

SANDIA REPORT

SAND97-1925 • UC--404

Unlimited Release

Printed August 1997

Potts-model Grain Growth Simulations: Parallel Algorithms and Applications

Steven A. Wright, Steven J. Plimpton, Thomas P. Swiler, Richard M. Fye,
Michael F. Young, Elizabeth A. Holm

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; distribution is unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND 97-1925
Unlimited Release
Printed August 1997

Potts-model Grain Growth Simulations: Parallel Algorithms and Applications

Steven A. Wright
Nuclear Technology & Research Department

Steven J. Plimpton
Parallel Computational Sciences Department

Thomas P. Swiler
Materials Processing Department

Richard M. Fye
Computational Materials Science Department

Michael F. Young
Modeling & Analysis Department

Elizabeth A. Holm
Theoretical & Computational Materials Modeling Department

Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-1146

Abstract Follows

Abstract

Microstructural morphology and grain boundary properties often control the service properties of engineered materials. This report uses the Potts-model to simulate the development of microstructures in realistic materials. Three areas of microstructural morphology simulations were studied. They include the development of massively parallel algorithms for Potts-model grain grow simulations, modeling of mass transport via diffusion in these simulated microstructures, and the development of a gradient-dependent Hamiltonian to simulate columnar grain growth.

Potts grain growth models for massively parallel supercomputers were developed for the conventional (non-accelerated) Potts-model in both two and three dimensions. Simulations using these parallel codes showed self similar grain growth and no finite size effects for previously unapproachable large scale problems. In addition, new enhancements to the conventional Metropolis algorithm used in the Potts-model were developed to accelerate the calculations. These techniques enable both the sequential and parallel algorithms to run faster and use essentially an "infinite" number of grain orientation values to avoid non-physical grain coalescence events.

Mass transport phenomena in polycrystalline materials were studied in two dimensions using numerical diffusion techniques on microstructures generated using the Potts-model. The results of the mass transport modeling showed excellent quantitative agreement with one dimensional diffusion problems, however the results also suggest that transient multi-dimension diffusion effects cannot be parameterized as the product of the grain boundary diffusion coefficient and the grain boundary width. Instead, both properties are required.

Gradient-dependent grain growth mechanisms were included in the Potts-model by adding an extra term to the Hamiltonian (total internal energy). Under normal grain growth, the primary driving term is the curvature of the grain boundary, which is included in the standard Potts-model Hamiltonian. To investigate columnar grain growth, a gradient-dependent term was added. This approach was taken because it fit easily into the mobility model in which the velocity of the grain boundary is proportional to its mobility times the driving forces. Results of these simulations produced the expected columnar grain structures in regions of high mobility and large temperature gradients.

Table of Contents

1. INTRODUCTION	8
1.1. SUMMARY OF MASSIVELY PARALLEL POTTS-MODELS	8
1.2. SUMMARY OF MASS TRANSPORT PHENOMENA IN POLYCRYSTALLINE MATERIALS.....	9
1.3. SUMMARY OF GRADIENT DEPENDENT POTTS-MODEL SIMULATIONS.....	9
2. POTTS-MODEL.....	11
3. PARALLEL POTTS-MODEL SIMULATIONS	13
3.1. CONVENTIONAL ALGORITHM IN PARALLEL.....	13
3.2. LARGE-SCALE PARALLEL POTTS-MODEL SIMULATIONS.....	17
3.3. NEW ALGORITHMS.....	20
3.3.1. <i>Drawbacks of Conventional and "N-Fold Way" Algorithms</i>	20
3.3.1.1. Conventional Algorithm.....	20
3.3.1.2. N-Fold Way Algorithm	21
3.3.2. <i>Fast Parallel Algorithms</i>	21
3.3.2.1. Algorithmic Goal.....	21
3.3.2.2. Masking.....	22
3.3.2.3. "Wild" moves	22
3.3.2.4. Fixed Neighbor Approximation	23
3.3.2.5. Infinite Q.....	25
3.3.2.6. Timing.....	26
4. MASS TRANSPORT IN POLYCRYSTALLINE MATERIALS.....	27
4.1. INTRODUCTION.....	27
4.2. BACKGROUND.....	27
4.3. METHOD	29
4.4. VALIDATION.....	31
4.5. STEADY STATE DIFFUSION.....	33
4.6. TRANSIENT DIFFUSION PROFILES	35
4.7. CONCLUSIONS.....	36
5. GRADIENT DEPENDENT POTTS-MODELING SIMULATIONS.....	38
5.1. BACKGROUND.....	38
5.2. MOBILITY AND BIASED MIGRATION	38
5.2.1. <i>Grain Boundary Velocities</i>	40
5.2.2. <i>Migration in a Temperature Gradient via Vapor-transport</i>	42
5.3. BIASED MIGRATION USING THE N-FOLD ALGORITHM.....	44
5.4. CONCLUSION OF BIASED MIGRATION POTTS-MODEL SIMULATIONS.....	44

Table of Figures

Figure 1: Example of grain growth at two different times in a 200 x 200 lattice with Mobility Ratio $M = 1$.
 Figure A is at an early time of 5,000 Monte Carlo time steps, and figure B is at 50,000 time steps. The average grain radii are 8.4 and 26.1 pixels respectively. 12

Figure 2. Checkerboard lattice used for parallel Potts grain growth algorithm. 14

Figure 3. Comparison of parallel grain growth algorithm versus sequential algorithm for a 200x200 lattice out to 10^5 Monte Carlo time steps. The Open symbols are from the parallel algorithm. The lines are from the sequential simulation. 16

Figure 4. Two dimensional parallel algorithm grain grow study to examine self similarity and finite size effects for lattice sizes from 100x100 to 4000x4000. Grain areas are vertically offset and show statistically identical growth behavior independent of the lattice size. 17

Figure 5. Three dimensional parallel Potts algorithm to study self similarity and finite size effects for varying lattice sizes. The three dimensional lattices vary from 30x30x30 lattice sites to 300x300x300. Grain areas are vertically offset for each lattice size. Statistical deviations occur for the smaller lattices because only a few grains remain after many time sweeps. 18

Figure 6. Second, third, and fourth order moments of the 2-D large lattice size grain growth simulations. 19

Figure 7. Local lattice configuration for selection of a spin flip. 20

Figure 8. Local lattice spin values for a grain interior. 21

Figure 9. Example of “wild” spin move, where central spin (2) changed to (5) with no energy change. 22

Figure 10. Parallel Potts simulation with and without allowing “wild” spin moves. 23

Figure 11. Comparison of Parallel Potts simulation using the Fixed Neighbor Approximations (FNA) to the conventional algorithm. 25

Figure 12. *The method by which a bitmapped microstructure is mapped to a diffusion grid. Note that diffusivities along grain boundaries correspond to the grain boundary coefficient, diffusivities across grain boundaries correspond to the bulk coefficient.* 30

Figure 13. Comparison of finite difference calculation results (lines) to analytic results (points) for (a) diffusion from a thin-film source at $x = 200$ and (b) diffusion in a heterogeneous diffusion couple with an interface at $x = 500.5$ 31

Figure 14. Comparison of finite difference calculation results (lines) to analytic results (points) for diffusion in a finite open slab in which direction of diffusion does not correspond with lattice directions. 32

Figure 15. Geometry of diffusion problem solved by Gilmer and Farrell.¹⁵ 32

Figure 16. Comparison of finite difference calculation results to semi-analytic results reported by Gilmer and Farrell¹⁵ for the solution of diffusion into a finite slab of thickness $X_0 = 200$ units thick with equally spaced grain boundaries normal to the surface and a ratio of boundary to bulk diffusivity $D_b/D_g = 1000$. Gilmer and Farrell’s results use a boundary separation $L = 200$ units and a grain boundary width $\delta = 1$. The finite difference results use $L = 400$ and $\delta = 2$. Dashed lines and open symbols correspond to the case of diffusion into a film with a diffusion barrier on the back side. Solid lines and closed symbols correspond to the case of diffusion into a film with a high diffusivity surface on the back side. 33

Figure 17. Effective diffusion coefficient D_{eff} as a function of average grain size d for simulated microstructures with $D_b = 10^{-1}$ and $D_g = 10^{-6}$. The rule of mixing coefficient $D_{eff} = D_b \delta / d$ and the corrected rule of mixing coefficient $D_{eff} = 0.27 D_b \delta / d$ are shown for comparison. 34

Figure 18. Solute concentration profile in a two-dimensional microstructure. An infinite source ($C = 1.0$) is at $x = 0$, an infinite sink ($C = 0.0$) is at $x = 100$, and the boundaries $y = 0$ and $y = 100$ are periodic. The grain structure is shown on the base plane of the plot. 35

Figure 19. Solute penetration into two-dimensional polycrystalline microstructures plotted in the functional form suggested by Levine and MacCallum.¹⁶ $D_g = 10^{-6}$, $D_b = 10^{-1}$, (a) $d = 5$, (b) $d = 10$ 36

Figure 20: Simulation mobility profile used to bias grain growth to large grains in the higher mobility regions. 41

Figure 21: Grain growth with spatially dependent mobility profile. Observe the large equiax grains on the high (hot) mobility side and the small equiax grains on the low (cold) mobility side. 41

Figure 22: Gradient term modifying the Hamiltonian. 43

Figure 23: Gradient dependent grain growth showing columnar grains at high mobility side, equiaxed grains in the middle, and little grain growth on the cold or low mobility side. 43

1. INTRODUCTION

Microstructural morphology and grain boundary properties often control the in-service properties of engineered materials making it essential to understand the development of microstructures in realistic materials. This report used the Monte Carlo Potts-model simulation technique to study grain growth and recrystallization phenomena. The Potts-model simulations were extended to include not only the traditional effects of crystal orientation and grain growth, but also to include new physics effects such as mobility and temperature gradients. In addition, because the protective ability of coatings is strongly controlled by the diffusion of corrosive species through the coating microstructure, the microstructures generated by the Potts-model simulations were used to simulate grain boundary and bulk diffusion of species through these realistic microstructures. Finally, massively parallel algorithms for the Potts-model simulations were developed to provide the capability of performing large scale simulations for macro-sized “real world” problems.

This report first briefly introduces Potts-model in section 2, and follows this introduction in the next three sections with the development of massively parallel algorithms (section 3), mass transport through realistic microstructures (section 4), and the extension of the Potts-model to include temperature gradient effects (section 5). A brief summary of the results obtained in these sections is described below.

1.1. Summary of Massively Parallel Potts-models

A number of Potts grain growth models for massively parallel supercomputers were developed and tested. The parallel algorithms were first developed for the conventional (non-accelerated) Potts-model in both two and three dimensions. The parallel algorithms first developed a lattice coloring scheme to insure that the Monte Carlo energy update rules were correctly followed when multiple processors were simultaneously updating the grain growth. In addition robust parallel random number generators were also developed to insure that the code grows grains that are statistically "identical", independent of the number of processors used, and independent of the initial random number seeds. These newly developed parallel codes were used on Sandia's 1840-processor Intel Paragon to perform large-scale Potts simulations on lattice sizes up to 4000x4000 in 2-D and 300x300x300 in 3-D, each with 100,000 or more lattice sweeps. These simulations were performed to evaluate self similarity of the grain growth and to explore finite size effects for previously unapproachable large scale problems. To our knowledge these are the largest Potts-model simulations performed to date. The results are described in Section 3.2.

In addition to conventional Potts simulation methods, several new enhancements to the conventional Metropolis algorithm used in the Potts-model simulations were

developed. These techniques enable both the sequential and parallel algorithms to run faster and to use essentially "infinite" numbers of spin values (grain orientations), so that coalescence events (when two different grains with the same spin value artificially coalesce) do not occur. These new algorithms are described in Section 3.3.

1.2. Summary of Mass Transport Phenomena in Polycrystalline Materials

Engineering materials exposed to aggressive environments often use coatings for their protective ability. Because the protective ability is strongly controlled by the diffusion of corrosive species through the coating microstructure, the microstructures generated by Potts-model simulations were used to simulate grain boundary and bulk diffusion of species through realistic two-dimensional microstructures. These simulated mass transport phenomena in polycrystalline materials are described in section 4.

Traditionally, such mass transport properties have been studied by examining model or idealized polycrystalline structures, such as a regular array of straight grain boundaries. However, these models do not account for a number of features of real grain ensembles, including the grain size distribution and the topological aspects of grain boundaries. In this study, numerical diffusion simulation techniques were developed to research transient and steady-state mass transport through realistic two-dimensional polycrystalline microstructures. In all cases the microstructures were generated using a single processor Potts-model N-Fold Way code. The effects of microstructural parameters such as average grain size and grain boundary topology were examined, as were the limitations of the model.

The results of the mass transport modeling showed excellent quantitative agreement with one-dimensional diffusion problems; however, the results also suggest that multi-dimensional diffusion and transient properties are more complicated than indicated by simplistic mixing rules. Specifically, transient grain boundary diffusion cannot be parameterized as the product of the grain boundary diffusion coefficient and the grain boundary width. Instead both properties are required. In general, the conventional scaling and mixing rules appear to hold; however, the quantitative results look as though they are dependent on geometric factors such as bottle-necks and tortuosity.

1.3. Summary of Gradient Dependent Potts-model Simulations

Large temperature gradients in ceramic materials at elevated service temperatures often lead to the development of columnar grains. This phenomenon occurs in nuclear reactor fuel pins. To investigate columnar grain growth caused by temperature gradients, a gradient-dependent term was added to the Potts-model Hamiltonian. This approach (of

modifying the Hamiltonian) was used because it fits easily into the mobility model for grain growth in which the velocity of the grain boundary is proportional to its mobility times the driving forces. Under normal grain growth, the governing growth mechanism is curvature, which is included in the standard Potts-model Hamiltonian. The additional driving term (growth mechanism) was added to account for gradient-dependent driving mechanisms such as vapor-transport across pores and voids (or other gradient dependent transport phenomena). Other physical transport mechanisms can also be included in this manner including: surface diffusion, volume diffusion, and migration of pores in stress gradients.

When a gradient term was added to the Hamiltonian having a one-dimensional linear mobility gradient, the grain growth turned to columnar growth at the high mobility (hot) side, equiax in the middle, and little growth at the low mobility (cold) side. Qualitatively, these changes reproduce the microstructures observed in uranium oxide fuels which are caused by temperature gradient-dependent vapor transport mechanism across pores in the ceramic fuel.

2. POTTS-MODEL

The Potts-model simulation method is described in an number of reports^{5,22} and briefly summarized here. In this type of simulation the microstructure of a material is mapped onto a 2-dimensional grid or lattice. At each lattice site an index, S_i , is assigned which corresponds to the orientation of the grain at that site. This index is called the spin. Adjacent sites with different spin orientations form interface sites while sites with only like nearest neighbors are interior sites. The total system energy of the simulation over all sites is specified by assigning a positive energy to interface sites and zero energy to interior sites. It is computed via the Hamiltonian

$$H = \frac{J}{2} \sum_{i=1}^N \sum_{j=1}^z (1 - \delta(S_i, S_j)) \quad (2-1)$$

where the outer sum (i) is overall all sites in the system, and the inner sum (j) is only over the z nearest neighbors. (In a square lattice, there are eight nearest neighbors because the diagonal neighboring sites are included.) The Kronecker delta function is defined as $\delta(S_i, S_j) = 1$ if $S_i = S_j$, and 0 otherwise. Typically in the simulations described in this report $S=100$, thus allowing 100 different types of grains. J is a constant used to scale the grain boundary energy. In essence the system energy is J times the number of interface boundary segments in the system.

Grain growth kinetics are determined by Monte Carlo selection of the site and then deciding whether or not to change the spin at this site based on the total system energy change. This is accomplished first by randomly selecting a lattice site i and a new site spin S_j different from the original spin S_i . The spin index of site i is then changed to the new index (S_j) with probability given by

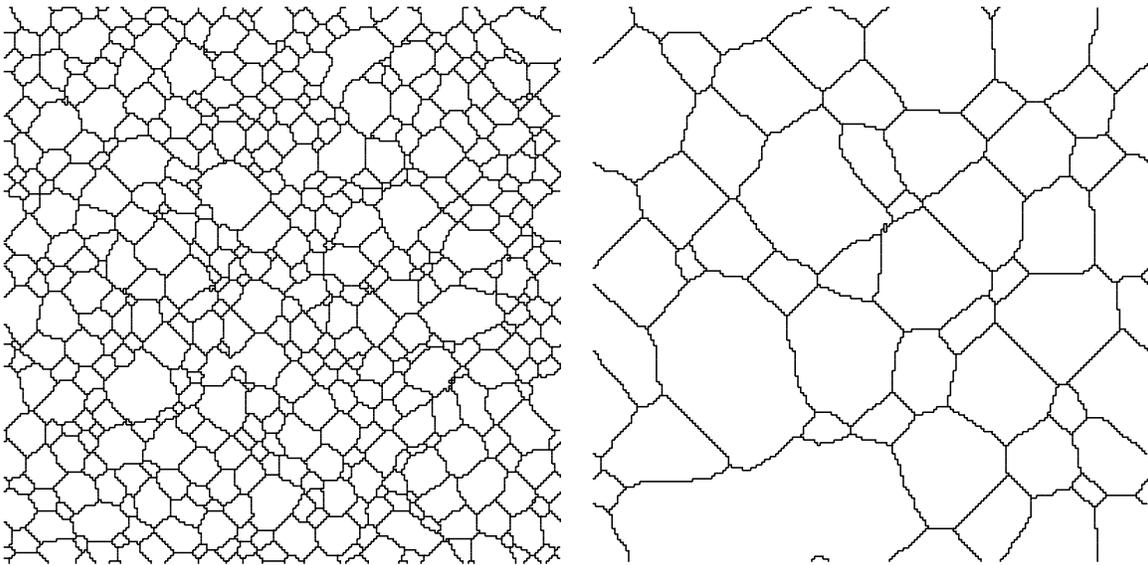
$$P_i(\Delta E) = \begin{cases} 0 & \text{if } \Delta E > 0 \\ M_i & \text{if } \Delta E \leq 0 \end{cases} \quad (2-2)$$

where M_i is a scaling function dependent on the mobility of site i , and ΔE is the energy change for assigning site i the new spin index S_j . ΔE is calculated by calculating the difference in like-spin neighbors around site i when it has the old spin and the new spin. When the number of nearest neighbors has more like spins with the newly selected spin S_j , then ΔE is negative, resulting in a lower value for the Potts-model Hamiltonian. ΔE is just the change in the Hamiltonian due to change in spin of a single site i .

Time is incremented by $1/N$ after each attempted spin flip. N is the total number of lattice sites in the system. Therefore, each Monte Carlo time step requires N attempted spin flips.

Note that the mobility function, $0 < M_i \leq 1$, scales the acceptance rate of the energetically favorable changes. Thus when the mobility function is high the likelihood of changing the spin to one of the neighbors is high and thus the grains grow more quickly when M_i is close to 1. Similarly, the growth rate is slow when M_i is small. Thus, the magnitude of the simulation mobility function M_i is not equal to a physical mobility; rather it represents a ratio of the physical mobility at the fastest and slowest locations within the simulation.

To illustrate the types of grains that are grown by the Potts model simulation, the results of a simple example are depicted in Figure 1. In this example the mobility term M_i was set to 1.0, and the two-dimensional field upon which grains were grown had a dimension of 200×200 lattice sites. Figure 1a shows the grain structure at early times (5,000 Monte Carlo time steps) where the grains had an average radius of 8.4 pixels. Similarly Figure 1b shows the grain structure later in time (at 50,000 Monte Carlo time steps.) after substantial grain growth occurred. It had an average grain radius of 26.1 pixels.



A. (Early time)

B. (Late time)

Figure 1: Example of grain growth at two different times in a 200×200 lattice with Mobility Ratio $M = 1$. Figure A is at an early time of 5,000 Monte Carlo time steps, and figure B is at 50,000 time steps. The average grain radii are 8.4 and 26.1 pixels respectively.

3. PARALLEL POTTS-MODEL SIMULATIONS

This section describes the implementation of the Potts grain growth model on massively parallel supercomputers. First, the basic parallel algorithm was developed for the 2-D and 3-D conventional Potts-model. This involves use of (1) a lattice coloring scheme to insure that the Monte Carlo energy update rules are followed and (2) robust parallel random number generators to insure that the code grows grains in a statistically "identical" way, independent of the number of processors used, and independent on the initial random number seeds. The new parallel codes on Sandia's 1840-processor Intel Paragon were used to perform large-scale Potts simulations on lattice sizes up to 4000x4000 in 2-D and 300x300x300 in 3-D, each with 100,000 or more lattice sweeps. These are the largest Potts-model simulations performed to date. The results are described in Section 3.2. New enhancements were developed for the conventional Metropolis algorithm used in the Potts-model simulations. These techniques enable both the sequential and parallel algorithms to run faster and use essentially "infinite" spin values so that grain coalescence events (where 2 different grains with the same spin value grow together) do not occur. These new algorithms are described in Section 3.3.

3.1. Conventional Algorithm in Parallel

The conventional sequential Potts-model on a periodic lattice of N sites is summarized in the previous section and may be described in the following way^{5,22}. First the lattice sites are initialized with random spin values from 1 to S. Then the following loop is executed N times:

1. Pick a lattice site randomly.
2. Pick a new spin value (from 1 to S) randomly.
3. Compute the energy change ΔE for the site to change to the new spin based on the number of unlike nearest neighbors.
4. Accept or reject the change based on the "zero temperature" Boltzmann

criterion describe in equation
(2-2).

$$P_i(\Delta E) = \begin{cases} 0 & \text{if } \Delta E > 0 \\ M_i & \text{if } \Delta E \leq 0 \end{cases}$$

N iterations of this loop constitute one "sweep" of the lattice which represents a unit of Monte Carlo "time". As sweep after sweep is performed spins preferentially flip to the same values as their neighbors to lower the system energy and individual "grains" (represented as geometric regions of the same spin) increase in size. A large workstation-class simulation to grow grains to a reasonable size might perform 100,000 sweeps on a 2-D lattice of size 200x200. On a large parallel machine we still want to perform 100,000 sweeps, but of much larger lattices. The most time-consuming step in the algorithm is step (3) which involves comparing the spin values of all neighboring lattice sites to the

old and new spin value to include their contribution to the energy change. In the Potts-model, "corner" neighbors are included in this operation, so the number of neighbors in a square lattice is 8 in 2-D and 26 in 3-D.

As stated above, the Potts algorithm does not readily parallelize in the obvious way of assigning each of P processors a subset of the loop iterations. This is because if step (1) is executed simultaneously on multiple processors, then two (or more) processors may pick adjacent (or the same) lattice sites. If this occurs, then when the two processors execute step (3), they will each attempt to flip the spin of their lattice site using incorrect information about neighboring spins (each other). This would violate the Monte Carlo rule of "detailed balance" which demands that two (or more) sites not be flipped simultaneously, if they interact with each other via their respective Hamiltonians.

An alternative route to parallelism which overcomes this difficulty is to partition the overall lattice so that each processor is assigned a contiguous sub-domain. In two dimensions this is a small rectangular section within the lattice, and in three dimensions it is a rectangular box. Each processor also stores a copy of the narrow strips (or planes in three dimensions) of lattice sites that immediately adjoin its sub-domain and which are actually owned by neighboring processors. This allows a processor to check neighboring spin values of sites on the edge of its sub-domain.

With these data structures, every processor can now simultaneously flip spins in its sub-domain without violating the rule of detailed balance, so long as one processor does not choose a lattice site on an edge of its sub-domain at the same time the processor adjoining that edge does likewise. We enforce this restriction in our parallel Potts algorithm by "checkerboarding" the lattice as shown in Figure 1. A lattice site is represented by a square (not the corners of the square) and assigned a "color" (or shading). For a 2-D lattice it is sufficient to checkerboard the lattice with 4 colors as in Figure 2a.

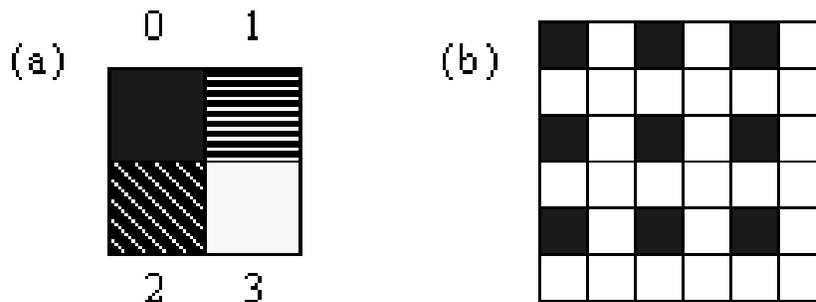


Figure 2. Checkerboard lattice used for parallel Potts grain growth algorithm.

This template is used to tile the entire lattice as in Figure 1b, where only the lattice sites assigned to color 0 are now shown as shaded. The key point is that the 8 neighbors of a color-0 lattice site do not include any other color-0 lattice sites. The parallel Potts grain growth algorithm for one sweep can now be written as follows:

Loop over colors (i)

(1) Loop over all lattice sites of color i within my sub-domain

(1a) Pick a new spin value (from 1 to S) randomly.

(1b) Compute the energy change for the site to change to the new spin.

(1c) Accept or reject the change based on the Boltzmann criterion.

(*) End lattice site loop

(2) Exchange sites along edge of my sub-domain with neighboring processors to acquire current neighbor spin values.

(*) End color loop

This algorithm works for both 2-D and 3-D lattices. In 3-D 8 colors are used to uniquely tile the lattice with a 2x2x2 template that insures none of the 26 neighbors of a lattice site have the same color as the center site. Also, the communication of sub-domain "edges" becomes "planes" in 3-D. Note that steps (1a)-(1c) are the same as steps (2)-(4) in the sequential algorithm, but they now are restricted to operating within a processors sub-domain.

This algorithm is highly parallel, with the only communication cost being the local exchanges of boundary spins (step 2) between neighboring processors. These exchanges also serve as synchronization points in the loop over colors to insure that all processors work on the same color at the same time. In practice, so long as the processor sub-domains were of reasonable size (50x50 or larger in 2-D), the communication costs were only a few percent of the total run time and thus the algorithm can simulate large lattices with parallel efficiencies of over 90% on large numbers of processors.

The parallel algorithm sequences through lattice sites differently than the sequential algorithm in two respects. First the parallel algorithm does not pick sites randomly, even within a processor's sub-domain. Second, the coloring scheme partitions the original lattice into colored sub-lattices which are updated one at a time. Some sequential implementation of Potts-models also alter the site selection sequence similar to the way our parallel algorithm does, without apparent ill effect. On vector supercomputers this is done to increase the vectorizability of the code. The most important criterion for this study was that our parallel algorithm not alter the grain growth statistics. Various tests were performed of this criterion; one example is shown in

Figure 3 where the average grain area and radius is plotted against time (Monte Carlo sweeps) for a simulation of a 200x200 lattice. The open squares and circles are results from a parallel simulation; the solid and dashed lines are from a sequential simulation. The excellent agreement between the two simulations indicates that the parallel algorithm reproduces the correct sequential Monte Carlo statistics.

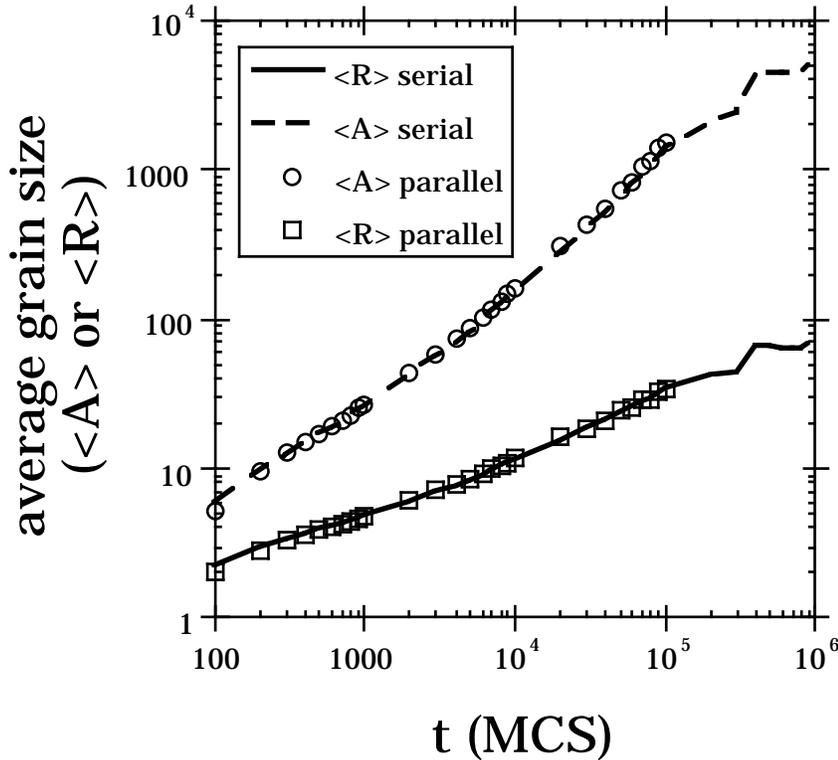


Figure 3. Comparison of parallel grain growth algorithm versus sequential algorithm for a 200x200 lattice out to 10^5 Monte Carlo time steps. The Open symbols are from the parallel algorithm. The lines are from the sequential simulation.

The parallel Potts algorithm uses random numbers to pick new spin values and, in non-zero temperature simulations, in deciding whether to accept uphill energy changes. Two different parallel random number generation (pRNG) schemes were implemented in our codes and verified that they produced independent (uncorrelated) random number streams that generated correct grain growth statistics. In practice one pRNG was used to initialize the lattice of spins and another to perform the Monte Carlo updates. The first pRNG is known as RANECU¹ and combines two linear congruential generators to create a stream with a period of 10^{18} . The second pRNG is based on a serial RNG due to Marsaglia² and is a lagged Fibonacci generator that can produce 900,000,000 independent streams each of length 10^{30} . On the parallel machine each processor produced random numbers from a different stream.

3.2. Large-Scale Parallel Potts-model Simulations

The parallel classical Potts algorithm runs at a speed of approximately 10 microseconds/site/sweep/processor for 2-D lattices on the i860 processors of Sandia's Intel Paragon. It is about 2-3x slower for 3-D lattices due to the increased number of neighbors that must be checked. This means that simulating lattices with millions of sites for 100,000 sweeps requires hours of CPU time even on thousands of processors.

Large-scale simulations were performed to study two phenomena that were previously computationally inaccessible. The first of these is finite-size effects on periodic lattices, to see if measurably different grain growth statistics could be observed when simulating larger and larger systems. The second was to test the theory of scaling self-similarity in the grain growth statistics. This theory states that grains should grow identically independent of their size (at least for sizes larger than a few pixels in area). In other words, a 2-D lattice with grains of average size 100 pixels should exhibit growth statistics the same as one with grains of size 400 pixels that was shrunk by a factor of 2 in each dimension. The goal in these large simulations was to test for finite size effects and self-similarity across a larger range of grain and lattice sizes than was previously possible.

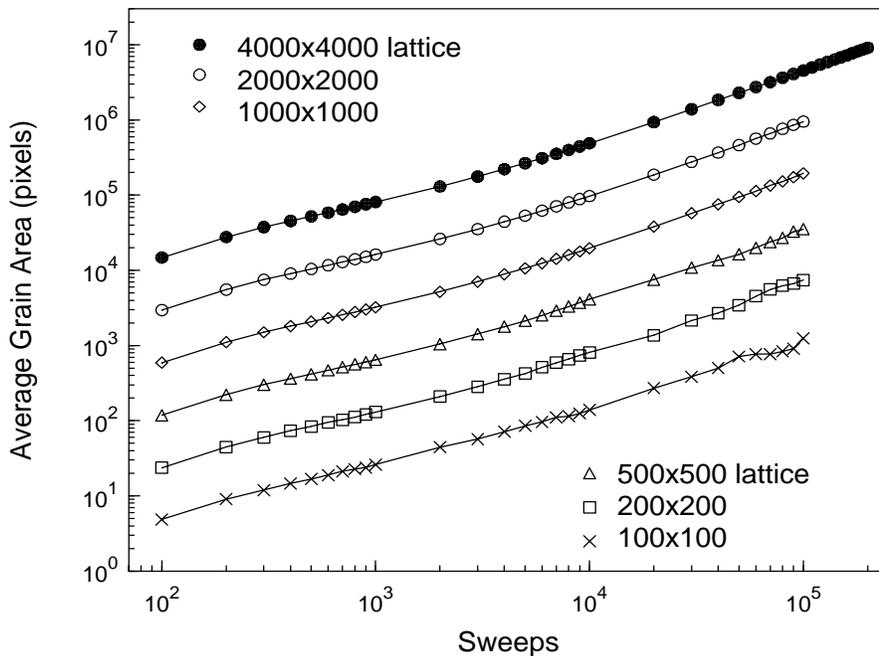


Figure 4. Two dimensional parallel algorithm grain grow study to examine self similarity and finite size effects for lattice sizes from 100x100 to 4000x4000. Grain areas are vertically offset and show statistically identical growth behavior independent of the lattice size.

The 2-D results for periodic lattice sizes ranging from 100x100 to 4000x4000 are shown in Figure 3 where the grain area is plotted vs. time, with the curves for each lattice size vertically offset from each other so as not to overlap. The area is defined as the

number of pixels per grain averaged over the entire lattice. To within statistical error these curves overlay each other "exactly"; no finite-size effects are observed. Similarly, the 3-D results for lattice sizes ranging from $30 \times 30 \times 30$ to $300 \times 300 \times 300$ are given in Figure 4. Again there is no evidence of finite-size effects, other than the statistical noise on the right side of the graphs for the smaller lattices, where the grains have grown to such a size that few of them are left.

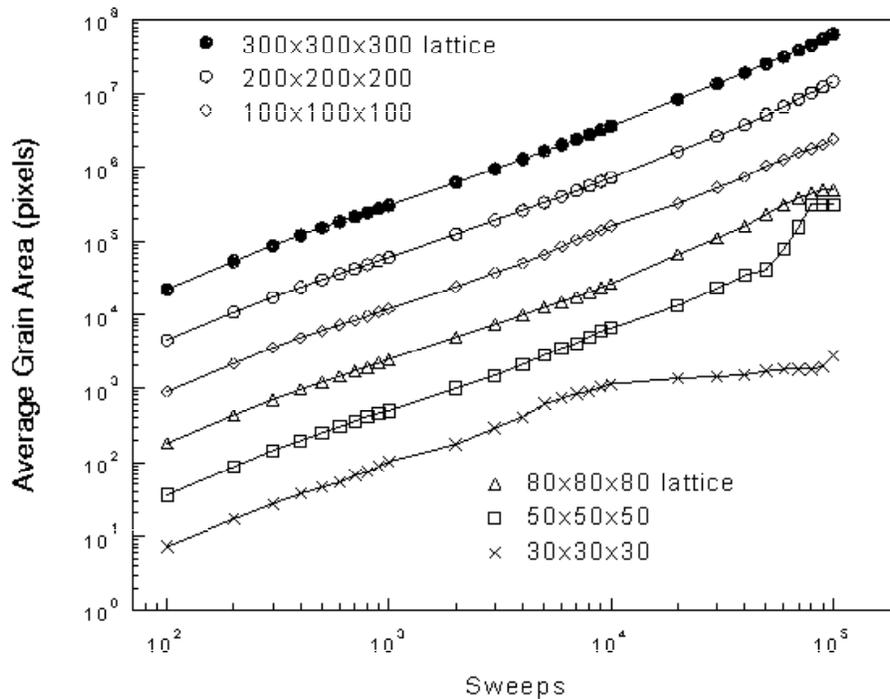


Figure 5. Three dimensional parallel Potts algorithm to study self similarity and finite size effects for varying lattice sizes. The three dimensional lattices vary from $30 \times 30 \times 30$ lattice sites to $300 \times 300 \times 300$. Grain areas are vertically offset for each lattice size. Statistical deviations occur for the smaller lattices because only a few grains remain after many time sweeps.

For completeness the second, third and four moments of the grain areas for the 2-D large scale Potts simulation were calculated and are shown in Figure 6. These results are discussed further in Plimpton³.

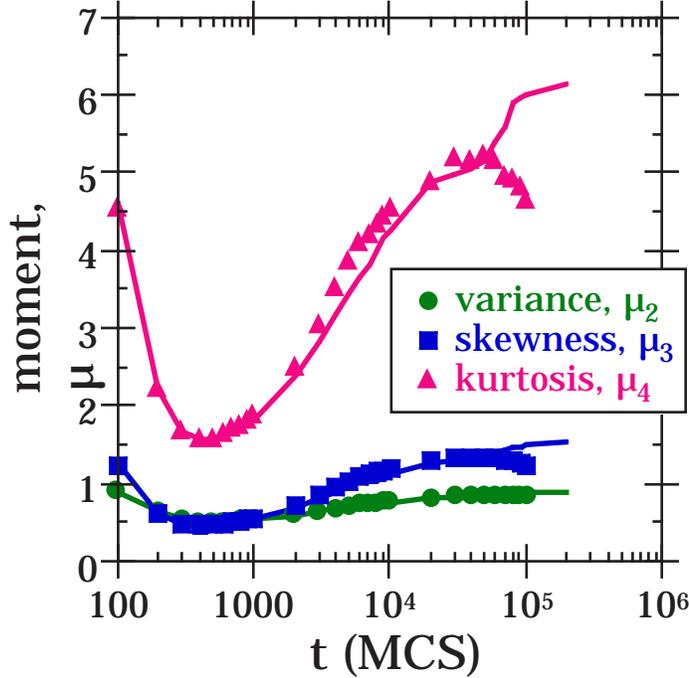


Figure 6. Second, third, and fourth order moments of the 2-D large lattice size grain growth simulations.

Performing these large simulations proved not only a computational challenge but also a data management challenge. A typical 100,000 sweep run requires dozens of lattice snapshots be saved for post-processing analysis. For a 16 million site lattice, each snapshot may be 16-64 Mbytes in size, depending on the precision (1 to 4 bytes) needed to store a single spin. Thus one run can generate up to a Gigabyte or more of data.

As part of the LDRD effort, several parallel and serial analysis tools were created to calculate appropriate statistics from these snapshot files. Some of the tools reorder the jumbled data as output in chunks by multiple processors into the parallel output files so that a serial processing code can interpret the files correctly. For the cases where "infinite" spin values were simulated (see the next section), a parallel on-the-fly analysis routine to identify individual grains within the global lattice and compute averaged statistics (radius, area, etc) across the entire lattice were developed. In this setting, no snapshot files need be saved since the grain statistics are computed in memory and the analysis itself can be done much more quickly using the full power of the parallel machine.

3.3. New Algorithms

3.3.1. Drawbacks of Conventional and "N-Fold Way" Algorithms

3.3.1.1. Conventional Algorithm

In the conventional $T=0$ Metropolis or Potts algorithm, as noted in the previous section, the simulation is begun by initializing all the sites with a Potts spin value randomly selected from the integers 1 through S , where S is the number of different allowed spins. Then, (1) a site is picked, either randomly for the serial algorithm or in a checkerboard sequence for the parallel algorithm; (2) a new spin value is randomly selected from 1 through S ; and (3) a change to the new spin value is made if and only if the change in energy is less than or equal to 0. This procedure is repeated until the desired number of sweeps through the lattice have been performed. A key point is that for large S , this procedure is very inefficient, as most of the attempted spin flips will have no chance of success. For example, consider the local configuration shown in Figure 7. An attempted flip of the central spin is only possible if one of the four neighboring spin values 5, 3, 4, or 1 is selected to be the "attempted" new spin. If $S=100$, for example, such a selection will occur on average only once every $100/4 = 25$ moves. If $S=200$, such a selection will occur on average only once every $200/4 = 50$ moves. Thus, the computer time necessary for a successful spin flip attempt scales linearly with S for the conventional algorithm and, for S large, much of the computer time is spent selecting and testing attempted spin flips which have no chance of success. In a 2-D or 3-D simulation, S must typically be set to at least 100-200, so that there is a negligible chance of two different grains with the same spin becoming adjacent to each other as the simulation progresses. When this happens nonphysical coalescence of the grains occurs.

```
5 3 4
5 7 1
5 7 1
```

Figure 7. Local lattice configuration for selection of a spin flip.

Another source of inefficiency in the conventional algorithm is trying to flip spins which are in grain interiors; i.e., which are in regions of the lattice which all have the same spin value. An example of such a local configuration is shown in Figure 8. It is clear that no spin flip is possible because a positive increase in energy is required. During the latter part of the simulation, when such configurations are common, much time can again be wasted in blindly attempting spin flips which can never occur.

7 7 7
7 7 7
7 7 7

Figure 8. Local lattice spin values for a grain interior.

In the following paragraphs some of these inefficiencies are addressed by an acceleration technique called the “N-fold way”.

3.3.1.2. N-Fold Way Algorithm

For each Monte Carlo step of the so-called "N-fold way" algorithm^{4 5}, a site is selected according to its probability of experiencing a spin flip and a change is then made at that site. There is a known relationship between the conventional and n-fold way Monte Carlo times, so that the algorithms give equivalent results. Because a spin flip actually occurs at each step of the n-fold way simulation, the inefficiencies of the conventional algorithm mentioned above are largely eliminated.

There are certain disadvantages, however, to the n-fold way algorithm. For large S , each Monte Carlo step involves either a $\log(N)$ or $N^{*0.5}$ search [see ref. 5], where N is the total number of lattice sites. In contrast, a conventional algorithm step involves only local operations with a small number of neighboring lattice sites. One consequence of this is that the conventional algorithm is faster than the n-fold way method at the beginning of the simulation [see ref. 5]. A typical procedure is thus to start the simulation using the conventional algorithm and then to switch later to the n-fold way. However, as mentioned previously, this initial use of the conventional algorithm brings with it a high computational cost for large S .

Another disadvantage of the N-fold way algorithm is that it is not easily or efficiently parallelizable. A method of parallelization has been discovered⁶ though not (to our knowledge) implemented; however, this method necessitates treating part of the lattice by the conventional algorithm, which will eventually become a bottleneck as the simulation progresses.

3.3.2. Fast Parallel Algorithms

3.3.2.1. Algorithmic Goal

The spin values 1 through S in the Potts-model correspond to the possible orientations of growing grains. Since real grains can orient in any direction, one would ideally like to perform Potts-model simulations with an infinite S . For practical purposes “infinite” means $S=N$ where N is the total number of lattice sites, so that the simulation

can initialize every lattice site with a unique spin value from 1 to N. Finite values of $S < N$ can lead to unphysical grain "coalescence", where two grains with the same spin value start to grow independently in different parts of the lattice, meet, and then merge (coalesce) unphysically. This coalescence can give rise to spurious fluctuations in the grain growth statistics.

One would also like an algorithm which is simply and efficiently parallelizable, to take advantage of the increased computing power now offered by massively parallel machines. Hence, our goal is the development of an algorithm which is fast and efficiently parallelizable, and which can explicitly handle the case of infinite Q ⁷.

3.3.2.2. Masking

As mentioned in the previous section, one inefficiency of the conventional algorithm is the amount of time spent attempting to flip spins in the interior of grains, attempts which have no chance of success. To remove this inefficiency, we assign a MASK variable to each lattice site, where MASK=0 if the site is in a grain interior and MASK=1 otherwise⁸ [ref 8]. Specifically, in the case of two dimensions, MASK is set to 0 if the spin value at a given site is the same as that of 5 or more of its 8 neighbors, and 1 otherwise. The MASK values are updated if the local environment of a site changes. Each time a site is visited, its MASK value is checked. If MASK=0, then no attempt is made to flip the spin and one proceeds to the next site. We have found that this implementation of "masking" leads to a speed-up of about three even for tests on small (100x100) lattices. The speed-up is greater as the simulation progresses and individual grains grow larger.

3.3.2.3. "Wild" moves

In the conventional Potts Monte Carlo algorithm, a spin which is different from any of its neighbors is allowed to flip to another spin value which is also different from any of its neighbors, since the energy change is 0. We call such flips "wild" moves⁵, and such spins "wild" spins; an example is given in Figure 9. Such wild moves are common during the early part of the simulation.

7 3 1		7 3 1
7 2 1	---->	7 5 1
6 4 1		6 4 1

Figure 9. Example of "wild" spin move, where central spin (2) changed to (5) with no energy change.

Since there is no reason why one value of a "wild" spin should be preferred on average over another, we conjecture that eliminating wild moves should lead on average to the same Monte Carlo kinetics as retaining such moves. To test this conjecture, we ran the conventional algorithm as before, allowing wild moves to be selected and attempted,

but always rejected such moves. As shown in Figure 10 for $S=100$ on a 1000×1000 lattice, this elimination of wild moves appears to have no effect. We combine the elimination of wild moves with the approximation of the next section.

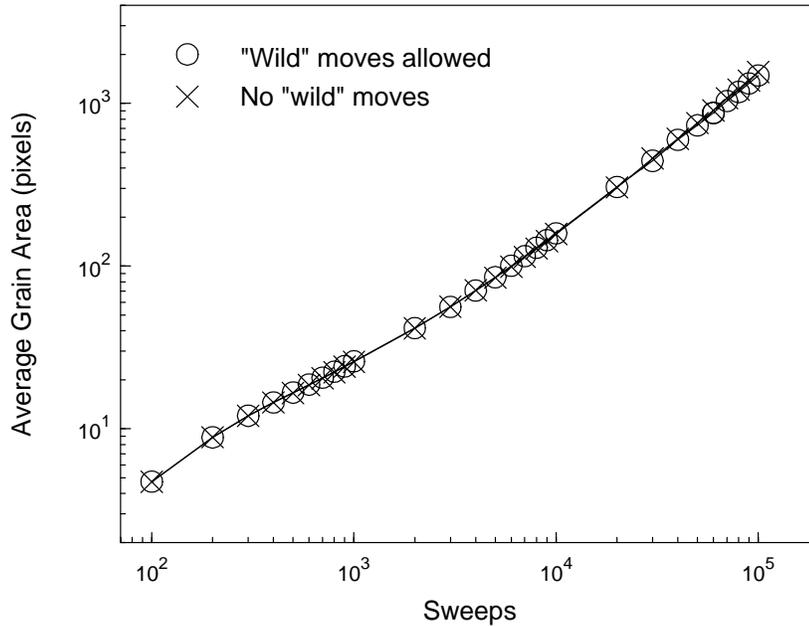


Figure 10. Parallel Potts simulation with and without allowing “wild” spin moves.

3.3.2.4. Fixed Neighbor Approximation

If we eliminate “wild” moves, then spins can only flip to one of their neighboring spin values. The probability of randomly selecting one of these neighboring values in a spin flip attempt is less than or equal to z/S , where “ z ” is the number of neighboring sites. For large S then, as noted previously, the probability of selecting an attempted new spin with non-zero chance of flipping is bounded by z/S , and the conventional algorithm simulation time hence scales linearly with S . As one way to eliminate these useless spin flip attempts, we utilize the “fixed neighbor approximation” (FNA). Given a so called “speed-up factor” F_S , this approximation is equivalent to attempting F_S spin flips every time a site is visited rather than attempting just one flip as in the conventional algorithm, with the neighboring spins held fixed during the F_S attempts. This leads to a speed-up of order F_S over the conventional algorithm; i.e., one sweep through the lattice using the FNA approximation corresponds to F_S sweeps with the conventional algorithm. We typically choose F_S to be of order S/z , so that the probability of one successful spin flip is of order 1. If desired, we can then extrapolate in F_S to the $F_S=1$ limit, since the FNA algorithm with $F_S=1$ is identical to the conventional algorithm.

The FNA approximation is implemented as follows. When a site is visited, we first calculate the number L of different neighboring spin values. For example, $L=5$ for the central spin in Figure 7 and $L=1$ for the central spin in Figure 8. We then compute the probability $P_S(m)$ of successfully selecting any of those L values m times in F_S

conventional spin flip attempts. Using standard probability arguments, $P_S(m)$ is given by:

$P_S(m) = \{ F_S! / [m! (F_S-m)!] \} (L/Q)^m \{ 1 - (L/Q) \}^{(F_S-m)}$. We then define the quantity

$R_S(m) = \sum_{m'=0, m} P_S(m')$, the probability of selecting one of the L neighboring spin values m or less times in F_S attempts.

As a check,

$$R_S(F_S) = \sum_{m'=0, F_S} P_S(m') = 1,$$

since the $P_S(m)$'s are just the terms in the binomial expansion of $\{ [1 - (L/Q)] + (L/Q) \}^{F_S} = 1$. Thus, the probability of making any type of change (including *no* change) is properly normalized to 1.

We then generate a random number R , with $(0 < R \leq 1)$. If $[R_S(m) < R \leq R_S(m+1)]$ (with $R_S(-1)$ set to 0), then, m times, we randomly pick one of the neighboring spin values and attempt a flip. We then proceed to the next site and repeat. Because neighboring spins are held fixed during this process rather than allowed to evolve, the FNA algorithm will underestimate grain growth.

We have so far implemented a simplified version of this algorithm. The simplification is that we do not attempt a spin flip if $[0 < R \leq R_S(0)]$ and make only one attempt if $[R_S(0) < R \leq 1]$. This limitation on the number of spin flip attempts will further slow grain growth. We compare with results obtained from the conventional algorithm in Figure 11, for a 1000x1000 system with $S=200$ and $F_S=25$. As predicted, the RNA algorithm underestimates the grain growth. However, we note particularly good agreement towards the end of the simulation. The full implementation of the FNA will be even more accurate.

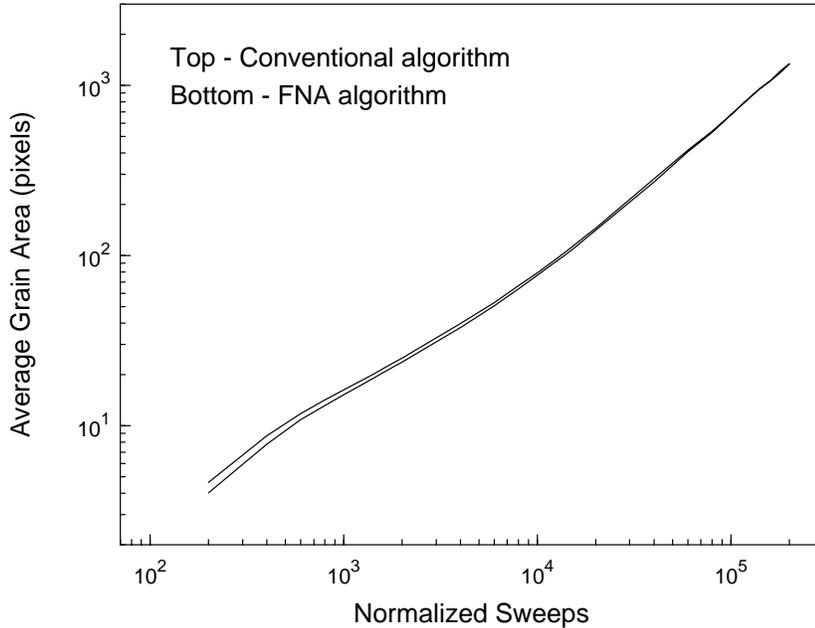


Figure 11. Comparison of Parallel Potts simulation using the Fixed Neighbor Approximations (FNA) to the conventional algorithm.

3.3.2.5. Infinite Q

As noted earlier, in initializing the spins, we can duplicate the effect of infinite S by simply setting each spin in the lattice equal to a different number⁹. In essence, it suffices that each initial spin be different from all others in the lattice. For example, initially each spin can be set to its lattice site number, 1 to N.

When the infinite S limit is taken on a large lattice, the conventional algorithm becomes even more computationally intensive because it picks up an additional power of N in computation time, as there are of order $S/z = N/z$ unsuccessful spin flip attempts for each successful attempt. Thus, in this limit, it is most convenient to express the speed-up F_S in the form $F_S = r(S/z)$. This "rescaling" of F_S has no fundamental significance, as Monte Carlo time is already only related to experimental time by some multiplicative factor.

In the infinite Q limit, the $P_{s(m)}$'s of the previous section now become $P_{r(m)} = \{ 1/m! \} (r*L/z)^m \exp[- (r*L/z)]$. The simplification of not more than one spin flip per step described above for finite Q is trivially extendible to the infinite Q case. Thus we have achieved our goal of a fast readily parallel algorithm which can explicitly handle infinite S, so as to avoid any possible grain coalescence events.

3.3.2.6. Timing

The FNA algorithm and masking can be used together, with the total speed-up when compared to the conventional algorithm combining multiplicatively. As mentioned previously, masking led to a speed-up of about 3 in test runs of small systems and, with $Q = 100$ and $F_S = 12.5$, the simplified FNA speed-up in computation time was about 5. Thus, the total speed-up was about 15. This is comparable to speed-ups on serial machines using the combined conventional-(n-fold way) algorithm for similar test systems. However, the FNA-masking algorithm is as simply and efficiently parallelizable as the conventional algorithm, in contrast to the n-fold way algorithm. Further, the FNA-masking algorithm can directly simulate infinite S , which is not possible with the combined conventional-(N-fold way) method. Thus, the FNA-masking algorithm has certain clear advantages over other existing methods with the caveat that FNA produces results that appear to underestimate grain growth by a small amount, particularly at early times in the simulations.

4. MASS TRANSPORT IN POLYCRYSTALLINE MATERIALS

Engineering materials exposed to aggressive environments often use coatings for their protective ability. Because the protective ability is strongly controlled by the diffusion of corrosive species through the coating microstructure, mass transport properties in polycrystalline materials were studied by simulating grain boundary and bulk diffusion of species through realistic two-dimensional microstructures. The microstructures were generated by Potts-model simulations. Traditionally, such properties have been studied by examining model polycrystalline structures, such as a regular array of straight grain boundaries. However, these models do not account for a number of features of realistic grain ensembles, including the grain size distribution and the topological aspects of grain boundaries. In this study, numerical diffusion simulation techniques were developed to study transient and steady-state mass transport through realistic two dimensional polycrystalline microstructures. The effects of microstructural parameters such as average grain size and grain boundary topology were examined, as were limitations of the model.

4.1. Introduction

Models that describe materials properties often over-simplify the effects of polycrystallinity because of the intractability of more precise methods. Models of grain boundary diffusion in polycrystalline single phase materials are a good example. Analytic solutions to this problem require two components -- a method for describing grain shapes and grain size distributions and a solution technique that can operate upon the microstructural description. However, both of these components have eluded modelers since a simple grain boundary diffusion solution was first proposed by J.C. Fisher in 1951.¹⁰ The lack of any significant advancement in modeling grain boundary diffusion in realistic microstructures and the availability of simulated microstructures for input into diffusion calculations prompted us to undertake a study of grain boundary diffusion using our simulated microstructures.

4.2. Background

Grain boundary diffusion coefficients are often six orders of magnitude larger than bulk diffusivities in polycrystalline materials. Therefore, although the volume of grain boundaries is a small fraction of the total material volume, grain boundary diffusion often dominates the effective diffusivity of solutes in real materials. Such accelerated diffusion affects various applications in different ways. For example, in protective films grain boundaries act as fast diffusion pathways through the material to an intolerant material beneath, shortening the effective life of the film. In a diffusion barrier, boundaries may increase the steady state flux of a contaminant through the film, lessening its overall effectiveness. In applications in which corrosion properties are composition dependent, the penetration of species into a specimen along grain boundaries may limit its

usable lifetime. Thus, an understanding of the progression of grain boundary diffusion is critical to the understanding of the long-term behavior of polycrystalline materials.

We will not go into the specifics of the numerous analytic solutions to grain boundary diffusion problems here. For an in-depth description of many of the solutions to such problems, refer to Kaur and Gust.¹¹ We will, however, provide an overview of what has been done.

The simplest approximated solution to a steady state grain boundary diffusion problem is one in which grain boundary diffusion and bulk diffusion are assumed to be independent of each other. Here, the overall diffusion coefficient is simply the weighted average, by cross section, of the two diffusion coefficients. This implies that there is no interaction of bulk grain flux and grain boundary flux, and that flux does not converge upon or diverge from grain boundaries. These assumptions may be reasonable in certain ideal cases, such as in microstructures with only longitudinal grain boundaries that only lie in the direction of the overall diffusion flux. However, having both transverse and longitudinal grain boundaries always results in an interaction between grain boundary and grain flux which leads to significant complications of the problem.

Better solutions to the problems of grain boundary diffusion have been proposed, but they are still generally limited to simple geometries. Both Fisher¹⁰ and Whipple¹² proposed solutions to the case of penetration of diffusant from a reservoir held at a constant concentration into a semi-infinite solid with an initial solute concentration of zero having a single grain boundary perpendicular to a surface. These solutions demonstrated that even a seemingly simple diffusion problem involving grain boundaries leads to a complicated, but still only approximate, solution. Mishin et al.¹³ investigated diffusion in the cases of single grain boundaries at different angles to the surface or grain boundaries between grains having different bulk diffusion coefficients. The case of diffusion into a polycrystalline thin film was investigated by Gilmer and Farrell.^{14,15} All these solutions still fall short of providing a complete solution to penetration and steady state diffusion in model microstructures, much less real microstructures, where the interaction of grain boundaries becomes much more complicated. Specifically, none of these models treats the complicated grain morphology encountered in most polycrystals.

The case of grain boundary diffusion in a 'real' microstructure was investigated by Levine and MacCallum,¹⁶ who proposed a different approach to the problem by solving the diffusion into a sample via a given volume fraction of high diffusivity boundary material coupled with diffusion into an average-shape grain. They found that the logarithm of the average concentration in a serial sample of a polycrystal varied as depth to the 6/5 power. Since all grain boundaries were combined into a single volume fraction of boundary material, grain boundary connectivity was again ignored.

Purely numerical methods have also been applied to the study of grain boundary problems. Levine and Losee¹⁷ used a Monte Carlo method to simulate the diffusion of impurities in grain boundaries. Their method was unique in that it explicitly allowed for equilibrium grain boundary segregation of impurity diffusants. Holloway et al.¹⁸ used a

finite difference method to solve the case of grain boundary diffusion along straight boundaries in thin films and compared their results to an analytic solution for the same problem. Gui et al.¹⁹ verified the transmission-line-matrix method for the solution of model grain boundary diffusion problems for which analytic solutions were available as well as applied the technique to other simple geometries.

In summary, a number of analytic and numerical models for grain boundary diffusion problems have been proposed.¹⁰⁻¹⁹ Most of these models use simplifying assumptions that restrict either the geometry and topology of the system or the time regime of applicability. In addition, most models ignore the effect of three-grain junctions which can act as bottlenecks to diffusive flux. To overcome these restrictions, we developed a flexible numerical simulation technique that can be applied to arbitrary bitmapped microstructures to simulate diffusion over all time regimes.

4.3. Method

The input microstructures for our numerical diffusion simulations are bitmaps wherein each pixel is assigned a spin index that identifies it as a member of a particular grain. The use of the bitmap lattice gives a grid of lines that separate lattice sites. These gridlines and their points of intersection represent solute diffusion paths and points of known solute concentration during the diffusion simulation, respectively. Each line segment is assigned a diffusivity corresponding either to that of a grain boundary or that of the bulk, as shown in

Figure 12. In these simulations, both the bitmap and the diffusion grid are square lattices, and grain boundaries are assumed to be one unit wide. The microstructures may have either periodic or closed boundary conditions at the edges of the lattice. Finite sources or sinks are created by setting the initial concentration at points in the diffusion lattice as desired, or we hold the concentration of these points constant to create infinite sources or sinks. The flux out of or into infinite sources or sinks can be measured so that the steady state flux in various diffusion problems can be determined.

Diffusion along each path in the grid takes place according to a discretized version of the usual diffusion equation

$$\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) \quad (4-1)$$

where C is the concentration, t is time, x and y are spatial coordinates, and D is the appropriate diffusion coefficient (bulk or boundary) for the path. Both finite-element and finite-difference methods were utilized to solve equation (4-1).

The finite difference (FD) method uses the Crank-Nicholson algorithm in 2-D as implemented by the Alternating Direction Implicit (ADI) method with an adaptive timestep. This method gives, in general, good stability and second-order accuracy, so that

timesteps that are long as compared to those in a purely explicit FD method can be utilized. Still, this method has limitations when applied to grain boundary diffusion. Its accuracy is diminished at grain boundary tri-junctions when using the very long time steps that we desire as the concentration profile approaches its steady-state value. Also, it is unable to model grain boundaries that are realistically thin as compared to average grain size. This simulation method results in the creation of grain boundaries with an effective width-to-grain-size ratio of at least $1/L$, where L is the number of pixels along a side of our Potts-model simulated microstructure. Ratio values of about 2.5×10^{-3} for a typical simulation fall well short of values of less than 1×10^{-4} for real materials.

The finite element (FE) method can be used to obtain solutions of both transient and steady-state mass diffusion problems, and promises several advantages for accurately modeling realistic grain structures. Because it, unlike our ADI implementation of the finite-difference method, is implicit in both dimensions simultaneously, it does not suffer from long timestep limitations. Also, the thickness of grain boundary elements can be a parameter independent of the grid, so a single grid can be used to model several different widths of grain boundaries. In this work, the same grids were used for the FE method as were used for the FD method, with each gridline between nodes defined as a 1-D element. At this stage, the FE method is used to solve diffusion problems, to take advantage of its greater flexibility in the future. Accessing the additional flexibility that the finite-element method can provide, requires developing a meshing technique to describe a microstructure free of the constraints of a uniform lattice and utilize a more advanced finite-element code than our home-grown version. Because the FE work is still in its early stages, most of the simulation results presented are derived from FD simulations.

The finite-element method converges to a steady-state solution much more rapidly than does our implementation of the finite-difference method on properly framed “finite-difference” type problems. This has less to do with the fact that one is FD and the other is FE than the fact that one (FE) is fully implicit while the other (FD) is not. This suggests that to get optimal performance from the FD method for grain boundary diffusion problems, it should be fully implicit, a point that we did not realize at the outset of this work.

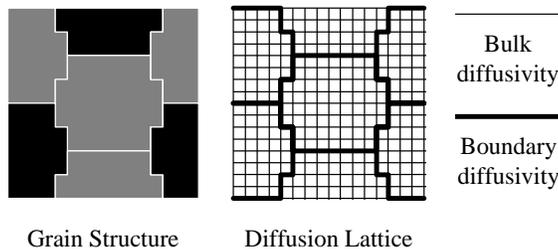


Figure 12. The method by which a bitmapped microstructure is mapped to a diffusion grid. Note that diffusivities along grain boundaries correspond to the grain boundary coefficient, diffusivities across grain boundaries correspond to the bulk coefficient.

4.4. Validation

Our FD technique was validated by comparing to a number of analytic results. Excellent agreement was found across all time regimes for diffusion from an infinite source into a finite single crystal, from a semi-infinite source into a semi-infinite crystal, from arrays of thick film solute sources, and others.²⁰

In numerical simulations, the concentration gradient is represented by the concentration at discrete points. To determine if this discretization affects diffusion results, comparisons were made to the analytic solutions for diffusion in a semi-infinite single crystal with a finite thin film solute source.²⁰ In the simulation, the solute source was a single unit wide line having a different initial solute concentration than the rest of the lattice. As shown in Figure 13(a), the simulation shows excellent agreement with the exact solution.

Another concern is stability and accuracy of the simulation when diffusivities have discontinuities. To examine this case, comparisons were made of simulation results to the analytic solution for a semi-infinite slab solute source with diffusivity D_1 bonded to another semi-infinite slab with diffusivity D_2 .²⁰ As shown in Figure 13(b), the simulation is in excellent agreement with the exact solution, even near the diffusivity discontinuity located at $x = 500.5$.

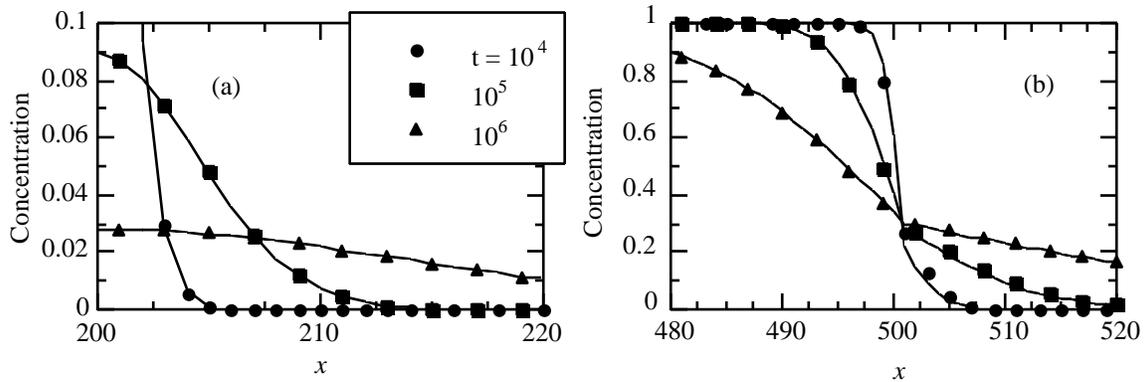


Figure 13. Comparison of finite difference calculation results (lines) to analytic results (points) for (a) diffusion from a thin-film source at $x = 200$ and (b) diffusion in a heterogeneous diffusion couple with an interface at $x = 500.5$.

In both these validation steps, the direction of diffusion was along the lines of the direction of the lattice, and thus these tests are treated as 1-D diffusion problems by the simulation procedure. However, our goal was to verify the accuracy of our simulation procedure in 2-D, but 2-D transient diffusion problems are difficult to solve analytically. To circumvent this limitation, the direction of diffusion was rotated with respect to the lattice by a 45° so that the simulation procedure would not solve them as 1-D problems. Figure 14 shows the results of such a test. Diffusion from an infinite source into a finite

open slab was simulated, and again excellent agreement was found with the exact solution.

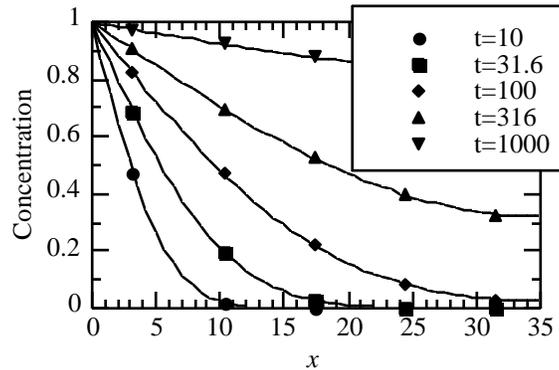


Figure 14. Comparison of finite difference calculation results (lines) to analytic results (points) for diffusion in a finite open slab in which direction of diffusion does not correspond with lattice directions.

Grain boundary diffusion along multiple grain boundaries running through a finite slab was simulated, as shown in **Figure 15**. Transient concentration results were compared to those published by Gilmer and Farrell.¹⁵ When using precisely the parameters they reported, quantitative disagreement was found between the two models, despite qualitative similarity. However, good agreement could be obtained if the width scale was doubled (i.e. grain boundary width and separation) in our films. This agreement is shown in Figure 16. Thus, it appears that there may be a factor of two discrepancy in the width scale definitions of the models. Furthermore, Gilmer and Farrell use a parameter that is the product of the grain boundary width and grain boundary diffusion coefficient in the description of their system. This single parameter is insufficient to describe the combined effect of grain boundary thickness and grain boundary diffusion coefficient when the transient concentration profile is desired, therefore discrepancies in the results that diminish with time are attributed to this ambiguity in their system description.

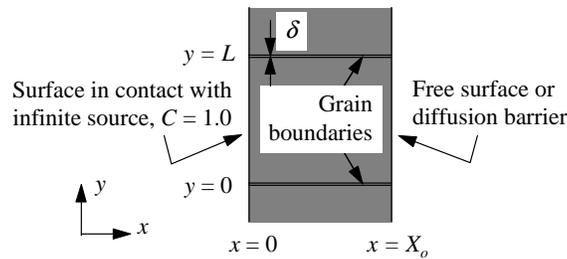


Figure 15. Geometry of diffusion problem solved by Gilmer and Farrell.¹⁵

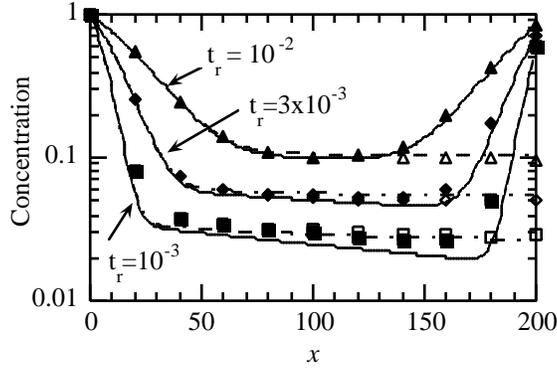


Figure 16. Comparison of finite difference calculation results to semi-analytic results reported by Gilmer and Farrell¹⁵ for the solution of diffusion into a finite slab of thickness $X_0 = 200$ units thick with equally spaced grain boundaries normal to the surface and a ratio of boundary to bulk diffusivity $D_b/D_g = 1000$. Gilmer and Farrell's results use a boundary separation $L = 200$ units and a grain boundary width $\delta = 1$. The finite difference results use $L = 400$ and $\delta = 2$. Dashed lines and open symbols correspond to the case of diffusion into a film with a diffusion barrier on the back side. Solid lines and closed symbols correspond to the case of diffusion into a film with a high diffusivity surface on the back side.

4.5. Steady State Diffusion

The validated finite difference technique was utilized to study steady state diffusive flux through polycrystalline microstructures. Realistic two-dimensional grain structures were generated by Monte Carlo Potts-models simulations for grain growth on the square lattice with first and second neighbor interactions.^{21,22} Our goal in this case is to compare the effective diffusion coefficient for a realistic microstructure to that for an idealized system.

Consider a system with an infinite solute source at $x = 0$ in which all grain boundaries are parallel to the x -axis. As usual, the steady state flux of solute $J = D_{eff} \Delta c / \Delta x$, where D_{eff} is the effective diffusivity for the medium. To a first approximation, the boundary and bulk contributions to the flux may be decoupled, so D_{eff} is given by a rule of mixtures relation

$$D_{eff} = \frac{D_b \delta}{d} + \frac{D_g}{d} \left(1 - \frac{\delta}{d}\right) \quad (4-2)$$

where D_g is the diffusion coefficient in the grain bulk, D_b is the grain boundary coefficient, δ is the grain boundary width, and d is the mean grain size (diameter). Geometry shows that δ/d is the approximate area fraction of grain boundaries. If $D_b \delta \gg D_g$, as in this study, then $D_{eff} \approx D_b \delta/d$, which is plotted as the upper line in Figure 17.

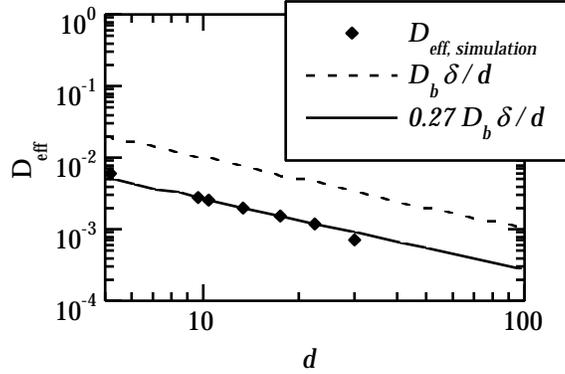


Figure 17. Effective diffusion coefficient D_{eff} as a function of average grain size d for simulated microstructures with $D_b = 10^{-1}$ and $D_g = 10^{-6}$. The rule of mixing coefficient $D_{eff} = D_b \delta / d$ and the corrected rule of mixing coefficient $D_{eff} = 0.27 D_b \delta / d$ are shown for comparison.

Using the finite difference diffusion simulation, effective diffusivities from the steady state flux through microstructures with varying average grain sizes were computed. These data are plotted as the symbols in Figure 17. The measured effective diffusivity varies inversely with grain size as predicted by the rule-of-mixing. However, D_{eff} is smaller than the rule-of-mixing predicts at all grain sizes.

Three factors which combine to decrease D_{eff} in the simulation microstructures were determined. These are boundary junction bottlenecks, indirect grain boundary paths, and discrete lattice effects.

First, three-grain junctions or so called “triple” points are grain boundary diffusion flux bottlenecks. Consider, for example, a three-grain junction in which the two inlet boundaries are carrying their full flux capacities. The outlet boundary can carry only half of the total possible inlet flux. Likewise, in a junction with one inlet boundary at full carrying capacity, both outlet boundaries receive only half of the inlet flux. Monte Carlo simulations which generate a random collection of both types of junctions and compute the average flux carrying capacity of the boundaries show that for large numbers of junctions, the flux carrying capacity seems to asymptote at 0.43 times the ideal capacity.

In addition, two geometric factors decrease boundary flux. In an array of polygonal grains, the boundaries form an indirect path across the sample. For example, in a hexagonal array of grains, the shortest point-to-point path is $2/\sqrt{3} = 1.15$ times the distance between endpoints. For endpoints at the least favorable misorientation, the shortest path is $4/3 = 1.33$ times the endpoint separation. We expect an average path to be about 1.24 times longer than the sample width, which decreases flux to about 0.8 times the ideal capacity.

The final factor is purely a simulation artifact. In the square lattice used here, each boundary is represented as a zigzag of boundary segments parallel and perpendicular to the x -axis of the simulation lattice. Due to this discretization, in a system of randomly oriented boundaries, the boundary path is a factor of $4 / \pi = 1.3$ times longer than the nominal point-to-point distance, and the flux is hence about $\pi/8 = 0.78$ times the ideal capacity.

Combining these three flux decreasing factors, we can predict a corrected rule-of-mixing diffusivity for realistic polycrystals. Specifically, $D_{eff} = 0.27D_b\delta / d$, which is plotted as the lower line in Figure 17. The agreement between the corrected rule-of-mixing diffusivity and the observed effective diffusivity is excellent. In fact, the rule-of-mixing diffusivity is very slightly lower than the observed diffusivity, because it ignores bulk diffusion around the flux bottlenecks, which does occur in the simulations, particularly at small grain sizes. In addition, deviations from this scaling occur when the grain size approaches the same order of magnitude as our finite lattice.

4.6. Transient Diffusion Profiles

The ability to closely observe transient diffusion is an advantage of our numerical methods. For example, Figure 18 shows solute concentration across a large-grain two-dimensional polycrystal during transient diffusion. The grain boundaries are clearly delineated by solute concentration peaks, and concentration is relatively depleted near the grain centers.

Levine and MacCallum have proposed a model for solute penetration into a polycrystal.¹⁶ They coupled transverse diffusion into the sample via a given volume fraction of high diffusivity boundary material with lateral bulk diffusion into an average-shape grain. Numerical analysis of their results predicts that the logarithm of the average concentration at depth x should scale with $x^{6/5}$ in certain time and penetration regimes. This result has become the basis for experimental determination of grain boundary diffusion coefficients. Solute concentration is measured in serial sections of a polycrystalline diffusion specimen and compared to the penetration curves of Levine and MacCallum.¹⁶

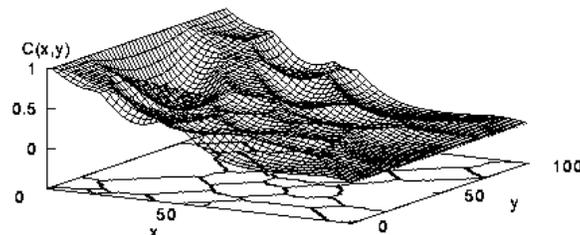


Figure 18. Solute concentration profile in a two-dimensional microstructure. An infinite source ($C = 1.0$) is at $x = 0$, an infinite sink ($C = 0.0$) is at $x = 100$, and the boundaries $y = 0$ and $y = 100$ are periodic. The grain structure is shown on the base plane of the plot.

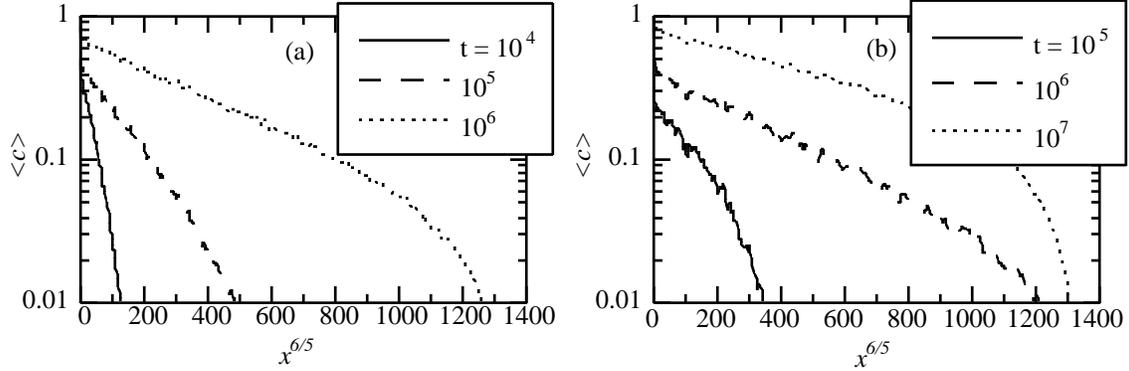


Figure 19. Solute penetration into two-dimensional polycrystalline microstructures plotted in the functional form suggested by Levine and MacCallum.¹⁶ $D_g = 10^{-6}$, $D_b = 10^{-1}$, (a) $d = 5$, (b) $d = 10$.

Solute penetration curves for finite difference diffusion through two-dimensional grain structures are shown in Figure 19. Curves show a fairly linear dependence of $\log \langle c \rangle$ on $x^{6/5}$ for small x , as predicted by Levine and MacCallum. The curvature at long times and large penetration depths is of the correct sign as well. The slopes of the linear regions of the curves are uniformly lower than the predicted slopes; that is, the solute concentration at a given penetration is smaller than predicted. There are two reasons for this solute deficiency. First, Levine and MacCallum combined all grain boundaries into a single volume fraction of boundary material, so bottlenecks at three-grain junctions and the path length effects discussed above are ignored. As in the steady state example, these effects will decrease the solute carrying capacity of boundaries in real polycrystals. Second, the stereological parameters used in the analysis of Levine and MacCallum are three-dimensional. The polycrystal in our study is strictly two-dimensional. Again, this decreases the solute carrying capacity of our boundaries. Nonetheless, solute penetration in transient state diffusion agrees well with the functional form given by analytical results.

4.7. Conclusions

Two numerical methods for solving grain boundary problems in realistic microstructures have been implemented. The first is a flexible finite difference technique for solute diffusion in a bitmapped microstructure based on the Crank-Nicholson scheme that uses the Alternating Direction Implicit (ADI) method to provide stability for long timesteps. This implementation uses a variable timestep to maintain a desired precision for both short-time and long-time regimes of diffusion. The second technique is a finite element technique that promises greater flexibility in the modeling of grain boundary diffusion, but is not fully implemented at this time.

The description of transient grain boundary diffusion cannot be parameterized using the product of the grain boundary diffusion coefficient and the grain boundary width. To get realistic diffusion behavior, both realistic grain boundary widths and grain

boundary coefficients and not just their product must be modeled. Real microstructures contain grain boundaries with thicknesses that are at most 10^{-3} of the average grain diameter. In our simulations, grain boundary thicknesses are at least 10^{-3} of the average grain diameter. This is a limitation for the FD method, since the narrowness of the grain boundaries that we can simulate is limited by the resolution of lattice on which we simulate diffusion, and modeling narrower grain boundaries would require using larger simulation lattices. The ability to model narrower grain boundaries without dramatically increasing the size of the problem is an expected benefit of solving this problem using the FE method.

Excellent quantitative agreement with analytic results for one-dimensional diffusion problems, including the thin film solute source and semi-infinite slabs of differing diffusivities was seen.

In simulations of steady-state grain boundary diffusion in two-dimensional polycrystals, it was found that the effective diffusivity varies with the inverse of average grain size, as predicted by a rule-of-mixing diffusivity. Flux bottlenecks at three-grain junctions and geometric factors which increase boundary path length decrease the magnitude of the observed diffusivity to about 0.27 times the rule-of-mixtures prediction.

Transient state boundary diffusion agrees with the solute penetration analysis of Levine and MacCallum.¹⁶ However, the magnitude of the concentration as a function of penetration is again decreased due to junction bottlenecks and geometric factors.

5. GRADIENT DEPENDENT POTTS-MODELING SIMULATIONS

The goal in this section is to investigate temperature dependent and temperature gradient dependent grain growth phenomena via Potts-model simulation methods. The simulations required modifying the grain growth dynamics to include mobility effects and to include additional terms in the Hamiltonian to account for biased migration along the temperature gradients.

5.1. Background

In nuclear reactor fuel (especially uranium oxide) a number of grain morphologies are observed across the radius of a fuel pellet. These morphologies and the physical mechanisms responsible for them are discussed in Olander.²³ Typically, near the center of the fuel pellet, long columnar grains are observed. In the middle portion of the pellet a region of equiax grains are observed, and at the outer edge of the pellet the grain structure resembles the as-fabricated grain structure and is called the unstructured zone.

These types of grain morphologies are primarily caused by two effects: 1) by grain boundary mobility dependence on temperature which varies substantially across an oxide fuel pellet, and 2) by vaporization and condensation mechanisms across grain boundaries. Oxide nuclear reactor fuels have very low thermal conductivities and as a consequence very large temperature differences exist across the fuel pellet. The temperature profile may vary by over 1000 K over just a few mm. The hotter fuel pellet regions produce larger grains due primarily to increased mobility of the molecules. In addition columnar grain structures are produced in the hottest regions when there is a temperature gradient because molecules vaporize from the hot side of a small void and condense on the cold side. This results in motion of voids or pores up the temperature gradient. As the pores move up the temperature gradient, columnar grains are left behind.

The goal of this section; therefore, is to simulate the growth of columnar grains due to biased migration effects, and to vary the mobility of the grain boundaries to produce substantial grain growth at hot portions of the simulation and little or no grain growth in the coldest portions.

5.2. Mobility and Biased Migration

The Potts-model simulation method is described in a number of reports^{5,22} and in section 2. In this simulation we included mobility effects in the basic computational model. For clarity we have reproduced and summarized most of the zero temperature Potts-model description in this section.

As usual the microstructure of a material is mapped onto a 2-dimensional grid or lattice. At each lattice site an index, S_i , is assigned which corresponds to the orientation of the grain at that site. This index is called the spin. Adjacent sites with different spin orientations form interface sites while sites with only like nearest neighbors are interior sites. The total system energy of the simulation over all sites is specified by assigning a positive energy to interface sites and zero energy to interior sites. It is computed via the Hamiltonian

$$H = \frac{J}{2} \sum_{i=1}^N \sum_{j=1}^z (1 - \delta(S_i, S_j)) \quad (5-1)$$

where the outer sum (i) is overall all sites in the system, and the inner sum (j) is only over the z nearest neighbors. The Kronecker delta function is defined as $\delta(S_i, S_j) = 1$ if $S_i = S_j$, and 0 otherwise. Typically in these simulations $S=100$, thus allowing 100 different types of grains. J is a constant used to scale the grain boundary energy. In essence the system energy is J times the number of boundary segments in the system.

Grain growth kinetics are determined by Monte Carlo selection of the site and then deciding whether or not to change the spin at this site based on the total system energy change. This is accomplished here by first randomly selecting a lattice site i and a new site spin S_j different from the original spin S_i . The spin index of site i is then changed to the new index (S_j) with probability given by

$$P_i(\Delta E) = \begin{cases} 0 & \text{if } \Delta E > 0 \\ M_i & \text{if } \Delta E \leq 0 \end{cases} \quad (5-2)$$

where M_i is a scaling function dependent on the mobility of site i , and ΔE is the energy change for assigning site i the new spin index S_j . ΔE is calculated by calculating the difference in like-spin neighbors around site i when it has the old spin and the new spin. When the number of nearest neighbors has more like spins with the new spin S_j , then ΔE is negative, resulting in a lower value for the Potts-model Hamiltonian. ΔE is just the change in the Hamiltonian due to change in spin of a single site i .

Note that the mobility function, $0 < M_i \leq 1$, scales the acceptance rate of the energetically favorable changes. Thus when the mobility function is high the likelihood of changing the spin to one of the neighbors is high and thus the grains grow more quickly when M_i is close to 1. Similarly, the growth rate is slow when M_i is small. Thus, the magnitude of the simulation mobility function M_i is not equal to a physical mobility; rather it represents a ratio of the physical mobility at fastest and slowest locations within the simulation.

5.2.1. Grain Boundary Velocities

The velocity of the grain boundary (or lattice site) is defined as the product of the mobility and the force on the system.

$$v_{gb} = M_{gb} F_{gb} \quad (5-3)$$

Thus both the mobility of the grain boundary system and the driving forces must be accurately modeled in the Potts-model simulations to accurately predict the grain boundary morphology of the system.

For normal grain growth the force on the grain boundary causing grain growth is proportional to the curvature of the grain

$$F_{gb} = 2 \sigma / r \quad (5-4)$$

where $2/r$ is the curvature of the grain and σ is the surface energy associated with the interface. Note that in the Potts-model simulation, the curvature of the grain is ΔE , and it is calculated by summing the difference in like neighbors for the new and old spin values.

The mobility of the grain boundary is calculated by the Nernst-Einstein equation:

$$M_{gb} = \frac{D_{gb}}{kT} \quad (5-5)$$

The grain boundary diffusion coefficient is just

$$D_{gb} = D_o e^{-Q/kT} \quad (5-6)$$

where D_o is the prefactor for the diffusion coefficient, Q is the activation energy of moving the grain boundary, k is the Boltzmann constant, and T is the temperature.

Because the simulation mobility function M_i is a ratio of physical mobilities it can now be defined as:

$$M_i = \frac{T_o}{T} e^{-Q \left(\frac{1}{kT} - \frac{1}{kT_o} \right)} \quad (5-7)$$

Note that when $T=T_o$, M_i equals one, and when T is small, M_i approaches zero. Thus the requirement that $0 < M_i \leq 1$ is satisfied.

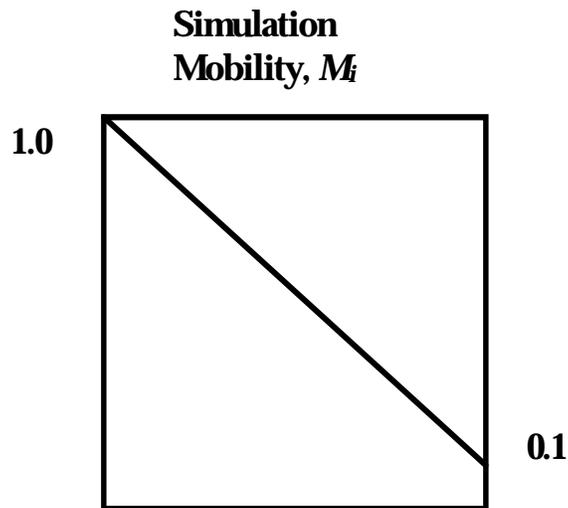


Figure 20: Simulation mobility profile used to bias grain growth to large grains in the higher mobility regions.

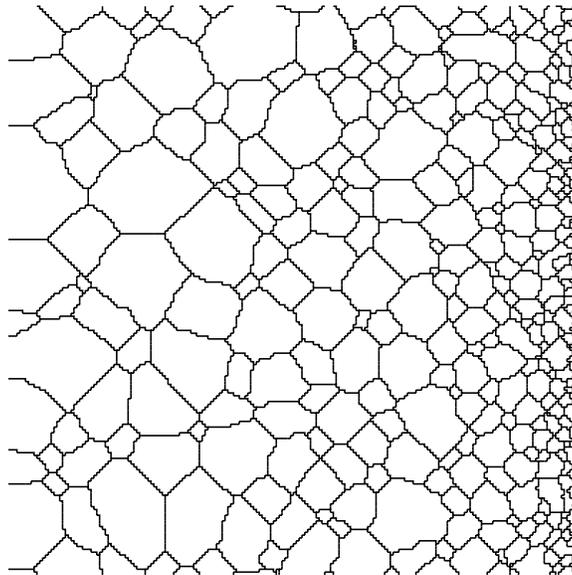


Figure 21: Grain growth with spatially dependent mobility profile. Observe the large equiaxed grains on the high (hot) mobility side and the small equiaxed grains on the low (cold) mobility side.

Tests of the above mobility model were performed by Holm²⁴ assuming a linear mobility curve with the value of 1 at the “hottest” portion of the simulation and a

simulation mobility of 0.1 at the coldest portion. Figure 20 illustrates this assumed mobility function.

The results of the grain growth simulation with this mobility profile are illustrated in Figure 21. This figure shows that changing the mobility function does indeed change the rate of growth of the grains. The left side (hottest) or highest mobility portion shows the largest grains, while the right hand side shows the least growth.

5.2.2. Migration in a Temperature Gradient via Vapor-transport

As described above, other driving forces or growth mechanisms can play important roles in the development of grain morphology. In materials at very high temperatures and under the influence of a temperature gradient, vapor-transport mechanisms become another governing force capable of moving pores and grain boundaries. Thus, the equation for the velocity of the grain boundary must be generalized to include vapor-transport terms as well as curvature.

$$v_{gb} = M_{gb} \left\{ F_{curvature} + F_{vapor} \right\} \quad (5-8)$$

The driving force for vaporization/condensation across grain boundary pores is given by Olander(ref)

$$F_{vapor} = N_p \frac{\Delta H_{vap}}{T} \left(\frac{dT}{dx} \right)_p \quad (5-9)$$

where N_p is the number of molecules displaced by the volume of the pores on the grain boundary, ΔH_{vap} is the heat of vaporization of the molecules, T is the local temperature, and dT/dx is the temperature gradient at the grain boundary.

Note, that in normal grain growth the driving force is curvature, and as a consequence the Potts-model simulation used a Hamiltonian or energy change ΔE to cause proper grain growth. For the case of biased migration in a temperature gradient (such as vapor-transport across grain boundaries), it again makes sense to modify the Hamiltonian to drive the columnar grain growth. Rigorously, this should be performed according to the driving term defined above in equation (2-1).

This was approximated in this study by simply introducing a temperature dependent modification to the Hamiltonian and providing a scale factor to adjust the relative strength of curvature driven grain growth to biased grain growth .

The gradient term was implemented by modifying local changes in the energy ΔE of a site by: adding 1 for like neighbors on the -x (hotter) side, or $\sqrt{2}/2$ for like neighbors in the diagonal. Thus,

$$\Delta E = \Delta E_{curvature} + \Delta E_{dT/dx} \quad (5-10)$$

The equation and figure below illustrate this modification.

$$\Delta E_{curvature} = C \sum_j^z (1 - \delta(S_i, S_j)) * G(j) \quad (5-11)$$

$G(j)$ is a multiplier to the like neighbor summation to bias the migration of like-spin-neighbors to align with the temperature gradient. C is a normalization constant, which in this simulation was set to 1.0.

0.707	0	0	=G(j)
1.0		0	
0.707	0	0	

Figure 22: Gradient term modifying the Hamiltonian

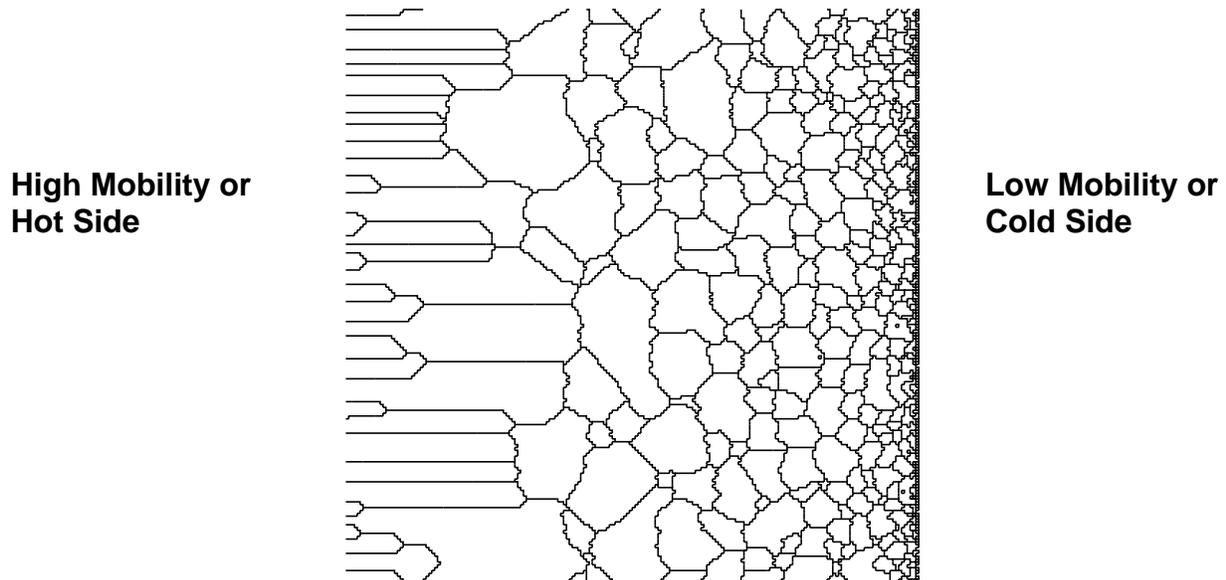


Figure 23: Gradient dependent grain growth showing columnar grains at high mobility side, equiaxed grains in the middle, and little grain growth on the cold or low mobility side.

With these modifications to the Hamiltonian the results of the biased grain growth including a 10/1 mobility profile as before are shown in Figure 23.

As desired, the grain growth turned to columnar growth at the hottest side, equiax in the middle, and exhibited little growth at the cold site. This simulation is found to replicated quite well the expected growth behavior observed in nuclear fuels.

However, as mentioned above, the true mechanism is vapor-transport across pores trapped on the grain boundary, which results in pores moving up the temperature gradient. We have experimented with moving pores to simulate more realistically true physics of the columnar growth phenomenon. It was found that the pores would move readily to the grain boundaries, and preferentially to the corners of the grains. However, moving across grains required extremely large gradient terms. It was felt that the pore movement in question could only be modeled using a Potts-model with a finite temperature, rather than the zero-temperature version used here. It is also believed that a more rigorous use of the real driving functions should be included to assure proper balancing of curvature, temperature gradient and other terms in the modeling.

5.3. Biased migration using the N-Fold Algorithm

The gradient or biased migration term can also be used with the zero-temperature N-fold algorithm. The N-fold algorithm works by setting up lists of sites that have neighboring spins that allow the target site to be flipped to match its neighbor. In other words, sites are excluded that cannot change (i.e. those where the target site already matches all of its neighbors). The energy of the site is determined, as in the conventional Potts algorithm, by the number of same-spin neighbors, as is the energy of the new (trial) spin. Basically this technique works because the list of sites selected for spin flip changes depends only on the Hamiltonian. Therefore, because biased migration only changes the Hamiltonian a new or augmented list can be set up by using the gradient-dependent algorithm described in the previous section. This new list can then be used in the N-fold method, resulting in the gradient-dependent variation in grain growth as seen with the conventional Potts algorithm. The biased migration N-fold algorithm was implemented, and in fact was used to create Figure 23.

5.4. Conclusion of Biased Migration Potts-model Simulations

To investigate the columnar grain growth seen in reactor fuel, a gradient-dependent term was added to the Potts-model Hamiltonian; this represents an additional energy term caused by the presence of a thermal gradient. This approach was taken because it fit easily into the mobility model which indicates that the velocity of the grain boundary is proportional to its mobility times the driving terms. Under normal grain growth, the dominant driving term is curvature, which is included in the standard Potts-model Hamiltonian. An additional driving term was included to account for the gradient-

dependent vapor-transport. Clearly, other physical transport mechanisms can also be included (surface diffusion, volume diffusion, and migration of pores in stress gradients).

The Potts-model Hamiltonian was implemented by summing the number of like-spin neighbors around a site with the old spin and the new spin. If the difference was positive (negative energy change), then the flip to the new spin was made. Tests were conducted on a problem with a linear mobility gradient in the x direction, hottest at $x=0$. This normally resulted in the fastest growth and larger grains at the hot side, becoming smaller toward the cool side. The grains were all equiax, however.

The gradient term in the Hamiltonian was implemented simply by adding a 1 to the energy sum for like neighbors on the -x (hotter) side, or $\sqrt{2} / 2$ for the like neighbors in the diagonal (hot side) direction. When this Hamiltonian was used with the one-dimensional linear mobility gradient, the grain growth turned to columnar growth at the hottest side, equiax in the middle, and showed little growth at the low mobility (cold) side.

We also implemented an N-fold algorithm version of the biased migration simulation, which produced the same results as classical solution method.

The type of grain growth in uranium oxide fuel, although extremely similar in appearance to the results of the gradient Hamiltonian, is actually caused by a temperature and temperature gradient-dependent vapor transport mechanism across pores in the fuel. The pores grow and move up the temperature gradient, leaving a columnar structure behind them.

In experimenting with moving pores, it was found that the pores would move readily to the grain boundaries, and preferentially to the corners of the grains. However, moving across grains required extremely large gradient terms. It was felt that the pore movement in question could only be modeled using a Potts-model which included more realistic matching of various driving forces and with a finite temperature Potts-model.

References

- ¹ F. James, "A review of pseudorandom number generators", *Comp. Phys. Comm.* **60**, pg. 329 (1990).
- ² P. D. Coddington, "Analysis of Random Number Generators Using Monte Carlo Simulation", *Int. J. Mod. Phys. C5*, pg. 547 (1995).
- ³ S. J. Plimpton, R. M. Fye, and E. A. Holm, "Grain Growth Scaling from Large-scale Potts Simulations", in preparation.
- ⁴ A.B. Bortz, M.H. Kalos, and J.L. Liebowitz, *J. Comp. Phys.* **17**, pg. 10 (1975).
- ⁵ G.N. Hassold and E.A. Holm, *Comp. in Phys.* **7**, pg. 97 (1993).
- ⁶ B.D. Lubachevsky, *J. Comp. Phys.* **75**, pg. 103 (1988).
- ⁷ R. Fye, S. Plimpton, and E. Holm, "Fast Parallel Algorithm for Kinetic Potts-model Simulations", in preparation.
- ⁸ A similar idea was discussed in P.S. Sahni, D.J. Srolovitz, G.S. Grest, M.P. Anderson, and S.A. Safran, *Phys. Rev. B* **28**, pg. 2705 (1983).
- ⁹ Also used for a different algorithm by B. Rhadakrishnan and T. Zacharia, *Met. and Mat. Trans. A* **26**, pg. 167 (1995).
- ¹⁰ J.C. Fisher, *J. Appl. Phys.* **22**, 74 (1951).
- ¹¹ I. Kaur, W. Gust, *Fundamentals of Grain and Interphase Boundary Diffusion*, 2nd ed. (Ziegler Press, Stuttgart, 1989).
- ¹² R.T.P. Whipple, *Phil. Mag.* **45**, 1225 (1954).
- ¹³ Y.M. Mishin, *Phys. Stat. Sol. A* **133**, 259 (1992).
- ¹⁴ G.H. Gilmer and H.H. Farrell, *J. Appl. Phys.* **47**, 3792 (1976).
- ¹⁵ G.H. Gilmer and H.H. Farrell, *J. Appl. Phys.* **47**, 4373 (1976).
- ¹⁶ H.S. Levine and C.J. MacCallum, *J. Appl. Phys.* **31**, 595 (1960).
- ¹⁷ J.P. Levine and D.L. Lossee, *J. Appl. Phys.* **56**, 924 (1983).
- ¹⁸ P.H. Holloway, D.E. Amos, and G.C. Nelson, *J. Appl. Phys.* **47**, 3769 (1976).
- ¹⁹ X. Gui, S.K. Dew, M.J. Brett, and D. de Cogan, *J. Appl. Phys.* **74**, 7173 (1993).
- ²⁰ J. Crank, *The Mathematics of Diffusion*, 2nd. ed. (Clarendon Press, Oxford, 1975).
- ²¹ M.P. Anderson, G.S. Grest, and D.J. Srolovitz, *Phil. Mag. B* **59**, 293 (1989).
- ²² E. A. Holm, J. A. Glazier, D. J. Srolovitz, and G. S. Grest, *Phys. Rev. A* **43**, 2662 (1991).
- ²³ Donald R. Olander, Fundamental Aspects of Nuclear Reactor Fuel Elements, Univ. of California, Berkeley, TID-26711-P1, 1976.
- ²⁴ Elizabeth A. Holm, N. Zacharopoulos, and D. J. Srolovitz, "Grain Boundary Mobility Effects", SAND94-1749J.

Distribution:

Copies	Mail Stop	Name	Org.
1	9018	Central Technical Files	8940-2
5	0899	Technical Library	4916
2	0619	Review and Approval Desk For DOE/OSTI	12690
1	0188	Donna Chavez	4523
1	0321	William J. Camp	9200
1	0333	Alan J. Hurd	1841
1	0333	Roy J. Bourcier	1841
1	0513	R. J. Eagan	1000
1	0820	Eliot Fang	9232
1	0835	Ahti J. Suo-Anttila	9363
10	1111	Steven J. Plimpton	9221
1	1111	Sudip S. Dosanjh	9221
1	1111	Grant Heffelfinger	9225
10	1111	Richard M. Fye	9225
10	1139	Michael F. Young	6421
1	1139	Al Reed	6421
1	1141	Susan C. Bourcier	9361
1	1145	Paul S. Pickard	9360
10	1146	Steven A. Wright	9363
1	1146	Theodore F. Luera	9363
2	1144	Larry Compton	9360
1	1175	LeAnn Miller	9364
10	1349	Thomas P. Swiler	1846
1	1405	Veena Tikare	1841
10	1405	Elizabeth A. Holm	1841