

X ENCONTRO ACADÊMICO

MODELAGEM COMPUTACIONAL

SIMULAÇÃO ORIENTADA A OBJETOS DA NUCLEAÇÃO E CRESCIMENTO VIA AUTÔMATO CELULAR

Jonathas Luis G. Ferreira

Weslley Luiz S. Assis

Tiago A. Neves

Guilherme C. de Oliveira

jonathasferreira@id.uff.br

weslleyassis@id.uff.br

tneves@id.uff.br

guilherme.ori@gmail.com

Universidade Federal Fluminense - EEIMVR

Av. dos Trabalhadores, No. 420, CEP 27140-360, Vila Santa Cecília, Volta Redonda, RJ, Brasil

Resumo. A evolução microestrutural em três dimensões de transformações que envolvem nucleação e crescimento em materiais cristalinos é simulada pelo método do Autômato Celular com vizinhanças de Von Neumann e condições de contorno periódicas. É assumido que toda a nucleação ocorre no começo da transformação (saturação de sítios). O programa usa o conceito de orientação a objetos que torna possível o melhor aproveitamento do código e facilita futuras melhorias. São construídas classes que simulam os processos de transformação e que controlam a entrada e saída de dados. Este trabalho está sendo comparado com as teorias analíticas e resultados da simulação de outros autores. O programa gera dados que podem ser usados para a visualização tridimensional de grãos e, também, para a comparação com equações tradicionais no estudo destas transformações. Os resultados obtidos têm sido coerentes com o atual entendimento destes tipos de fenômenos. É visto como a programação orientada a objetos permite a facilização na implementação das simulações de nucleação e crescimento e na adaptação do código a futuros implementos.

Palavras-chave: autômato celular, microestrutura, cinética, orientação a objetos

1 INTRODUÇÃO

As transformações que envolvem processos de nucleação e crescimento são muito comuns em vários fenômenos da Natureza como transformações de fase e processos de recristalização em materiais cristalinos. Nesses fenômenos, ocorre o surgimento de partículas muito pequenas (núcleos) que crescem até completar a transformação.

Rios e Padilha (2007) observaram que o mecanismo mais comum de surgimento de uma nova fase envolve o rearranjo atômico por meio do qual uma fase mais estável (β) se forma e se desenvolve na fase matriz (α). No entanto, na recristalização não há flutuações atômicas, mas sim uma migração de contorno de grãos ou subgrãos, ou ainda, um coalescimento de subgrãos.

Enquanto a nucleação está relacionada à formação de uma interface entre a matriz e a nova fase, o crescimento diz respeito à migração dessa interface. A migração, por sua vez, depende de uma força motriz e de um mecanismo de migração. Todavia, o estudo termodinâmico desses processos, que busca responder ao porquê ocorre uma transformação, será omitido neste trabalho, sendo apenas considerados os aspectos geométricos e as características cinéticas da transformação.

A teoria clássica de Johnson, Mehl, Avrami e Kolmogorov (JMAK) (citada por Hesselbarth e Göbel, 1991) descreve o crescimento de núcleos. Para o crescimento tridimensional isotrópico de grãos esféricos, a teoria de JMAK leva à seguinte equação para o grau de recristalização (ou fração volumétrica transformada) V_V :

$$\ln \frac{1}{1 - V_V} = \frac{\pi}{3} \dot{N} G^3 t^4 \quad (1)$$

Nesta equação, \dot{N} é a taxa de nucleação, considerada constante; G é a velocidade de crescimento de grãos, também definida como constante; e t é o tempo.

O tratamento analítico da nucleação e do crescimento de novas fases pode se tornar extremamente complexo devido a várias características que devem ser consideradas e que põem em dúvida algumas aproximações que geralmente são feitas nesses casos. Sendo assim, a simulação computacional é uma ferramenta de grande valor pois pode reproduzir situações que vão além do tratamento analítico.

Dentre os modelos usados para simulações, o método do autômato celular tem se mostrado adequado para simulação de transformações de fase em geral e de recristalização (Rios et al., 2006). De modo contrário da abordagem de JMAK, o modelo do autômato celular fornece uma descrição desses tipos de transformações que é dependente do tempo e do espaço (Hesselbarth, Göbel, 1991). Tal modelo não está restrito à premissas como distribuição homogênea dos núcleos ou valores estatísticos médios.

Autômatos Celulares têm a vantagem de serem modelos simples para descreverem fenômenos físicos. Processos dependentes do tempo e do espaço são geralmente descritos por equações diferenciais parciais. Todavia, tais equações, na maioria dos casos, precisam ser resolvidas por meio de métodos numéricos. Autômatos Celulares, por sua vez, já são um algoritmo numérico e, portanto, permitem uma análise mais imediata e ampla, além de permitirem a variação de muitas premissas e parâmetros que, muitas vezes, não podem ser trabalhados por meio de equações (Hesselbarth, Göbel, 1991).

Tradicionalmente, modelos de Autômatos Celulares vêm sendo implementados por meio de programação estruturada. Apesar desses programas se mostrarem muito úteis e eficientes para o estudo de transformações de fase, eles possuem as deficiências comuns aos códigos construídos sob o paradigma procedural: a dificuldade de reutilização do código e os problemas de construção e adaptação do código por meio de trabalho em equipe (Deitel, Deitel, 2001).

Um paradigma de programação que tem sido usado para transpor esses problemas é a orientação a objetos. Objetos são partes reutilizáveis de um programa que modelam o mundo real. Tem-se percebido que a utilização de uma abordagem de implementação de códigos orientados a objetos torna os grupos de desenvolvimento mais produtivos do que quando utilizadas somente técnicas de programação estruturada. A orientação a objetos permite a criação de programas mais fáceis de entender, corrigir e modificar (Deitel, Deitel, 2001).

O propósito deste trabalho é apresentar o desenvolvimento de um programa escrito segundo o paradigma de orientação a objetos que utiliza o método do autômato celular para o estudo das transformações de fase. Esse desenvolvimento é motivado pela busca da melhoria no trabalho em conjunto e na produtividade do grupo de estudos do Núcleo de Modelamento Microestrutural da Universidade Federal Fluminense.

A princípio, é assumido que toda a nucleação ocorre no início da transformação e é utilizado o tipo de vizinhanças de Von Neumann. Os dados resultantes das simulações são comparados com as previsões de equações analíticas e com o trabalho de Rios et al. (2006).

2 Desenvolvimento do Código

O programa é formado por classes que buscam simular o comportamento de objetos que representam um material que passará por um processo de nucleação e crescimento. As principais classes são: Dados, Matriz, Nucleação, Crescimento, Autômato e Saída.

Foi utilizada a linguagem C++ por suportar a orientação a objetos e ser uma linguagem amplamente utilizada. C++ possui vasta documentação e elevada contribuição de usuários de todo o mundo.

2.1 A classe “Dados”

A classe “Dados” é responsável pela coleta dos parâmetros que definirão as transformações durante a execução. Optou-se pela utilização de um arquivo de entrada padronizado que contém os parâmetros necessários ao programa. Realiza-se, então, a criação de um objeto que armazena os dados e os torna acessíveis a qualquer momento sem a necessidade de releitura do arquivo de entrada. Os principais dados armazenados são: tipo e tamanho da malha, tipo de nucleação, número de núcleos iniciais, tipo de vizinhança e critérios de parada do código.

2.2 A classe “Matriz”

Esta classe é a responsável pela criação da malha onde serão realizadas as transformações. A princípio, foi implementada uma malha tridimensional do tipo cúbica.

Como forma de melhoria de desempenho, a malha é construída em um vetor unidimensional ao invés de usar uma matriz tridimensional. A dimensão deste vetor é igual ao número

de elementos da matriz tridimensional equivalente. Embora essa escolha implique em algoritmos mais sofisticados para realização de algumas tarefas apresentadas a seguir, os ganhos com desempenho, inclusive relacionados à paralelização, a tornam justificável.

O método do autômato celular trabalha com a ideia de estados transformados na matriz. Sendo assim, dois vetores são construídos: o primeiro para representar os estados da malha antes dos passes de transformação e o segundo para armazenar as transformações durante o passe. No instante inicial, ambos os vetores possuem todos os valores iguais a zero, representando a fase não transformada.

Além disso, a classe “Matriz” contém os métodos relacionados à medições como fração volumétrica transformada, V_V , e área interfacial entre região transformada e não transformada por unidade de volume, S_V .

2.3 A classe “Nucleação”

A classe “Nucleação” altera o vetor de estado conforme o tipo de nucleação e o número de núcleos iniciais declarados na entrada do programa. Inicialmente foi implementado um algoritmo para simular a nucleação aleatória, homogeneamente distribuída pela malha e que ocorre totalmente no início da transformação (saturação de sítios).

Para a distribuição dos núcleos, foi usada a geração de números pseudoaleatórios com uma distribuição uniforme disponível na biblioteca “Random” de C++. Conforme as posições no vetor de estado são sorteadas, estas posições recebem um valor sequencial de 1 até n , onde n é o número total de núcleos. Se alguma posição onde haja um núcleo for novamente escolhida, o sorteio é repetido.

2.4 As classes “Crescimento” e “Autômato”

A classe “Crescimento” determina o tipo de crescimento que os núcleos terão na malha. Neste trabalho, foi implementada a classe “Autômato” para o crescimento por meio do método do autômato celular.

Um autômato celular tem as seguintes características (Hesselbarth, Göbel, 1991):

1. A geometria das células.
2. Os estados que uma célula pode possuir.
3. A definição da vizinhança da célula.
4. As regras de transição que determinam o estado das células no próximo passo com base em seu estado atual e no estado atual das células vizinhas.

Conforme já mencionado, a geometria das células implementada é a cúbica. Os estados que a célula pode possuir são o estado não transformado, representado pelo valor zero, e o estado transformado, representado por valores de 1 até n , sendo n o número total de núcleos. Cada núcleo possui um valor e esse valor é propagado para a vizinhança conforme ocorre o crescimento.

A vizinhança implementada foi a de Von Neumann que é composta pelas células imediatamente próximas nas direções axiais. Para simular uma amostra infinita, são usadas condições

de contorno periódicas, isto é, no caso de uma célula que se encontre em um dos planos limites da malha, é considerada sua vizinha a célula pertencente ao plano oposto na mesma direção.

A regra de transição é uma verificação do estado atual da célula e de suas vizinhas. Se a célula já estiver transformada, a verificação passa para a próxima. Caso contrário, verifica-se se alguma das células vizinhas está transformada e, se estiver, a célula analisada recebe o valor da vizinha transformada.

Como é possível haver mais de uma célula vizinha transformada, a ordem de checagem das vizinhas é aleatória e a primeira vizinha transformada encontrada é a escolhida. Isso evita que haja uma orientação preferencial quando duas regiões transformadas se encontram e suas fronteiras tendem, portanto, a serem próximas a planos.

Em cada intervalo de tempo, toda a malha é atualizada conforme a regra de transição. O vetor de estados que armazenava os valores de antes do passe recebe os valores do vetor que registrou as transformações e o processo se repete. O crescimento para quando algum valor preestabelecido de fração volumétrica transformada ou número de passes é atingido. Durante as iterações, os valores de V_V e S_V são calculados e registrados.

2.5 A classe “Saída”

Esta última classe é responsável pela saída dos dados gerados pelo programa em forma de arquivos de texto. O programa gera como saída os estados da malha no final da transformação ou em alguma etapa e os valores de V_V e S_V ao longo dos passes. Esses dados podem ser utilizados em algum programa gráfico.

3 Resultados e Discussão

O programa gera como saída valores de estado para cada posição na malha. Como exemplo, foi simulada uma transformação em uma malha de dimensão $100 \times 100 \times 100$ com 100 núcleos iniciais. O código foi executado até que 100% da malha fosse transformada. Foram realizadas 43 iterações para que o processo se concluísse.

A Figura (1) apresenta a construção gráfica desta malha e a evolução dos grãos durante o crescimento. A coloração é aleatória. Pode-se perceber a presença de grãos poligonais e equiaxiais típicos de uma estrutura recristalizada. Os grãos crescem e se chocam gerando contornos de grão que tendem a ser retos ou serrilhados. Esse aspecto deve-se à aleatoriedade na escolha dos vizinhos durante a transformação das células. Além disso, pode-se perceber que os grãos no início da transformação têm a forma octaédrica, característica do método do autômato celular. Também pode-se notar que, durante o crescimento, o uso de condições de contorno periódicas proporciona que grãos que crescem próximo aos limites da malha continuem crescendo na extremidade oposta.

O programa também gera como saída valores de fração volumétrica transformada, V_V , e área interfacial entre região transformada e não transformada por unidade de volume, S_V , ao longo das iterações que podem ser comparados com previsões analíticas. Cada iteração é considerada como uma unidade de tempo. Todas as unidades serão omitidas.

Rios et al. (2006) demonstraram que, para a nucleação do tipo de saturação de sítios, a teoria de JMAK leva a uma expressão, dada pela Eq. (2), para o cálculo da fração volumétrica, V_V , em função do número de núcleos por unidade de volume, N_V , e do tempo, t .

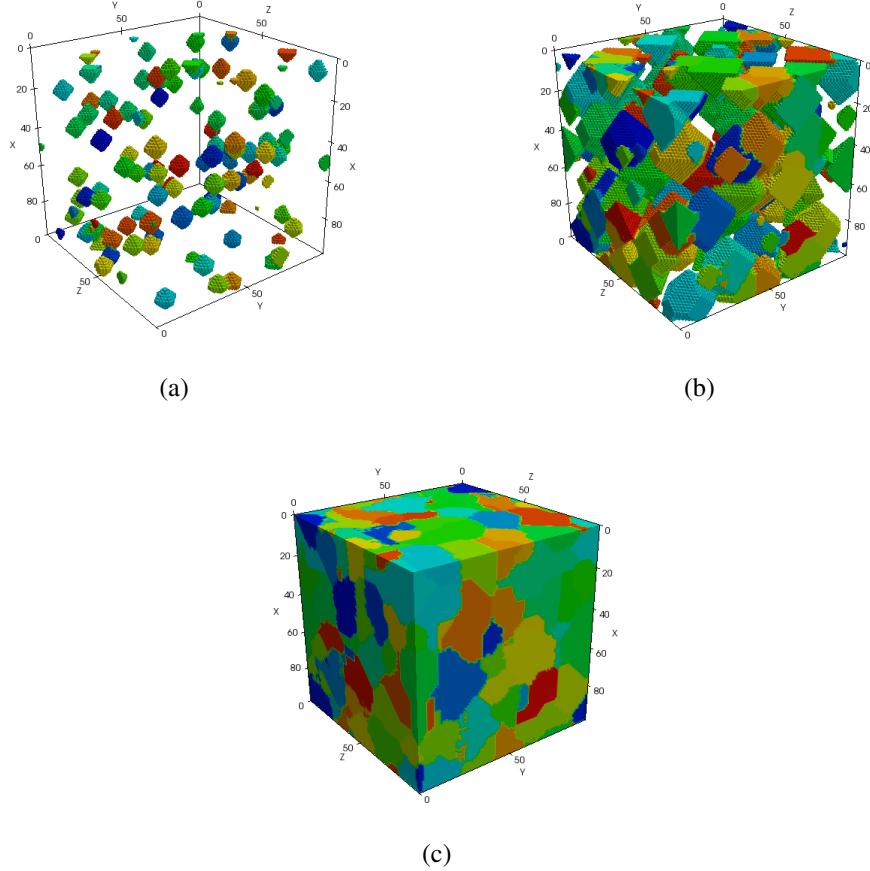


Figura 1: Estrutura de grãos gerada pela simulação computacional. (a) 5%, (b) 50% e (c) 100% de transformação.

$$V_V = 1 - \exp\left(-\frac{4}{3}N_V t^3\right) \quad (2)$$

Além disso, Rios et al. (2006) também obtiveram uma expressão, dada pela Eq. (3), para o cálculo da área interfacial entre região transformada e não transformada por unidade de volume, S_V , em função do número de núcleos por unidade de volume, N_V , e do tempo, t .

$$S_V = 12N_V t^2 \exp\left(-\frac{4}{3}N_V t^3\right) \quad (3)$$

Desta forma, a Figura (2) apresenta a comparação entre a V_V e a S_V calculadas pela Eq. (2) e pela Eq. (3) e as obtidas pela simulação do autômato celular.

Pode-se perceber que há bom acordo entre os valores da Eq. (2) e da Eq. (3) e os valores da simulação. É visto que o crescimento pode ser dividido em três partes principais: um período inicial de lenta transformação, seguido por um onde a taxa de crescimento é consideravelmente mais alta, terminando em um período onde a taxa de transformação volta a ser lenta até terminar.

O algoritmo utiliza valores pseudoaleatórios na nucleação e na transformação de regiões onde serão os contornos de grão. Uma não uniformidade nesses valores pode ser o fator respon-

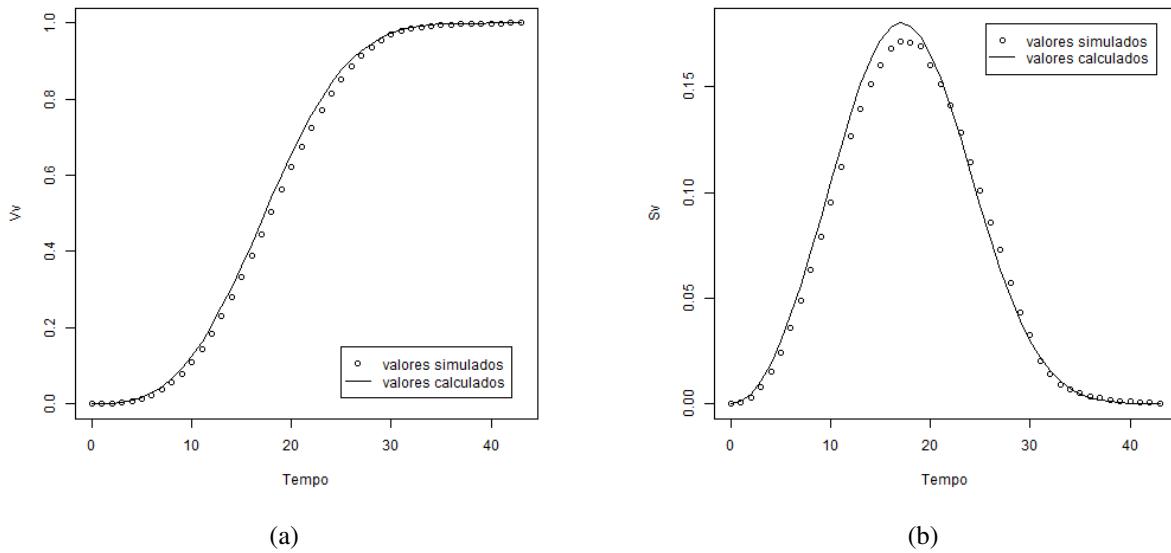


Figura 2: Comparação entre valores calculados analiticamente e obtidos na simulação: a) Fração volumétrica, V_V , em função do tempo e b) área interfacial entre região transformada e não transformada por unidade de volume, S_V , em função do tempo.

sável pela pequena discrepância observada entre os valores calculados e simulados da Figura (2).

Uma outra forma de analisar o crescimento de grãos é chamada de caminho microestrutural e consiste em escrever S_V em função de V_V . Rios et al. (2006) utilizaram a Eq. (4) para isso.

$$S_V = 3(36N_V)^{\frac{1}{3}}(1 - V_V) \left[\ln \left(\frac{1}{1 - V_V} \right) \right]^{\frac{2}{3}} \quad (4)$$

Assim, a Figura (3) apresenta a comparação entre o caminho microestrutural calculado e o simulado.

Há bom acordo entre os valores da Eq. (4) e os valores da simulação. Os valores simulados aparecem um pouco abaixo da curva calculada devido à interação entre os grãos no início do crescimento, diminuindo a quantidade de área interfacial entre as regiões recristalizadas e não recristalizadas (S_V) no decorrer da transformação (Oliveira et al., 2006).

Tanto os resultados apresentados na Figura (2) e na Figura (3) estão em conformidade com os obtidos por Rios et al. (2006). Como mencionado por esses autores, o bom acordo entre os valores calculados e simulados mostram que o método do Autômato Celular é confiável para este tipo de transformação.

4 Conclusão

A programação orientada a objetos se mostrou bastante adequada na implementação de um algoritmo de autômato celular para simulação de fenômenos de nucleação e crescimento. O programa gerou resultados coerentes com as previsões analíticas e com trabalhos de outros autores.

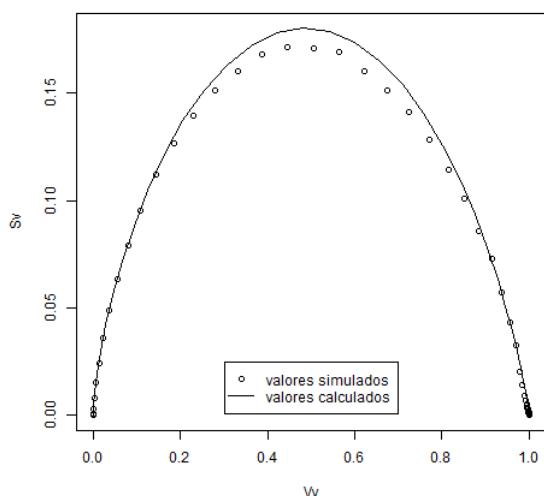


Figura 3: Caminho microestrutural: Área interfacial entre regiões transformada e não transformada por unidade de volume, S_V , contra Fração volumétrica, V_V .

O uso do paradigma de orientação a objetos permite que novas funcionalidades sejam incorporadas ao programa de forma bem mais simples que o convencional. A simulação de outros tipos de nucleação, geometrias das malhas e vizinhanças das células é pretendida para trabalhos futuros. A adaptabilidade do programa ao recebimento de novas funções tem sido um dos objetivos principais desde sua criação.

Agradecimentos

Os autores agradecem o apoio da Pró-Reitoria de Assuntos Estudantis (PROAES) da Universidade Federal Fluminense, do Mestrado em Modelagem Computacional em Ciência e Tecnologia e do Núcleo de Modelamento Microestrutural.

REFERÊNCIAS

- Deitel, H., & Deitel, P., 2001. *C++ Como Programar, 3 ed..* Bookman.
- Hesselbarth, H., & Göbel, I., 1991. Simulation of Recrystallization by Cellular Automata. *Acta Metallurgica et Materialia*, vol. 39, n. 9, pp. 2135 à 2143.
- Oliveira, V., Oliveira, L., Rios, P., & Castro, J., 2006. Desenvolvimento de um Código em 3D para Simular a Recristalização pelo Método do Autômato Celular. *Tecnologia em Metalurgia e Materiais*, vol. 2, n. 4, pp. 34 à 39.
- Rios, P., Oliveira, V., Pereira, L., Pereira, M., & Castro, J., 2006. Cellular Automata Simulation of Site-saturated and Constant Nucleation Rate Transformations in Three Dimensions. *Materials Research*, vol. 9, n. 2, pp. 223 à 230.
- Rios, P., & Padilha, A., 2007. *Transformações de Fase.* Artliber.