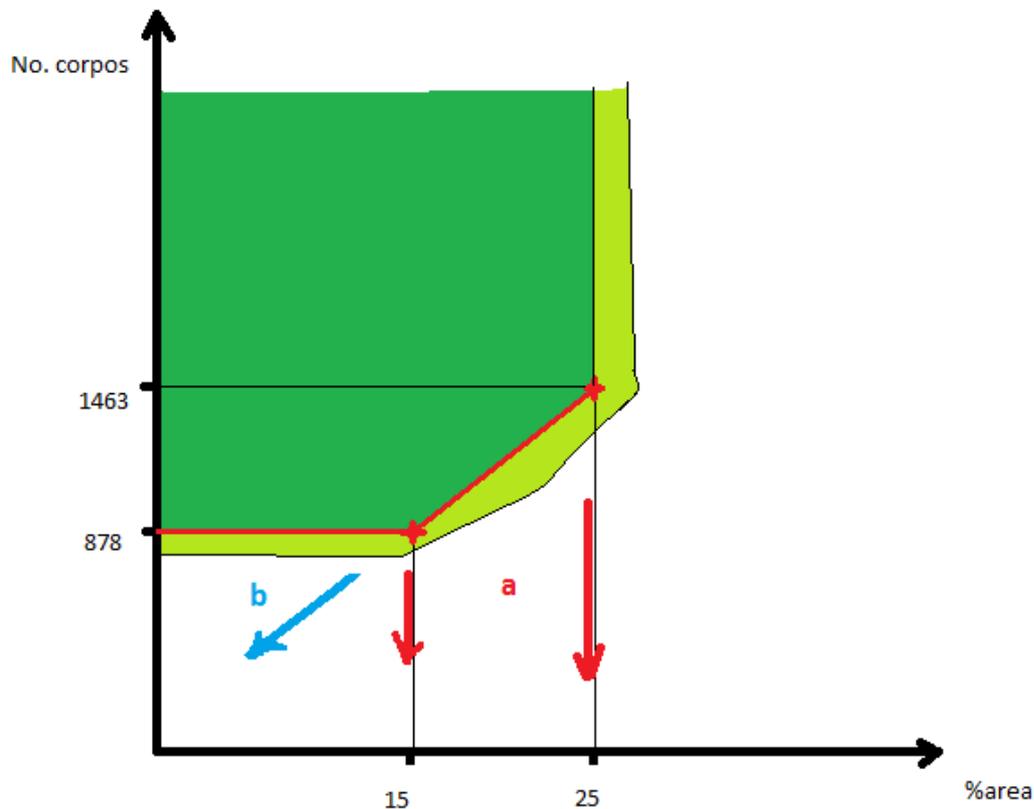


Figura 26 – Área de validade dos resultados Yang e co-autores [33] (verde): Novas direções de pesquisa propostas

Criar e caracterizar ensembles diminuindo o número de partículas sem alterar a fração de área equivale a medir o Cov_{dmean} associado a um campo local (com determinado tamanho e posição) dentro da imagem inicial. Para a pesquisa proposta nesta tese, a partir de uma imagem inicial (RHE) com 878 corpos (concentração: 15%) ou com 1463 corpos (concentração: 25%), foram escolhidos e caracterizados campos concêntricos cada vez menores. Os tamanhos destes campos foram: 75%, 50%, 30%, 10%, 5%, 2% com relação à imagem inicial. Para cada campo foi contabilizado o número de corpos.

O algoritmo computacional para determinar o Cov_{dmean} local é descrito na seção 2.3. Por outro lado, este estudo foi programado para ser totalmente automatizado mediante a rotina apresentada na Figura 27. Como indica o fluxograma o procedimento foi realizado 10 vezes para garantir um adequado suporte estatístico.

O restante dos parâmetros escolhidos para a imagem inicial foi:

- Tamanho da imagem 1000x1000.
- Corpos circulares: $r = 8$.

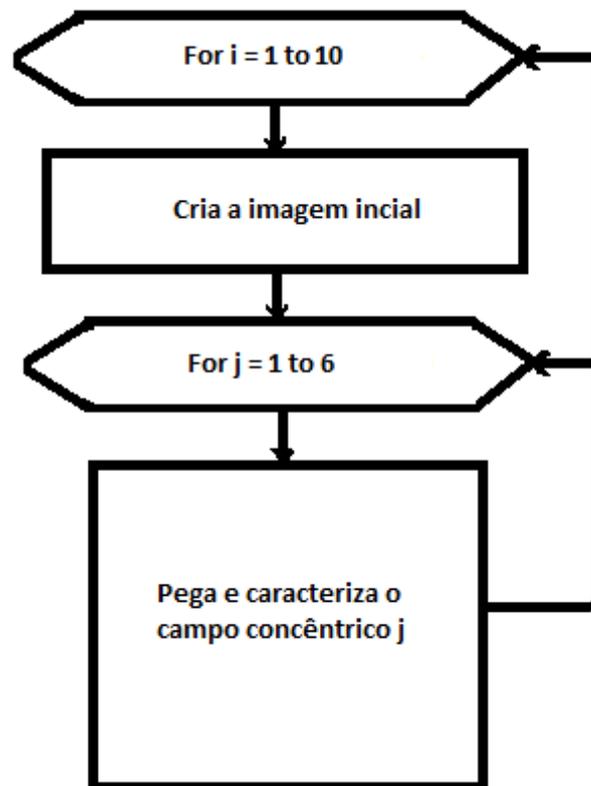


Figura 27 – Algoritmo usado para criar e caracterizar ensembles diminuindo o número de partículas sem alterar a fração de área.

Por outro lado, construir e caracterizar ensembles diminuindo o número de partículas mantendo o tamanho delas constante é simplesmente criar imagens com cada vez menos corpos usando o mesmo valor para o restante dos parâmetros. Para este estudo foram construídos RHE com as seguintes quantidades de partículas: 878, 600, 400, 200, 100, 50 e 25.

- Foram construídos 10 ensembles para cada valor do número de partículas.
- O tamanho das imagens geradas foi de 1000x1000.
- Partículas circulares: $r = 8$.

Para automatizar esta pesquisa, foi implementada uma simples rotina cíclica mostrada na Figura 28.

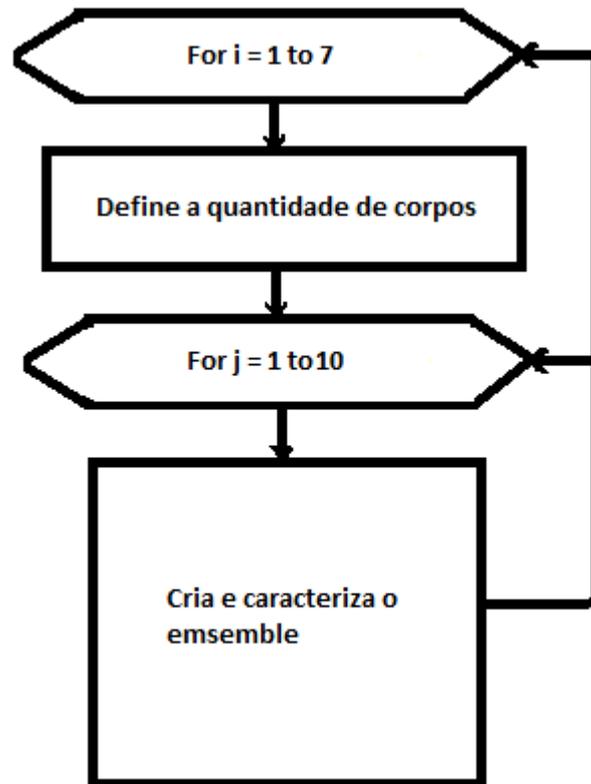


Figura 28 – Algoritmo usado para criar e caracterizar ensembles diminuindo o número de corpos com o tamanho do campo e dos corpos fixo.

3.4.3. Construindo mapas de Cov_{dmean}

Foi construído o mapa de Cov_{dmean} da seguinte imagem (Figura 29).

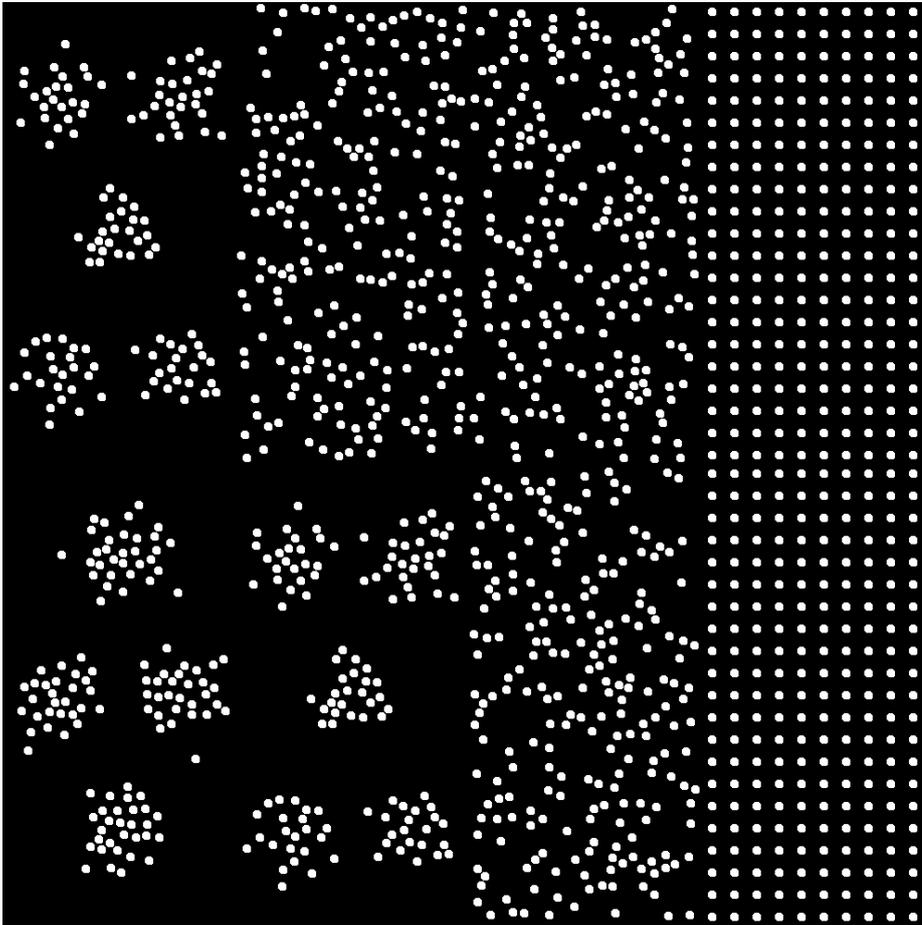


Figura 29 – Imagem para a qual foi construído o mapa de Cov_{dmean}

3.5. Algoritmo e metodologia para a implementação de ensaios virtuais.

Foi usado o software OOF2 para realizar ensaios virtuais que permitiram estimar o módulo de Young transversal de peças virtuais e reais de compósitos com fibras infinitas e paralelas. Logo a seguir, a descrição do procedimento.

3.5.1. Materiais

Foram estudadas 4 peças virtuais. As peças foram concebidas como vigas infinitas de seção transversal retangular. A configuração espacial das fibras nesta seção das 3 primeiras peças está representada nos ensembles virtuais (Figura 30 a-c). Para a quarta peça é usada uma imagem real (ensemble real) da seção

transversal de um compósito pultrudado de matriz polimérica e fibras de vidro, contendo também uma fração significativa de carga inerte (Figura 30d). Este quarto caso seria uma peça virtual de um material real que, para efeitos desta tese, será chamado de “peça real”.

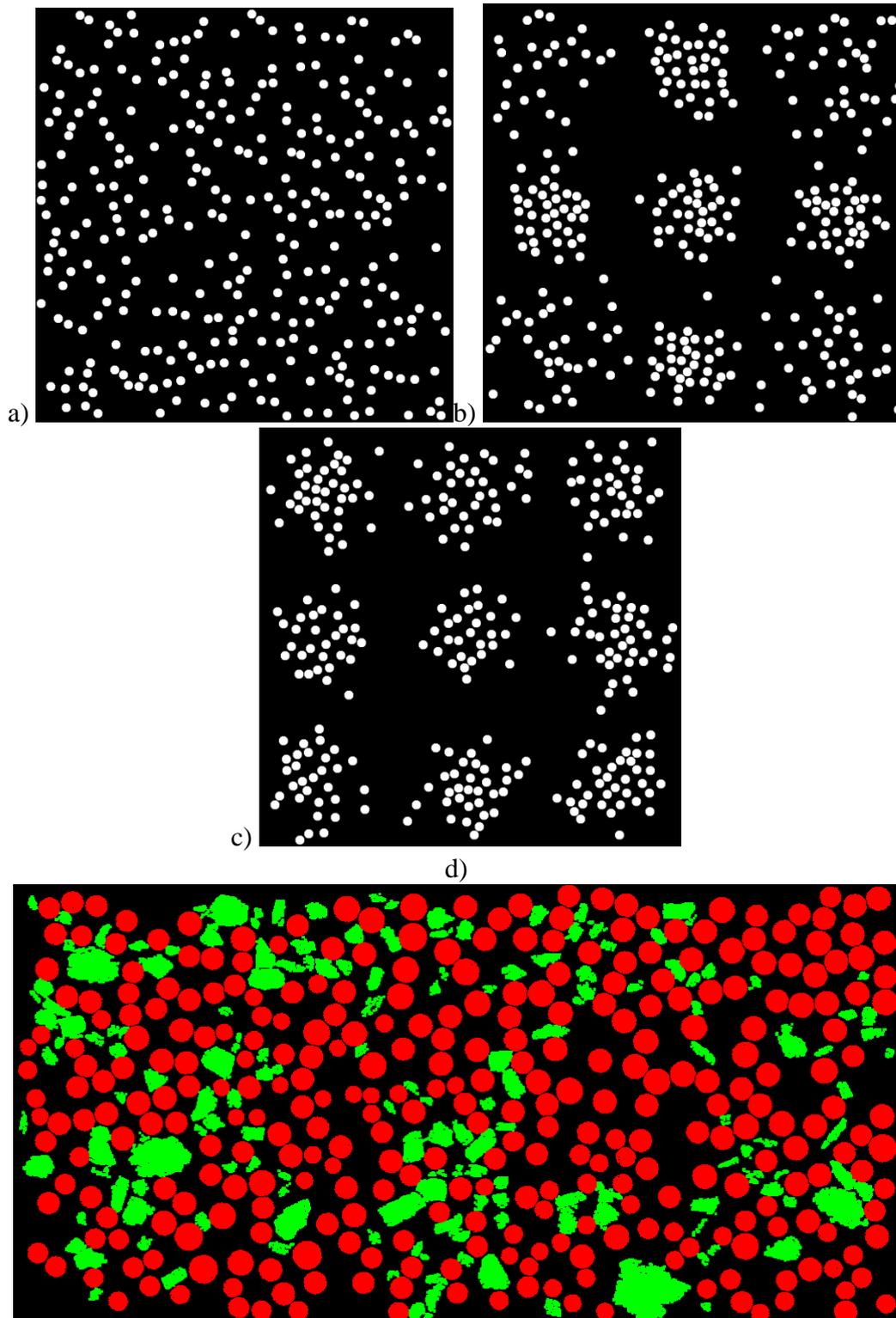


Figura 30 – Ensembles usados no experimento virtual. a-c) Ensembles virtuais d) Ensemble real

No início do experimento o ensemble (imagem) é fornecido ao software para definir a geometria e configuração espacial da peça. Depois é necessário identificar os materiais constituintes, dando valor às propriedades de interesse. Em nosso caso, é de interesse estudar a resposta mecânica da peça, particularmente propriedades elásticas; de forma que é definido o valor de módulo de Young (E) e o coeficiente de Poisson (ν) para cada material constituinte. Os valores usados para estas propriedades estão apresentados na Tabela 2.

Tabela 2 – Parâmetros usados nos ensaios virtuais

	Peças virtuais		Peça real		
	Matriz	Fibra	Matriz	Fibra	Carga
E (GPa)	5	100	3,58	70	0,001
ν	0,2	0,3	0,2	0,2	0,2

Os valores utilizados para a peça virtual foram escolhidos arbitrariamente, apenas considerando dar ao material da fibra um valor para o módulo de Young razoavelmente maior que o dado para a matriz.

3.5.2. Definição do problema físico

O módulo de Young transversal em tração pode ser definido para uma viga de base retangular deformada sob a ação de uma força trativa no eixo “x”, como a tensão na face onde está sendo aplicada a força dividida pela deformação no eixo “x”. Considerando que o material se encontra sempre em regime elástico [1] (Figura 31).

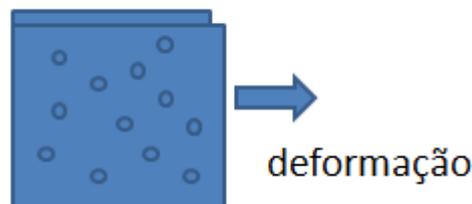


Figura 31 – Seção transversal de uma viga de compósito reforçada por fibras, deformada na direção x.

Usando esta definição, o ensaio virtual consiste precisamente em aplicar uma deformação transversal na peça e medir a tensão na face lateral perpendicular

à direção de deformação. O valor de deformação usado foi de 0.01, imposto na face lateral direita. As posições das faces, inferior e esquerda, foram fixadas nas condições de fronteira para ficarem imóveis, enquanto que a posição da face superior foi deixada livre (Figura 31).

3.5.3. Aplicação do método de elementos finitos

Uma vez definida a estrutura do material e o problema físico a resolver, segue-se a parte de aplicar a teoria de elementos finitos para a resolução do problema desejado. A primeira etapa nesse sentido é a construção do esqueleto.

Primeiro é criado um esqueleto básico de elementos quadrados, com as dimensões dos elementos do mesmo tamanho das partículas da imagem associada. Para uma imagem de dimensões $D \times D$ com partículas de raio R , o esqueleto básico associado seria de $Q \times Q$ elementos onde $Q = D/2R$.

Em seguida este esqueleto básico é refinado aplicando a seqüência de passos a seguir, que utiliza a nomenclatura original utilizada pelo programa OOF2:

- 1- Refine (Homogeneity 1, alpha 0,5)
Subdivide elementos do esqueleto em elementos da mesma geometria
- 2- Refine (Homogeneity 0,9, alpha 0,5)
- 3- Snap Nodes (Homogeneity 0,9, alpha 1)
Desloca nodos do esqueleto ate os pixels da fronteira ou interface dos materiais
- 4- Rationalize (alpha 0,8)
Corrige elementos com formas não desejadas (ângulos extremos, etc.)
- 5- Split squad (Homogeneity 0,9, alpha 0,9)
Divide elementos quadriláteros em dois triângulos
- 6- Rationalize (alpha 0,8)
- 7- Anneal (Homogeneity 0,9, alpha 0,9, Conditional iterations)
Desloca os nós do esqueleto randomicamente aceitando as mudanças que cumpram a condição estabelecida
- 8- Rationalize (alpha 0,8)
- 9- Anneal (Homogeneity 0,9, alpha 0,9, Conditional iterations)
- 10- Rationalize (alpha 0,8)
- 11- Pin node (Pin internal boundary nodes)
Fixa a posição dos nós que estão na interface entre os materiais

12- Swap Edges (all elements, alpha 0,9)

Reajusta as fronteiras internas em pares de elementos vizinhos

13- Rationalize (alpha 0,8)

14- Merge triangles (all elements, Criterion: average limited energy [alpha 0.5, hom. 0.9, shape 0.4])

Une triângulos vizinhos homogêneos para formar quadriláteros

15- Smooth (all nodes, Criterion: average limited energy [alpha 0.5, hom. 0.9, shape 0.4])

Desloca os nós até a posição média dos vizinhos mais próximos aceitando aquelas mudanças que cumpram as condições estabelecidas

Antes de proceder a expor a idéia associada ao algoritmo linear de passos enumerados é bom lembrar, da seção 2.6, que no processo de refinamento do esqueleto se procura o melhor compromisso entre os seguintes parâmetros:

- A) Índice de homogeneidade: os elementos da malha devem ser homogêneos com respeito ao material contido naquela área.
- B) Formas dos elementos: a forma dos elementos devem ser as menos “complicadas” possíveis (evitar ângulos abruptos, etc.)
- C) Número de elementos: é bom não incrementar em demasia o número de elementos para não sobrecarregar a memória do computador e afetar o desempenho do mesmo.

Atendendo a estas considerações, uma primeira etapa (passos 1-10) se foca em conseguir o maior valor de homogeneidade possível, realizando várias operações que dividem e deformam os elementos (melhorar o parâmetro “A” às custas dos parâmetros B e C). A operação de “Rationalize”, reiteradamente usada, atenua o custo das restantes operações da etapa no que se refere à produção de elementos com formas não desejadas. Logo a seguir, no passo 11 se fixam os nós que ficam na interface dos materiais. Mais dois passos (12, 13) orientados ao mesmo objetivo dos primeiros dez; e já está pronta para iniciar a segunda etapa (passos 14-15) que tem como objetivo recuperar parte do sacrificado com respeito aos parâmetros B e C para obter o máximo de homogeneidade na primeira etapa. Para isto as operações destes passos estão orientadas a unir corpos (diminuir a quantidade de elementos) e a suavizar as formas dos objetos, isto sem alterar o valor de homogeneidade atingido.

Uma descrição detalhada de todas as funções enumeradas com seus respectivos parâmetros encontra-se nas referências [56, 57].

Depois de refinado o esqueleto, o tipo de malha escolhido foi de ordem 2 (vide a Seção 2.6) para todos os casos.

Após este último passo, é realizada a simulação e são analisados os resultados.