Simulation of Curvature-Driven Grain Growth by Using a Modified Monte Carlo Algorithm

B. RADHAKRISHNAN and T. ZACHARIA

The Monte Carlo (MC) algorithm that currently exists in the literature for simulating curvature-driven grain growth has been modified. The modified algorithm results in an acceleration of the simulated grain growth and an early estimate of the grain growth exponent that is close to the theoretical value of 0.5. The upper limit of grain size distributions obtained with the new algorithm is significantly lower than that obtained with the old, because the new algorithm eliminates grain coalescence during grain growth. The log-normal function provides an excellent fit to the grain size distribution data obtained with the new algorithm, after taking into account the anisotropy in grain boundary energy.

I. INTRODUCTION

THE use of Monte Carlo (MC) technique to simulate grain growth evolved from the extension of the Ising model[1,2] that was used for simulating systems with two-fold degeneracy to systems having infinite degeneracy. Grain structures in metallic and ceramic systems exhibit infinite degeneracy, since there can be a large number of grain orientations in a finite volume of material. However, it was shown[3] in MC simulations that when the number of degenerate states was increased from a value of 2 (Ising model) to a number typically greater than 36, the simulation results approached the behavior of systems with infinite degeneracy. Grain growth simulations that use such a model have been well documented in the literature.[3–17] The simulations, when extended to two-phase alloys, in which the second phase was a dispersion of insoluble particles[7] predicted the stabilization of grain size by the pinning of the boundaries by the insoluble particles. The technique has also been used to simulate abnormal grain growth,[9,10] and primary recrystallization.[10]

A. *The MC Grain Growth Algorithm*

The algorithm used for simulating curvature-driven grain growth[3–17] is as follows. A simulation domain of a suitable size (200 × 200) is chosen. Each of the points in the domain is assigned a random number S_i between 1 and N_Q, where N_Q is the total number of grain orientations. A grain is defined by a collection of points that have the same orientation number. N_Q is typically greater than 36, above which the grain growth exponent becomes independent of N_Q.[3,5] Each of the points is then accessed randomly, and the energy change associated with switching the orientation to another random value between 1 and N_Q is calculated. The energy change is calculated by measuring the local energy of a cluster of lattice points before and after the switching process. The energy of the cluster of lattice points in defined as

$$E = -J\left[\sum_{nn} \delta_{S_iS_j} - 1\right] \qquad [1]$$

where J is a constant that is proportional to the grain boundary energy, δ_{ij} is the Kronecker δ function, S_i is the orientation of the lattice point whose orientation change is being attempted, and S_j is the orientation of a nearest neighbor. The summation is taken over all the nearest neighbors. The probability of switching is based on the magnitude of the net energy change ΔE as a result of the switching process. The probability of switching is defined as follows:

$$p = \begin{cases} 1 & \Delta E \leq 0 \\ \exp(-Q/RT) & \Delta E > 0 \end{cases} \qquad [2]$$

The quantity T in Eq. [2] is not related to the physical temperature but is a quantity that is introduced to take care of certain lattice effects to be described subsequently. The simulation time is defined by a quantity known as the Monte Carlo step (MCS), which is related to the number of reorientation attempts. The MCS is equal to 1 when the number of attempts equal the number of points in the domain. It is related to the real time through a temperature-dependent jump frequency. Grain size in the simulations is calculated either from the mean of the number of lattice points within a grain, which is proportional to the grain area in two-dimensional (2-D) simulations, and the grain volume in three-dimensional (3-D) simulations, or from linear intercept measurements carried out on grain structures obtained in 2-D simulations and on cross sections of grain structures obtained in 3-D simulations. Grain growth kinetics are followed by plotting the grain size as a function of the MCS. The grain growth exponent n is calculated from the "long-time" slope of the log-log plot of the grain size *vs* the MCS.

The 2-D and 3-D simulations have been carried out by previous investigators on a variety of lattice models. Two-dimensional simulations used either a triangular lattice, in which the local energy was calculated by using a shell consisting of six first-nearest-neighbor lattice points,[3–10] or a square lattice, in which the local energy change was calculated[3,4,5] by using either a shell of four first-nearest-neighbor lattice points (SQ 1) or a shell of eight lattice points[14] containing the first- and second-nearest neighbors (SQ 2). Three-dimensional simulations were carried out by using a cubic lattice with a shell of 26 lattice points containing the first, second, and third nearest neighbors.[10–15]

B. RADHAKRISHNAN, Research Staff Member, and T. ZACHARIA, Group Leader, are with the Materials Process Modeling Group, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6140.

Manuscript submitted March 2, 1994.

Earlier simulations in which square, triangular lattices[3,4,5] were used showed that grain growth kinetics was a function of the lattice type. In SQ 1 lattice, for example, the grain growth completely ceased unless an artificial "thermal" effect was introduced into the simulations to allow for certain fluctuations that would perturb the system into local, higher-energy states. The quantity T in Eq. [2] indicates this effect. In fact, it was shown that in SQ 1 lattice, the so-called "temperature" had to be increased to a value close to the melting point to make the grain growth go to completion. However, the triangular lattice showed a grain growth rate that was independent of temperature. Hence, no thermal activation was necessary to complete the grain growth. A similar result was also obtained for the SQ 2 lattice. Hence, for lattices that do not curb grain growth because of certain geometric restrictions, the following probability considerations can be used to carry out the simulations:

$$p = \begin{cases} 1 & \Delta E \leq 0 \\ 0 & \Delta E > 0 \end{cases} \qquad [3]$$

B. *Survey of Simulation Results*

1. *Grain growth exponent*

In the simulations that first appeared in the literature,[3-13] the grain growth exponent was found to be significantly lower than that obtained for the Ising model. The simulations predicted an exponent of 0.41 ± 0.03 for grain growth in metallic and ceramic systems. Grain growth exponents were obtained from the long-time slope of the log-log plots of grain size *vs* the MCS. However, in the case of the Ising model, in which only two degenerate states exist, the exponent was approximately equal to 0.5, the theoretical limit for curvature-driven growth. This difference was believed to be a consequence of the presence of vertices in a topologically connected grain structure and the absorption of curvature at these vertices.[3,4,5] More recent simulations[14,15,16] carried out on bigger domains and extended to vary large growth periods showed that the growth exponent, at least in 2-D, did approach the theoretical limit of 0.5. The domain sizes used in 3-D simulations so far have not been large enough to predict the limiting growth exponent of 0.5.

2. *Grain size distribution*

Grain size distribution functions were obtained from both 2-D[4] and 3-D[15] simulations, by plotting the frequency of occurrence defined as the number of grains belonging to a size-class relative to the total number of grains as a function of the normalized grain size $R/\bar{R}$, where R is the size class and $\bar{R}$ is the average grain size. Grain sizes were calculated by measuring three different parameters: linear intercepts, grain areas, and grain volumes. The distribution curves were characterized by a sharp rise to a peak size fraction, followed by a gradual decay. The maximum size fraction occurred in the $R/\bar{R}$ range of 0.5 to 1.0, and the maximum grain size was in the $R/\bar{R}$ range of 3 to 3.5. The best fit to the simulated grain size distribution curve when the grain sizes were obtained from grain volume data was given by the log-normal function.[15] However, when the linear intercept or grain area measures were used to calculate the grain sizes, the best fit was obtained by using a distribution function proposed by Louat.[18] Grain size distributions obtained from earlier 2-D simulations[4,6,10] also matched closely the size distributions measured by using 3-D cross-sectional data.[15] Hence, it follows that the distribution function proposed by Louat should also provide the best fit to the data obtained from 2-D simulations. The distribution function proposed by Hillert[19] was found to be inappropriate, because the function predicted an upper limit to the grain size given by $R/\bar{R} = 1.8$, which was very much lower than the simulated value of 3 to 3.5. Also, the peak of Hillert's function occurred at a higher grain size than in the simulations. The simulated grain size distributions obtained by using cross-sectional grain area data of 3-D simulations and grain area data from 2-D simulations closely matched the distribution curves obtained experimentally for high-purity Fe and Al-Mg alloy.[15]

3. *Anisotropic grain boundary energy*

The simulation results described in Sections 1 and 2 were obtained under the assumption that the grain boundary energy was isotropic. However, in real materials, the grain boundary energy is anisotropic, because it depends on the misorientation between the two adjacent grains. The effect of grain misorientation on grain boundary energy in 2-D simulations was related to an anisotropy parameter,[8] which was proportional to the difference between two adjacent grain orientations. The analysis indicated that the presence of anisotropy in grain boundary energy lowered the grain growth exponent compared with the isotropic case and broadened the grain size distribution curve.[8]

II. LIMITATIONS OF THE EXISTING GRAIN GROWTH ALGORITHM

This section describes the limitations of the MC grain growth algorithm that currently exists in the literature. The limitations include (1) the prediction of a low value for the grain growth exponent at reasonable grain sizes, (2) the occurrence of grain coalescence, and (3) the occurrence of grain nucleation.

A. *Low Grain Growth Exponent*

As explained previously, simulations with the old algorithm result in the prediction of a significantly lower value for the grain growth exponent than the theoretical limit of 0.5 at reasonable grain sizes. This section describes why such a deviation occurs.

Curvature-driven grain growth satisfies an equation of the form

$$D^m - D_0^m = k(t - t_0) \qquad [4]$$

where D is the grain size at time t; D_0 is the grain size at time t_0; m is a constant ($1/m$ is the grain growth exponent, n) that depends upon the purity, texture, and anisotropy in grain boundary energy; and k is the mobility that is temperature dependent. The mobility, k, can be represented as

$$k = k_0 \exp\left(-\frac{Q}{RT}\right) \qquad [5]$$

where k_0 is a constant, T is the absolute temperature, and Q is the activation energy for grain growth. The constant k_0 can be expressed as

$$k_0 = C_1 a^2 \nu \frac{C_4 J}{\lambda T} \qquad [6]$$

where C_1 is a constant, ν is the atomic jump frequency, a is the atomic jump distance, C_4J is the grain boundary energy, λ is Boltzman's constant, and T is the absolute temperature. If the MC grain size D_{MC} is related to the real grain size D_R as

$$D_R = aD_{\text{MC}} \qquad [7]$$

where a is the atomic size, Eqs. [5], [6], and [7] can be substituted into Eq. [4] to yield

$$D^m_{\text{MC}} - D^m_{0_{\text{MC}}} = C_1 \frac{a^2}{a^m} \nu \frac{C_4 J}{\lambda T} \exp\left(-\frac{Q}{\text{R}T}\right)(t - t_0) \qquad [8]$$

The MCS is related to the real time t as[17,20]

$$\text{MCS} = \nu t \exp\left(-\frac{Q}{\text{R}T}\right) \qquad [9]$$

It is known that for an ideal material (no solute-drag, texture etc.), m has a value of 2.0. Hence, Eq. [8] can be simplified as

$$D^2_{\text{MC}} - D^2_{0_{\text{MC}}} = C_1 \frac{C_4 J}{\lambda T}(\text{MCS} - \text{MCS}_0) \qquad [10]$$

It can be seen from Eq. [10] that the quantity $C_1(C_4J/\lambda T)$ represents the "mobility" term in MC simulations. The constant C_1 can be represented as a product of two constants, C_2 and C_3. The term C_2 is a geometric constant that relates the increase in the grain area (in 2-D simulations) and grain volume (in 3-D simulations) as a result of the atomic jump into an effective increase in grain size. The term C_3 is a probability that the atomic jump is directed toward one of the nearest-neighbor atoms that belongs to an adjacent grain. In the MC simulation, an atomic jump is equivalent to the reorientation attempt at an MC lattice site described previously.

In the old algorithm, the reorientation attempt at any point includes all orientations, 1 through N_O, in the simulation domain. Hence, the constant C_3 should be proportional to $N_L/(N_Q - 1)$, where N_L is the total number of nearest-neighbor orientations that are different from that of the central site and N_Q is the total number of initial orientations. For example, in a triangular lattice with six nearest neighbors, N_L varies from 6 to 1, depending on the grain size and the location of the MC site. At the beginning of grain growth, when the orientations of the MC sites are randomly distributed and each site effectively represents a grain, N_L is equal to 6. However, after sufficient grain growth has taken place, N_L reduces to either 1 or 2, depending on whether the site is close to a grain boundary or a triple point. As further grain growth occurs, the fraction of triple point sites to grain boundary sites decreases and N_L slowly tends toward a constant value of 1. The underlying reason for the low grain growth exponent and its asymptotic increase to the theoretical limit in MC simulations is that grain growth occurs in the presence of decreasing boundary mobility, as explained in the following paragraph.

Figure 1(a) shows a set of grain growth curves for mobilities ranging from k_1 to k_6, where k_1 is the highest and k_6 is the lowest. The solid curve shows the effective grain growth kinetics that will be observed when the grain growth starts with the mobility k_1, continues in the presence of decreasing mobility, and ends with the mobility k_6. The effective grain growth curve was computed with an initial k_1 value of 2.0, which was allowed to change asymptotically to the final k_6 value of 1, after an MCS of 1000. Such a variation is typical in MC simulations using the old algorithm, as explained in the previous paragraph. Figure 1(b) shows the log-log plot of grain size *vs* the MCS for the effective grain growth kinetics curve shown in Figure 1(a). Notice that the effective grain growth exponent remains lower than the theoretical value as long as the mobility is decreasing. Once the lowest mobility is reached the exponent becomes equal to the theoretical limit, and grain growth continues along the curve for which the mobility is k_6.

B. *Grain Nucleation*

One of the potential sources of error in the old algorithm is grain nucleation, which can occur especially when the anisotropy in grain boundary energy is considered. The variation in grain boundary energy with misorientation involves the introduction of an energy function[8] defined as

$$E_i = \sum_j^{nn} V(S_i, S_j) \qquad [11]$$

where the summation is taken over the nearest neighbors, S_j is one of the nearest-neighbor orientations, and V is a local anisotropy potential. The local anisotropy potential is related to the misorientation between two adjacent grains, which is given by:[8]

$$\theta_{ij} = 2\pi \frac{(S_i - S_j)}{Q} \qquad [12]$$

The local anisotropy potential is given by

$$V(\theta) = \begin{cases} J\dfrac{\theta'}{\theta^*}\left[1 - \ln\left(\dfrac{\theta'}{\theta^*}\right)\right] & \theta' < \theta^* \\ J & \theta' \geq \theta^* \end{cases} \qquad [13]$$

where θ^* is the anisotropy parameter, $\theta' = |\theta|$, for $0 \leq |\theta| \leq \pi$, and $\theta' = 2\pi - |\theta|$, for $\pi \leq |\theta| \leq 2\pi$. The term θ' is the misorientation angle above which the grain boundaries are considered to be high angle. Now, let us use the old algorithm and apply Eqs. [11] through [13] to calculate the local energy changes in the presence of anisotropy in grain boundary energy. Figures 2(a) and (b) show the orientations of a cluster of points before and after the orientation of the central point is changed from 1 to 11. For an anisotropy parameter of 0.3π and a Q value of 64, the initial energy of the cluster is $4J$. After the orientation of the central points is changed to

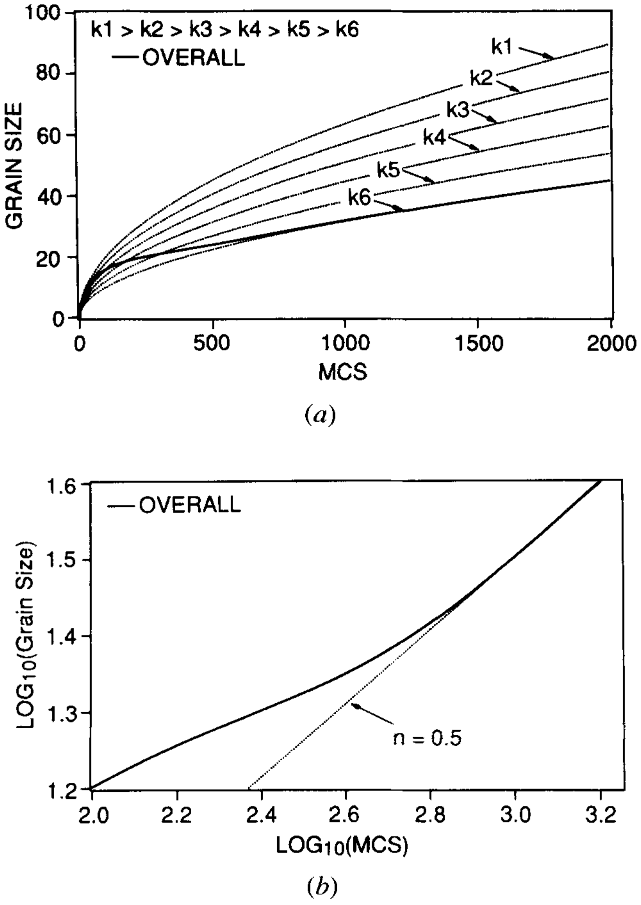


Fig. 1—Grain growth in the presence of decreasing grain boundary mobility. The solid curve in (*a*) represents the effective grain growth kinetics, when the mobility decreases from K_1 to k_6 during grain growth. The log-log plot of grain size *vs* the MCS of the solid curve in (a) is shown in (*b*). Notice that the grain growth exponent is lower than the theoretical limit of 0.5, as long as the mobility is decreasing.

11, the energy of the cluster is 3.75J. Because the orientation change results in a net decrease of energy, it would be successful. However, it results in grain nucleation, which does not occur during normal grain growth.

C. *Grain Coalescence*

According to the old algorithm, the total number of grain orientations N_Q need only be greater than a finite positive value (typically 36) to prevent the coalescence of grains, which does not occur during normal grain growth. The probability of coalescence is estimated to be $1 - (1 - 1/Q)^Z$,[5] where Z is the number of second-nearest-neighbor grains and Q is the total number of orientations. In 2-D simulations, because an average grain has six sides, there will be 12 second-nearest-neighbor grains. For an N_Q value of 64, for example, the probability of grain coalescence is about 0.17, which is not negligible. In 3-D simulations, in which an average grain has 14 sides, the number of second-nearest-neighbor grains on average is 28, resulting in a probability of coalescence of 0.36, which is quite significant. A careful examination of the simulation results documented in the literature showed that some grain coalescence did indeed occur during grain growth.

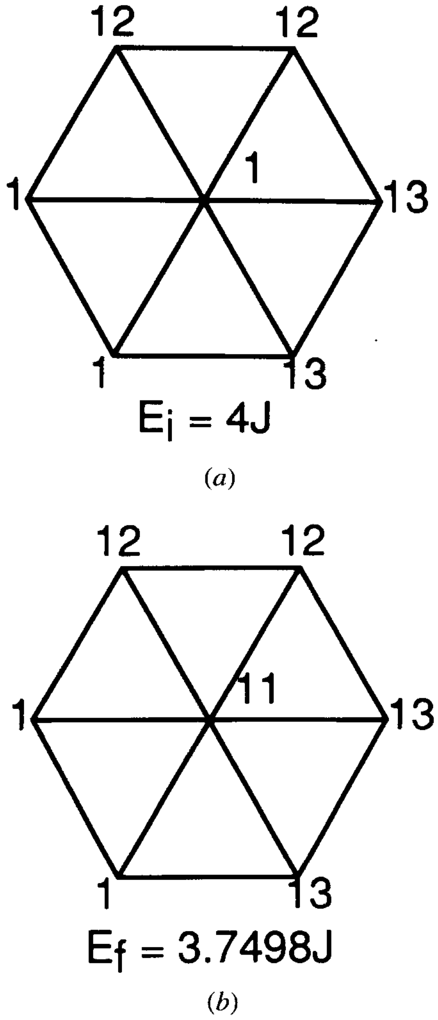


Fig. 2—The potential for grain nucleation during grain growth in the presence of anisotropy in grain boundary energy using the old algorithm. Using $Q = 64$ and $\theta^* = 0.3\pi$, the change in energy, $\Delta E = (E_f - E_i)$, when the orientation of the central site is changed from 1 in (*a*) to 11 in (*b*), is negative. Hence, the orientation change is allowed, even though it leads to grain nucleation, which does not occur during grain growth.

III. PROPOSED NEW ALGORITHM

In the original development of the MC algorithm, the individual lattice points are considered as microcrystals or atom clusters. A probabilistic change in the occupancy of an atom cluster close to a grain boundary from an existing grain to an adjacent grain results in the migration of the boundary. However, in a real material, the migration of grain boundaries occurs as a result of thermally activated atomic jumps across a boundary. The change of a cluster of atoms close to a grain boundary from one grain to another is a cumulative change in the occupancy of each of the atoms belonging to the cluster, as boundary motion occurs following each atomic jump. The driving force for the atomic jumps is a decrease in the local energy state of an atom as it jumps from the concave side of a highly curved boundary to the convex side, which causes a reduction in the curvature of the boundary. The vibrational modes of atoms can be considered as linear oscillations in which the number of degrees of freedom is determined by the number of nearest neighbors. Hence, atomic jumps should always be directed toward one of the nearest neighbors. If the activation barrier can be surmounted with the available thermal energy, the atom can then lower the energy by establishing a bond with the neighboring atom toward which the jump is directed. Grain boundary migration occurs if the successful jump is toward a nearest-neighbor atom that belongs to an adjacent grain. Hence, from a fundamental mechanistic point of view, only interactions with the nearest-neighbor atom clusters should be considered when a change of occupancy of an atom cluster is attempted. Hence, the new attempted orientation should be limited to one of the nearest-neighbor orientations and should not include all possible orientations in the system, as the old algorithm did.

The suggested modification should result in a grain boundary mobility that is independent of the total number of orientations in the system. Since the boundary mobility is not a function of the total number of orientations, the total number of initial orientations can be as high as the total number of sites within the domain so that no coalescence can occur at any stage of grain growth. (In the old algorithm, the mobility was inversely proportional to the number of orientations in the system. The number of orientations had to be kept finite so that the grain boundaries had a reasonable mobility.) The random number generator generates a number between 1 and N_N at every point, where N_N is the number of nearest neighbors. The reorientation attempt is directed to the orientation of the nearest neighbor defined by the random number 1 through N_N.

In the new algorithm, the probability that a reorientation is directed toward a nearest neighbor belonging to an adjacent grain would, as in the old algorithm, also depend on whether the MC site under consideration is close to a triple point or far away from it. Hence, the average mobility would still be a function of grain size, as in the old algorithm. However, the ratio of the mobility of triple point sites to that of grain boundary sites is 2 for the old algorithm, while it is ≈ 1.5 for the new algorithm. Hence, the average mobility as a function of grain size should decay to a constant value much faster in the new algorithm than in the old algorithm. Consequently, the theoretical limit of the grain growth exponent should be reached much faster in the new algorithm than in the old algorithm. This point is illustrated through actual simulations that are discussed in Section VI.

The proposed modification should also eliminate the problem of grain nucleation when the anisotropy in grain boundary energy is taken into consideration. Because only the nearest-neighbor orientations are considered during a reorientation attempt, the grain nucleation that occurred with the old algorithm (Figure 1) will be totally eliminated in the new algorithm.

IV. SIMULATIONS

The new algorithm had to be tested on problems of known solution to see if the predictions of curvature-driven growth would be confirmed by the proposed algorithm. One such problem, which is a special case of the von Newmann law,[21,22] is the shrinking of a sphere (in three dimensions) or a circle (in two dimensions) in an infinite matrix, demonstrated by Anderson *et al.*[5] The 2-D simulations of a shrinking circle in an infinite matrix were carried out for old and new algorithms, with the same initial diameter and starting seed for the random number generator for both.

The modifications described in Section III were implemented in 2-D and 3-D simulations of grain growth in single-phase materials. A triangular lattice[3-10] of 200 × 200 points was used for 2-D simulations, and a cubic lattice of 50 × 50 × 50 points was used for the 3-D simulations. The simulations were carried out according to the switching probabilities given by Eq. [3] ($T = 0$ simulation).

The following simulations were carried out.

(1) The old algorithm was run on the 200 × 200 triangular lattice with 64 initial grain orientations. The purpose of these simulations was to generate baseline data against which the results of the new algorithm could be compared.

(2) The old algorithm was modified so that the reorientation attempts were restricted to the orientations of the nearest neighbors only. The total number of initial orientations in the system was still 64, the same as in (1). The purpose of these simulations was to determine the change in the grain growth characteristics introduced by restricting the reorientation attempts to the orientations of the nearest neighbors. These simulations will hereafter be referred to as "modified old" algorithm.

(3) The "new" algorithm was run on the 200 × 200 triangular matrix. In the new algorithm, the total number of initial orientations was 200^2. The reorientation attempts were restricted to the orientations of the nearest neighbors. The only difference between the modified old and new algorithms was in the total number of initial orientations.

(4) The new algorithm was also run on a 200 × 200 triangular lattice, with the condition that the grain boundary energy was not isotropic. The use of the new algorithm for simulating the effect of anisotropy in grain boundary energy should eliminate the occurrence of grain nucleation, because the reorientation attempts cover only the nearest-neighbor orientations. The methodology used for treating grain boundary energy anisotropy was identical to that in the literature,[8] described previously.

(5) The 3-D simulations on the 50 × 50 × 50 cubic lattice were carried out by using 50^3 initial orientations for the new algorithm and 64 orientations for the old algorithm. The purpose of these simulations was to compare the grain growth characteristics in 2-D and 3-D cases obtained by using the old and new algorithms.

In the previous studies[14] the grain growth exponent was obtained from the long-time slope of the log-log plots of grain size *vs* the MCS. The long-time slope is used because Eqs. [8] and [9] when $D \gg D_0$ and MCS $\gg$ MCS$_0$, can be simplified as $D_{\text{MC}} = k(\text{MCS})^{1/m}$, so that a log-log plot of D_{MC} *vs* the MCS would yield a straight line of slope $(1/m)$. However, the use of long-time slope is not satisfactory, because it can fail to detect the presence of curvature in the plots due to other fundamental reasons. For example, the slow, asymptotic increase in the grain growth exponent with grain size was not detected until grain growth simulations were extended to much larger grain sizes for which bigger simulation domains were used. The grain growth exponent can also be obtained by plotting the quantity $D^m - D_0^m$ *vs* (MCS − MCS_0), where D is the grain size at time t, D_0 is the grain size at time MCS_0, and m is a constant that can be varied so that a straight line passing through the origin is obtained. However, in the MC simulations, the quantity D_0 is not known. Because the initial grain size distribution resulting from assigning a random number to each lattice site is far removed from the equilibrium grain size distribution, it is not reasonable to assume that D_0 is the grain size at the beginning of the simulations. However, the grain size distribution stabilizes after a certain amount of grain growth, and the value of D_0 corresponds to the grain size at which the grain size distribution has just stabilized.

The grain growth exponent in the present simulations was obtained by fitting to the log-log plot of grain size *vs* the MCS an equation of the form

$$y = p_1 x + p_2 \exp(-p_3 x) + p_4 \qquad [14]$$

where p_1, p_2, p_3, and p_4 are constants, y is log (grain size), and x is log (MCS). The grain growth exponent at a given value of x is obtained by the derivative of Eq. [14] given by

$$\frac{dy}{dx} = p_1 - p_2 p_3 \exp(-p_3 x) \qquad [15]$$

Notice that the slope given by Eq. [15] increases continuously and reaches a steady-state value when x becomes very large. The grain size for these plots was obtained by linear intercept measurements. Each simulation was repeated five times, and in each simulation, different seed values for the random number generator were used; the results given are the average of the five different runs. Grain size distribution curves were obtained by calculating the number fraction of grains belonging to a size interval and plotting this fraction against the midpoint of the corresponding size interval. The grain size for these plots was calculated from the grain areas in the 2-D case and from grain volumes in the 3-D case. The grain areas and grain volumes in 2-D and 3-D cases were obtained by computing the total number of points within each grain. In the 2-D case, the grain size was the square root of the grain area, whereas in the 3-D case, the grain size was the cube root of the grain volume. Grain size distributions were obtained at an MCS value of 5000 for the old algorithm, corresponding to a grain size ≈10. The distributions were obtained at the same MCS value for five different seed values, and the result shown is the average of the five runs. Grain size distributions for the modified old and new algorithms were obtained at an identical average grain size by using the different seed values, and an average distribution curve was obtained for each case.

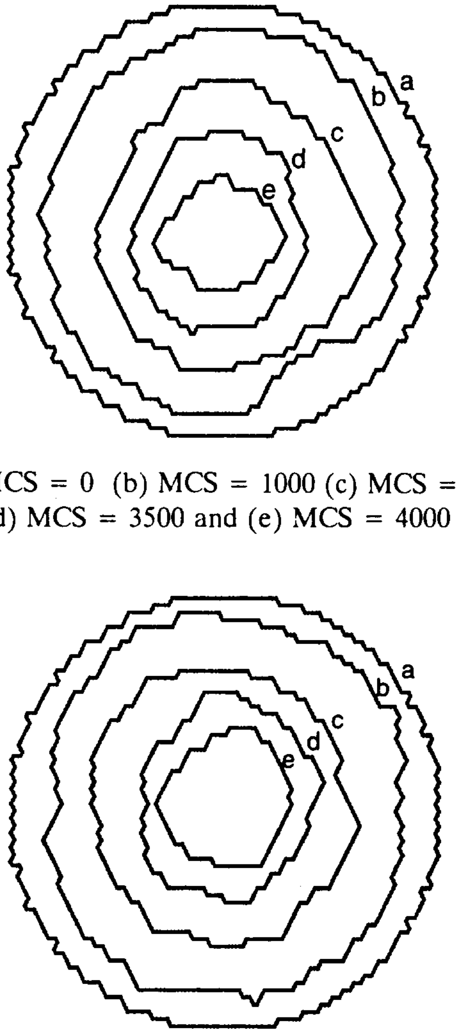
The potential for grain coalescence, when the number of grain orientations in the system was finite, was examined by plotting the ratio $R_{\max}/\bar{R}$ against the MCS. The occurrence of grain coalescence was marked by a sudden increase in the ratio. The occurrence of grain coalescence was monitored for the same grain growth range by using the old, modified-old, and new algorithms.

V. RESULTS

Initially, the simulation results for the shrinking of a circular grain in an infinite matrix will be compared for the old and new algorithms, in order to check the validity of the new algorithm for curvature-driven grain growth. Subsequently, the simulation results of grain growth exponent, grain size distribution, and the occurrence of grain coalescence will be presented for 2-D and 3-D simulations with the old, modified-old, and new algorithms. The effect of anisotropy in grain boundary energy on the grain growth exponent and the grain size distribution in 2-D simulations will be presented for the new algorithm.

A. *Shrinking of Circular Grain in an Infinite Matrix*

Figure 3 shows the temporal evolution of a circular grain in an infinite matrix for the old and new algorithms. In both cases, the shrinking grain remains circular, except for statistical fluctuations. Figure 4 shows



(a) MCS = 0 (b) MCS = 1000 (c) MCS = 2500
(d) MCS = 3500 and (e) MCS = 4000

(a) MCS = 0 (b) MCS = 100 (c) MCS = 300
(d) MCS = 400 and (e) MCS = 450

Fig. 3—Temporal evolution of a circular grain in an infinite matrix for old (top) and new (bottom) algorithms. In both cases, the shrinking grain remains almost circular, except for statistical fluctuations.

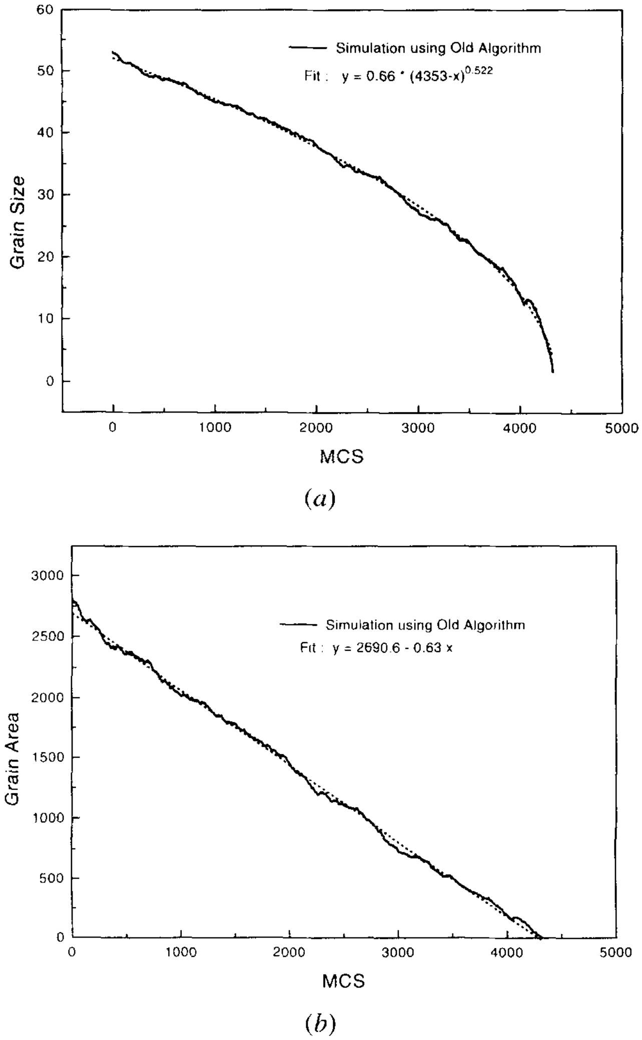


Fig. 4—(*a*) Average size and (*b*) area of the circular grain shown in Fig. 3 as a function of the MCS for the old algorithm.

the grain size and the grain area as a function of the MCS for the old algorithm, and Figure 5 shows the same for the new algorithm. The time exponent is close to the theoretical value of 0.5 for both cases. Also, the grain area linearly decreases with the MCS for both algorithms. It is clear that the new algorithm essentially reproduces the result obtained for the old algorithm and, hence, clearly depicts curvature-driven grain growth.

B. *Isotropic Grain Boundary Energy*

1. *Grain growth exponent*

Figures 6 through 8 show the log-log plot of grain size *vs* the MCS for 2-D grain growth obtained by using the old, modified-old, and new algorithms. The grain growth exponents at various grain sizes obtained from the fitting functions are shown in Table I. The grain growth exponents shown in Table I for the old algorithm are consistent with the results available in the literature.[3,5,10] However, the results presented in Table I may be more accurate than the values in the literature,[3,5,10] because the fit obtained using Eq. 14 is better than the straight line fit to the long-time data presented in the literature.

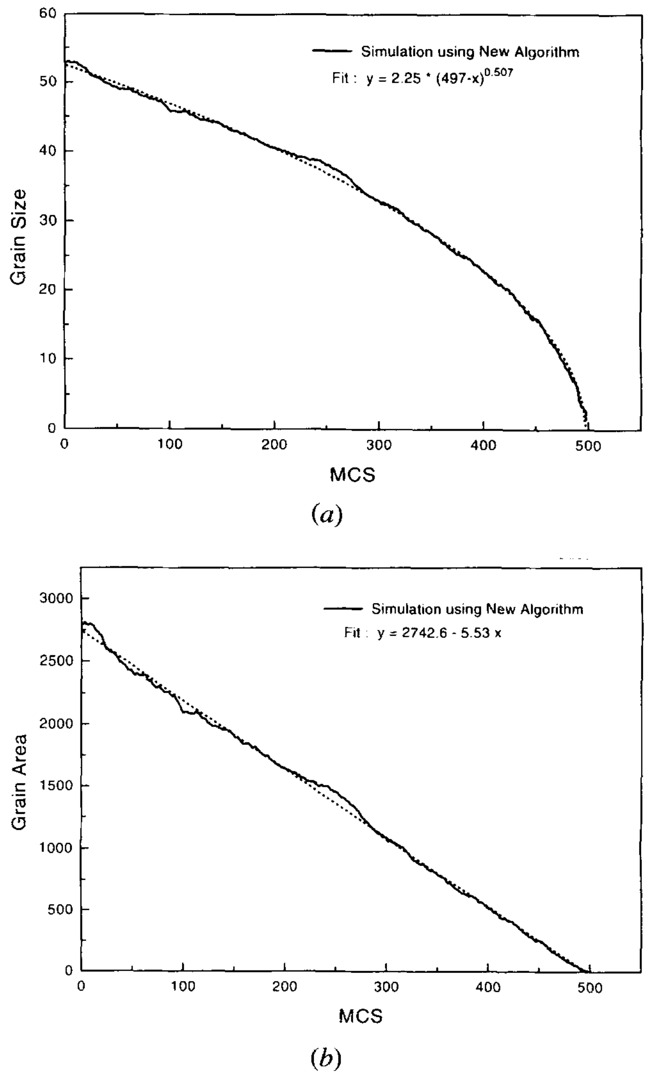


Fig. 5—(*a*) Average size and (*b*) area of the circular grain shown in Fig. 3 as a function of the MCS for the new algorithm.

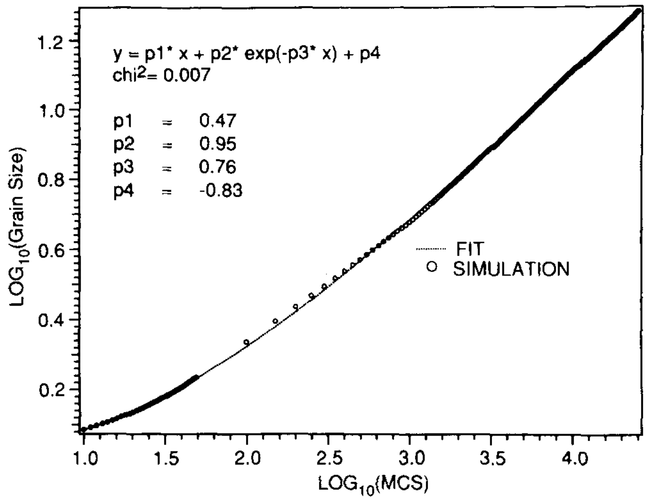


Fig. 6—Log-log plot of grain size *vs* the MCS in 2-D simulations with the old algorithm. The dotted line is a fit to the simulation data using the function shown.

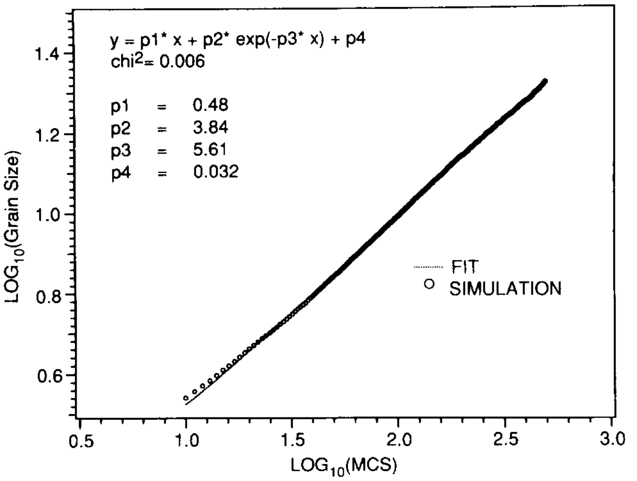


Fig. 7—Log-log plot of grain size *vs* the MCS in 2-D simulations with the modified-old algorithm. The dotted line is the fit to the simulation data using the fitting function shown.

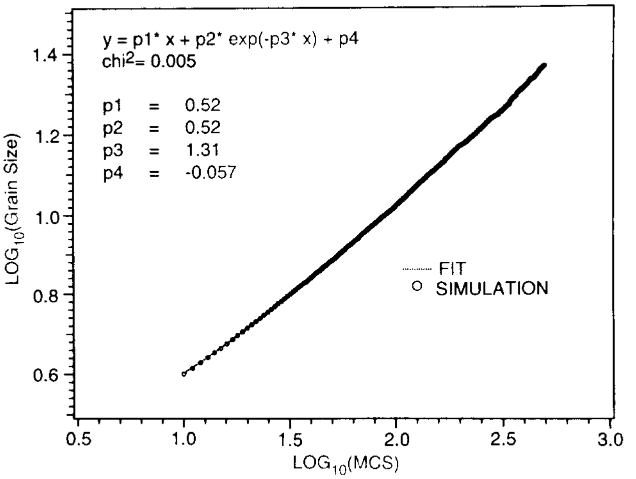


Fig. 8—Log-log plot of grain size *vs* the MCS in 2-D simulations with the new algorithm. The dotted line is the fit to the simulation data using the fitting function shown.

Table I. Variation of the Grain Growth Exponent with Grain Size for the Old, Modified-Old, and New Algorithms

Algorithm	Grain Size	Grain Growth Exponent
New	15.8	0.49
	10.0	0.47
Modified-old	15.8	0.48
	10.0	0.48
Old	15.8	0.44
	10.0	0.43

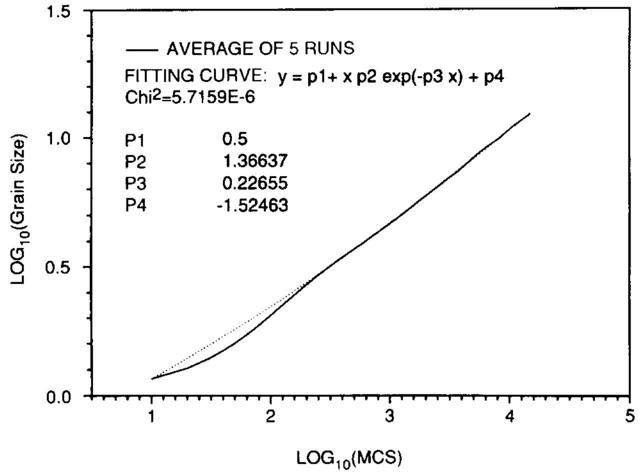


Fig. 9—Log-log plot of grain size *vs* the MCS in 3-D simulations with the old algorithm. The dotted line is the fit to the simulation data using the fitting function shown.

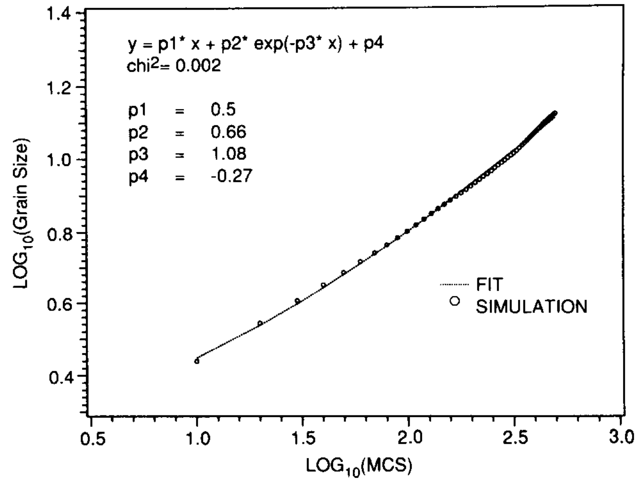


Fig. 10—Log-log plot of grain size *vs* the MCS in 3-D simulations with the new algorithm. The dotted line is the fit to the simulation data using the fitting function shown.

Table II. Variation of Grain Growth Exponent with Grain Size in 3-D Simulations Using the Old and the New Algorithms

Algorithm	Grain Size	Grain Growth Exponent
New	12.6	0.46
	10.0	0.45
Old	12.6	0.38
	10.0	0.37

The grain growth exponents for the modified-old and new algorithms are significantly higher than those of the old algorithm at corresponding grain sizes. Figures 9 and 10 show the log-log plot of grain size *vs* the MCS for 3-D grain growth using the old and new algorithms. Table II shows the grain growth exponents in 3-D simulations using the two algorithms at various grain sizes. Notice that the grain growth exponent obtained by using the old algorithm is again significantly lower than that obtained by using the new algorithm at corresponding grain sizes. Also, for the old algorithm, the grain growth exponents obtained for 3-D simulations are significantly lower than those obtained for 2-D simulations at identical grain sizes, as seen by comparing Tables I and II.

2. *Grain size distribution*

Figures 11 and 12 show the grain size distribution curves obtained in 2-D simulations with the old and new algorithms, respectively, at an identical mean grain size. The grain size distribution for the modified-old algorithm was identical to the one obtained with the old algorithm. The grain size distributions corresponding to the old and modified-old algorithms are characterized by an upper limiting grain size given by $R/\bar{R} \approx 3.0$ and a peak size fraction of $\approx$0.20 occurring at an $R/\bar{R}$ value of $\approx$0.6. However, the grain size distribution curve obtained by using the new algorithm is distinctly different from those obtained by using the old and modified-old algorithms. The upper limiting grain size obtained by using the new algorithm is $R/\bar{R} \approx 2.2$, which is significantly lower than the value of 3.0 obtained for the old and modified-old algorithms. Both the Louat function and the log-normal function were fitted to the 2-D grain size distributions corresponding to the old and new algorithms. The Louat function was obtained by using

$$y = p_1 x \exp(-p_2 x^2) \qquad [16]$$

where p_1 and p_2 are constants. The log-normal function was obtained by using

$$y = \frac{a_3}{a_4 \pi^{1/2}} \exp\left[-2\frac{(\log(x) - a_2)^2}{a_4^2}\right] + a_1 \qquad [17]$$

where a_1, a_2, a_3, and a_4 are constants. The Louat function was shown to provide a better fit than the log-normal function to the 2-D simulation data obtained by using the old algorithm (Figure 11), in agreement with the results obtained by Anderson *et al.*[15] However, the log-normal function appears to give a better fit than the Louat function to the 2-D grain size distributions obtained by using the new algorithm (Figure 12). Figure 13 shows the grain size distribution obtained in 3-D simulations by using the new algorithm. The log-normal function appears to provide a much better fit to the 3-D simulation data than the Louat function, also in agreement with the results of Anderson *et al.*[15]

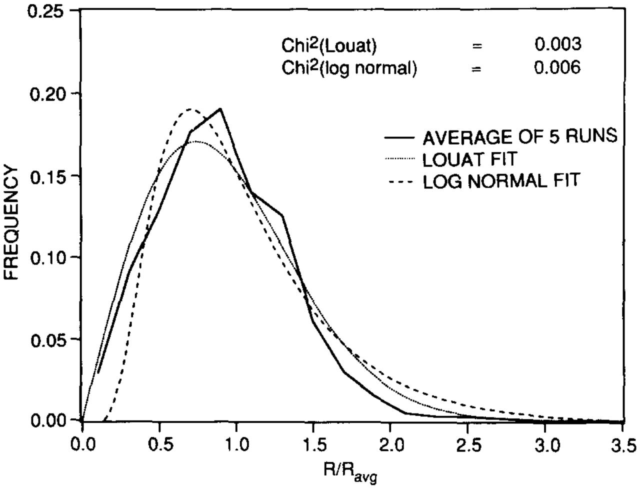


Fig. 11—Two-dimensional grain size distribution obtained with the old algorithm at a mean grain size of ≈10. The dotted lines are fits to the simulation data using the Louat function and the log-normal function. The Louat function gives a better fit to the simulation data than the log-normal function.

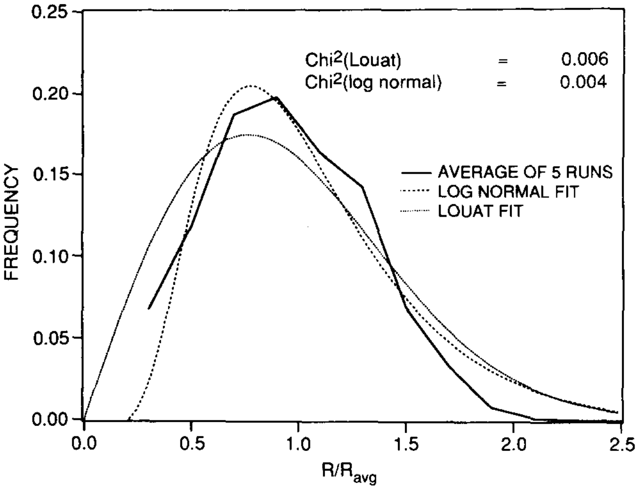


Fig. 12—Two-dimensional grain size distribution obtained with the new algorithm at a mean grain size of ≈10. The dotted lines are fits to the simulation data using the function proposed by Louat and the log-normal function. The log-normal function gives a better fit than the Louat function.

3. *Grain coalescence*

Grain coalescence during grain growth was monitored by plotting the ratio $R_{max}/\bar{R}$ as a function of the MCS. Any grain coalescence that gives rise to the formation of unusually large grains will be reflected by $R_{max}/\bar{R}$ values much greater than the expected range of 2 to 3. Figures 14, through 16 show the $R_{max}/\bar{R}$ ratio as a function of the MCS in 2-D simulations, obtained by using the old, modified-old, and new algorithms, respectively. Notice that the occurrence of grain coalescence resulted in the formation of unusually large grains with both the old and modified-old algorithms, especially in the early stages of grain growth, with the $R_{max}/\bar{R}$ value in the range 4 to 5. With continued grain growth, the grain coalescence becomes much less probable, although its occurrence at later stages of grain growth causes sudden jumps

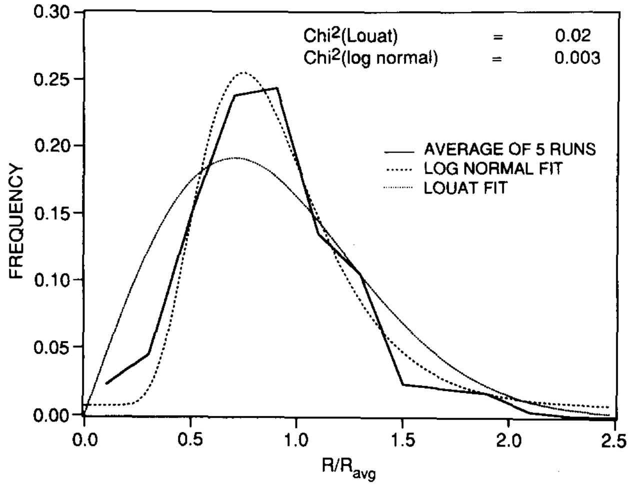


Fig. 13—Grain size distribution in 3-D obtained with the new algorithm. The dotted curves are fits to the simulation data using the Louat and the log-normal functions.

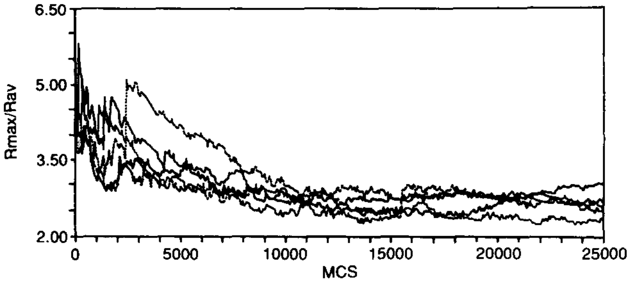


Fig. 14—The variation of $R_{max}/\bar{R}$ with grain size in 2-D simulations with the old algorithm. Grain coalescence in the initial stages of grain growth is indicated by $R_{max}/\bar{R}$ ratios in the 4 to 5 range and, in the later stages, by sudden jumps in the ratio.

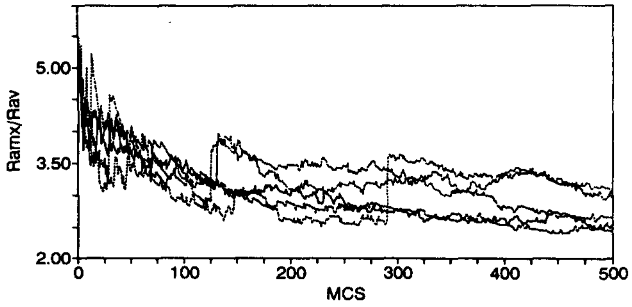


Fig. 15—The variation of $R_{max}/\bar{R}$ ratio with grain size in 2-D simulations with the modified-old algorithm. Grain coalescence is indicated by the same criterion as that described in Fig. 14.

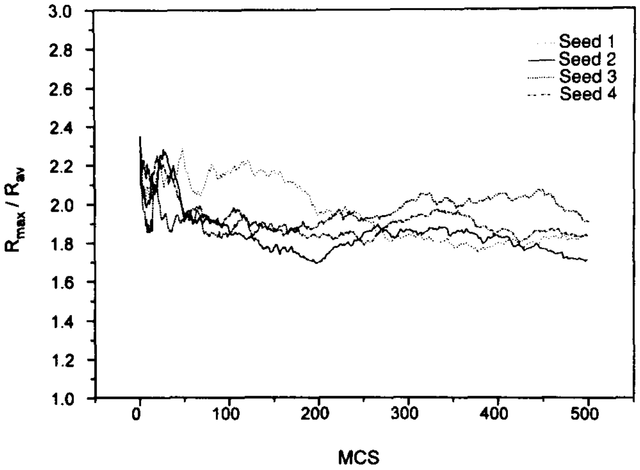


Fig. 16—The variation of $R_{max}/\bar{R}$ ratio with grain size in 2-D simulations with the new algorithm. Notice that values in the 4 to 5 range seen in Figs. 14 and 15 are absent. The values range from 2.4 to 1.8 throughout the grain growth range.

in the ratio, as can be seen in Figures 14 and 15. On the contrary, the $R_{max}/\bar{R}$ ratio for the new algorithm remains in the 1.8 to 2.4 range throughout the grain growth range, indicating absence of grain coalescence. The fluctuations in the initial stages of grain growth in Figure 16 result essentially from the fact that, initially, all grains have the same size (1.0). Figures 17 and 18 show the $R_{max}/\bar{R}$ ratio as a function of the MCS in 3-D simulations obtained by using the old and new algorithms, respectively. Notice the frequent occurrence of grain coalescence in Figure 17 and its absence in Figure 18. By comparing Figures 14 and 17, it can be seen that the limiting value of $R_{max}/\bar{R}$ obtained by using the old algorithm is higher in 3-D compared with 2-D simulations, suggesting that the frequency of grain coalescence in 3-D simulations is higher than that in 2-D ones for the old algorithm. Because grain coalescence is completely eliminated in the new algorithm, the limiting value of $R_{max}/\bar{R}$ is the same for both 2-D and 3-D simulations.

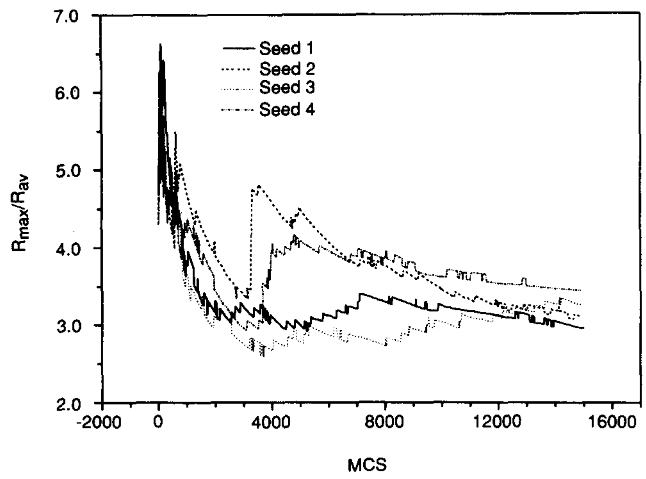


Fig. 17—The variation of $R_{max}/\bar{R}$ ratio with grain size in 3-D simulations with the old algorithm. Grain coalescence is again indicated as in Figs. 14 and 15. The limiting value of $R_{max}/\bar{R}$ is higher than in 2-D simulations using the old algorithm shown in Fig. 14.

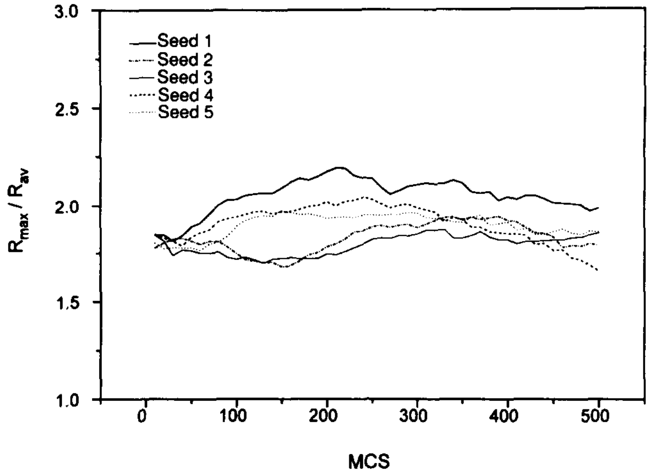


Fig. 18—The variation of $R_{max}/\bar{R}$ with grain size in 3-D simulations with the new algorithm. Grain coalescence is absent, and the limiting value of $R_{max}/\bar{R}$ is the same as in 2-D simulations using the new algorithm shown in Fig. 16.

C. *Anisotropic Grain Boundary Energy*

1. *Grain growth exponent*

Figure 19 shows the grain growth kinetics obtained in 2-D simulations with the new algorithm for various degrees of anisotropy. Notice that the presence of anisotropy results in a reduction of the grain growth exponent at a given grain size, as can be seen in Figure 19 and Table III.

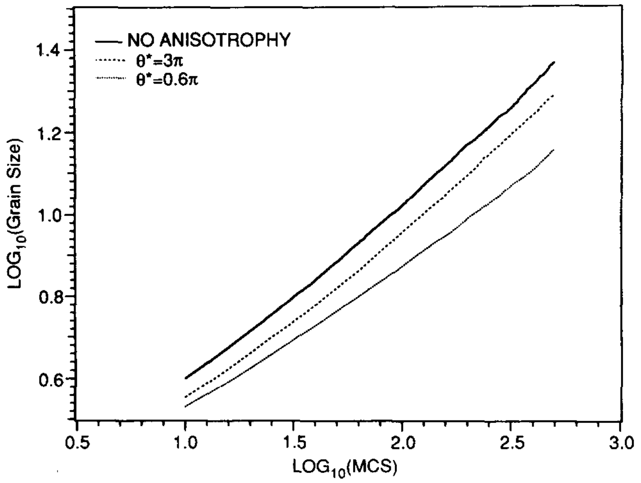


Fig. 19—The effect of anisotropy in grain boundary energy on grain growth kinetics in 2-D simulations, with the new algorithm. The grain growth exponent decreases with increasing degree of anisotropy.

Table III. **Effect of Anisotropy in Grain Boundary Energy on Grain Growth Exponent in 2-D Simulations Using the New Algorithm**

Anisotropy Parameter (θ^*)	Grain Size	Grain Growth Exponent
0	14.5	0.487
	10.0	0.47
0.3π	14.5	0.487
	10.0	0.468
0.6π	14.5	0.410
	10.0	0.395

2. *Grain size distributions*

Figure 20 shows the variation of $R_{\max}/\bar{R}$ with the MCS for 2-D simulations obtained using the new algorithm for different seed values and an anisotropy parameter of 0.6π. The $R_{\max}/\bar{R}$ increases from an initial value of ≈ 2.2 and appears to stabilize at ≈ 3.0 with grain growth. Figure 21 shows the grain size distribution in 2-D simulations obtained with the new algorithm for an anisotropy parameter (θ^*) of 0.6π. Notice that the fit to the log-normal function is significantly improved when the anisotropy in grain boundary energy is taken into account. The fit of the Louat function is shown for comparison, although there is no theoretical significance to the Louat function in the presence of anisotropic grain boundary energy.

VI. DISCUSSION

A. *Isotropic Grain Boundary Energy*

1. *Grain growth exponent*

A major difference between the old-algorithm, on one hand, and the modified-old and new algorithms, on the other, is in the magnitude of n. The old algorithm yielded a significantly lower grain growth exponent than the modified-old and new algorithms at any given grain size. Both the modified-old and new algorithms gave a grain growth exponent close to the theoretical value of 0.5 at a grain size ≈ 16, whereas at the same grain size, the old algorithm yielded an exponent ≈ 0.44. Because a significantly higher grain growth exponent was obtained in 2-D simulations by using the new and modified-old algorithms *vs* the old algorithm at a constant number of triple points or vertices (constant grain size), the absorption of curvature at the vertices does not seem to be the only reason that the grain growth exponent obtained in the old algorithm is lower. A comparison of the results based on the old and new algorithms only might lead to the conclusion that the difference in the grain growth exponents between these two algorithms could be caused by the occurrence of grain coalescence in the former and its absence in the latter. However, a comparison of the results based on the modified-old and new algorithms indicates that the modified-old algorithm, in spite of producing grain coalescence, yielded an almost

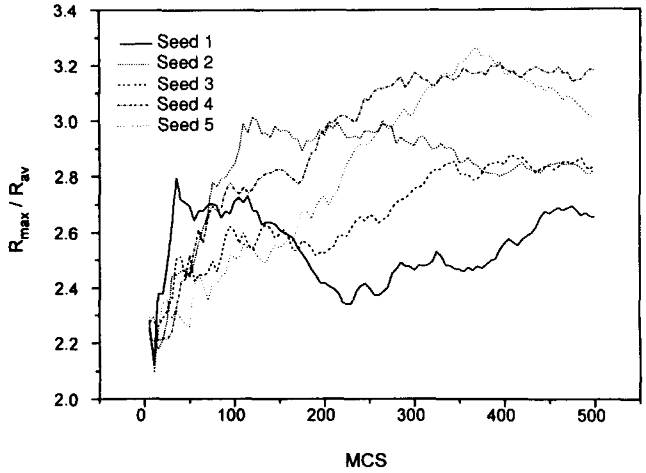


Fig. 20—The variation of $R_{\max}/\bar{R}$ with the MCS for 2-D simulations with the new algorithm, with an anisotropy parameter of 0.6π. The ratio increases from an initial value of ≈ 2.2 with the progress of grain growth and appears to stabilize at about 3.0.

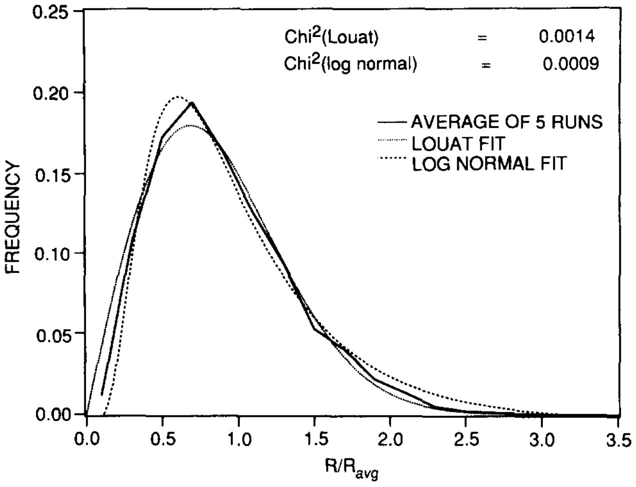


Fig. 21—Fits of log-normal and Louat functions to the grain size distribution obtained in 2-D simulations obtained by using the new algorithm, with an anisotropy parameter of 0.6π. Notice that the fits are much better than those in Fig. 12, in which grain boundary energy was considered isotropic.

identical grain growth exponent as the new algorithm, in which grain coalescence was completely absent. Hence, it is clear that grain coalescence does not have any significant effect on the magnitude of the grain growth exponent.

However, there is a significant difference between the old algorithm, on one hand, and the modified-old and new algorithms, on the other, with regard to the total number of orientations that are considered when a given MC site makes a reorientation attempt. As already explained, in the old algorithm, the reorientation attempt is toward all the possible orientations in the system, whereas in the modified-old and new algorithms, it is limited to those of the nearest neighbors only. The observed difference in the grain growth exponents can be fully attributed to this. As already explained, this difference between the two sets of algorithms results in a much faster decay of the average mobility as a function of grain size for the new and modified-old algorithms than for the old. Since a constant mobility is reached much faster, the asymptotic limit to the grain growth exponent is also reached much faster in the new and modified-old algorithms than in the old.

Figure 22 shows the variation of the average grain boundary mobility for the old and new algorithms. The simulations were carried out using a 200 × 200 lattice, with 64 initial orientations for the old algorithm, and 200^2 initial orientations for the new one. For the new algorithm, the asymptotic limit to the mobility (in arbitrary units) occurs at 0.33, while for the old algorithm, it occurs at 1.0. Notice that the mobility decays much faster in the new algorithm than in the old. An estimate of the grain growth exponent as a function of grain size for the old and new algorithms can be obtained as follows. At a given time, the total number of grains $G(t) \approx A/D^2$, where A is the area of the simulation domain and D is the average grain diameter. The total number of triple points or vertices $V(t)$ is twice the number of grains,[5] hence, $\approx 2A/D^2$. From stereology, the total length of grain boundaries in the simulation domain equals $(\pi/2D)A$. For a triangular lattice, the distance between two grain boundary points is equal to 1/2, and hence, the total number of grain boundary points $\approx \pi A/D$. The ratio of triple points to grain boundary points is then given by $2/\pi D$. Since the ratio of triple points/grain boundary points varies as $1/D$, the plots shown in Figure 22 were fitted to equations of the form $M = p_1 + p_2/D$, where p_1 and p_2 are constants, M is the mobility, and D is the grain size. The constant p_2 is proportional to the rate of change of mobility at any grain size. Since the grain growth exponent increases at the same rate that the mobility decreases, it is assumed that the grain growth exponent can be expressed in the form $n = n_f - p_2/D$, where n_f is the theoretical limit to the grain growth exponent and n is the grain growth exponent at any grain size D. Hence, it can be shown that

$$\frac{n_{\text{old}} - 0.5}{n_{\text{new}} - 0.5} = \frac{p_2^{\text{old}}}{p_2^{\text{new}}} \qquad [18]$$

The old algorithm gave grain growth exponents of 0.43 and 0.44 at grain sizes of 10 and 15.8, respectively (Table I). From Eq. [18], grain growth exponents 0.478 and 0.481 are obtained for the new algorithm at the same grain sizes. These values are in good agreement with the values obtained for the new algorithm, as shown in Table I.

A comparison of Table I and II shows that there is a significant reduction in the grain growth exponent on going from 2-D to 3-D simulations with the old algorithm. In 2-D simulations, the fraction of sites with enhanced jump probability is limited to those sites surrounding the triple points. However, in 3-D simulations, the fraction of sites with enhanced jump probability occurs near both quadruple points and triple edges. Hence, at a given grain size, the fraction of the total number of sites with enhanced mobility is higher in 3-D simulations compared to 2-D ones. Consequently, the deviation from the limiting mobility is higher than in 2-D simulations, and hence, the grain growth exponent is lower than in 2-D simulations. However, the new algorithm gives a significantly higher growth exponent than the old one at corresponding grain sizes because of the much faster attainment of equilibrium mobility as in the 2-D case. From the values of p_1, p_2, and p_3 for the fitting function shown in Figure 10 and from Eq. 18, the grain size at which the grain growth exponent becomes very close to the theoretical limit can be calculated. For example, a grain growth exponent of 0.49 can be obtained at a grain size of 8924. As mentioned previously, the apparent reduction in the grain growth exponent in MC simulations in the isotropic case is due to the increased mobility at the quadruple points and triple lines in the case of 3-D simulations and at triple points in 2-D simulations. Assuming an atomic dimension of 0.3 nm, MC grain size of 8924 roughly corresponds to 3 μm. Hence, in real materials, where the grain sizes are of the order of several microns, the ratio of atoms near quadruple points and triple lines to the atoms near grain boundaries is extremely small, and a grain growth exponent close to the theoretical limit of 0.5 is generally observed, provided there are no impurity drag and texture effects.

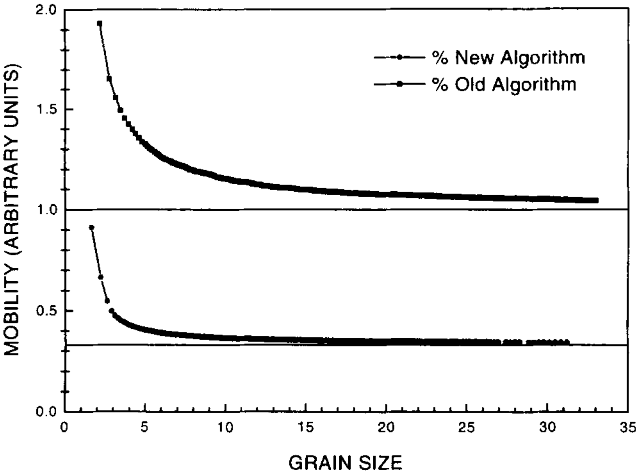


Fig. 22—Variation of average grain boundary mobility with grain size for the old and new algorithms. The horizontal lines at mobility equaling 1 and 0.33 indicate the asymptotic limits for the old and new algorithms, respectively.

2. *Grain size distribution*

As previously described, there was an excellent fit between simulated and experimentally measured grain size distributions for the old algorithm, in both 2-D and 3-D simulations. Hence, it was concluded that the old algorithm adequately described the grain size distribution function in real materials. However, one cannot overlook the fact that the shape of the grain size distribution curves obtained by using the old algorithm could be significantly influenced by the occurrence of grain coalescence. Comparison of Figures 14 through 18 shows that the effect of grain coalescence on the upper limiting grain size could be significant for both 2-D and 3-D simulations with the old algorithm. In fact, Figures 14 and 17 show that grain sizes in the range $R/\bar{R}$ = 3 to 3.5 are produced exclusively by grain coalescence.

Comparison of Figures 11 and 12 shows that a better fit to the Louat function is obtained in the 2-D simulations with the old algorithm than with the new algorithm. This is essentially due to the fact that the old algorithm produces a longer tail in the grain size distribution, which extends to $R_{\max}/\bar{R}$ ratios greater than 3.0. A log-normal fit to the 3-D grain size distribution produced by using the old algorithm has not been obtained in this study. However, the extent of grain coalescence in 3-D simulations obtained by using the old algorithm shown in Figure 17 demonstrates that a similar tail in the 3-D grain size distribution would also have been obtained by using the old algorithm, thereby improving the fit to the log-normal distribution. Hence, it can be concluded that the excellent fits of the Louat and log-normal functions to the 2-D and 3-D grain size distributions, respectively, obtained by using the old algorithm were essentially the result of the occurrence of grain coalescence in the simulations. The fact that real materials show such a grain size distribution could result from the presence of anisotropy in grain boundary energy, which also increases the upper limit of $R_{\max}/\bar{R}$ ratio in the grain size distributions. Grain coalescence could also occur, probably in the presence of subgrains or texture in the initial microstructure. However, the effect of the presence of subgrains or texture on grain growth kinetics and grain size distribution should be dealt with in a more fundamental manner than in the old algorithm. It is important to note that these parameters were not intended variables in the old algorithm.

B. *Anisotropic Grain Boundary Energy*

1. *Grain growth exponent*

From Figure 19 and Table III, it can be seen that the presence of anisotropy in grain boundary energy results in a reduction in the grain growth exponent. This reduction is apparently related to changes in the grain size distribution. With increasing anisotropy, there seems to be a larger number of both fine and coarse grains in the microstructure, with a corresponding decrease in the number of grains with mean grain size. This result is similar to that obtained by Grest *et al.*[8] with the old algorithm. However, the effect of anisotropy on grain growth exponent appears to be more pronounced. For example, with the old algorithm, the grain growth exponent appeared to be insensitive to the anisotropy parameter up to $\theta^* = 0.6\pi$. However, in the simulations presented here, the grain growth exponent drops significantly (from 0.49 to 0.41 at a grain size ≈15) at $\theta^* = 0.6\pi$. A comparison of the grain size distribution curves obtained with the old[8] and new algorithms at $\theta^* = 0.6\pi$ shows that the reduction in the peak size fraction is much more significant in the old algorithm than in the new algorithm. Also, with the old algorithm, the increase in the $R_{\max}/\bar{R}$ ratio compared with the isotropic case becomes significant only at $\theta^* = 0.9\pi$. However, with the new algorithm, the ratio increases from 2.2 in the isotropic case to about 3.0 for $\theta^* = 0.6\pi$. As described previously, the use of the old algorithm can result in grain nucleation during grain growth in the presence of anisotropy in grain boundary energy. However, the effect of grain nucleation on grain size distribution and the grain growth exponent is not clear at this point.

Recently,[23] curvature-driven grain growth in the presence of anisotropic grain boundary energy has been studied using the linear bubble model (LBM). The results indicate that in the presence of anisotropic grain boundary energy, the asymptotic limit to the grain growth exponent does not change from the theoretical limit of 0.5 obtained for the case of isotropic grain boundary energy. The log-log plot of grain size *vs* time in the presence of anisotropic grain boundary energy shown in Figure 19 does show a curvature. Fitting Eq. [14] to the $\theta^* = 0.6\pi$ curve, one obtains a grain size of 26,103,418, at which the grain growth exponent becomes 0.49. Hence, it is clear that it is impossible to achieve a grain growth exponent of 0.5 in MC simulations for the preceding level of anisotropy in grain boundary energy, since the lattice size required for such a simulation would become prohibitively large.

In addition to the decrease in the grain growth exponent, there seems to be a decrease in mobility. From Eq. [10] it can be seen that the mobility is a function of the grain boundary energy. In MC simulations with isotropic grain boundary energy, the quantity J is considered to be a constant. In simulations where the grain boundary energy is not isotropic, it varies from 0 to J, depending on the misorientation between the grains that form the boundary. Hence, the average grain boundary energy decreases with the increasing degree of anisotropy. It can be seen readily from Eq. [10] that the average mobility decreases with the increasing degree of anisotropy.

2. *Grain size distribution*

Figure 21 shows that the grain size distribution obtained by using the new algorithm in 2-D simulations shows an excellent fit to the log-normal function after taking into account the presence of grain boundary anisotropy. The effect of anisotropy in grain boundary energy on the grain size distribution in 3-D simulations is rather difficult to simulate, because there are additional degrees of freedom in defining the axis-angle pairs for grain boundaries in 3-D cases. The fraction of grain boundaries that are special boundaries is a function of prior thermal history and composition. For example, the presence of small amounts of impurities can effectively eliminate the difference in the mobility between special and general boundaries. Because these features are not treated in the simulations from a fundamental viewpoint, the degree of anisotropy should at best be considered to be an adjustable parameter at this time. In any case, the consideration of anisotropy in grain boundary energy in 3-D simulations is also expected to improve the log-normal fit to the 3-D grain size distribution, which is very much similar to the 2-D case shown in Figure 21. Hence, it can be concluded that the log-normal function can provide an excellent fit to grain size distributions both in 2-D and 3-D simulations, provided the anisotropy in grain boundary energy is taken into account.

VII. SUMMARY AND CONCLUSIONS

The following modifications have been implemented to the existing MC grain growth algorithm. The first relates to limiting the reorientation attempts to the orientations of the nearest neighbors only. The second modification involves increasing the number of initial grain orientations to the total number of points in the domain, so that grain coalescence can be completely eliminated during grain growth. Implementation of these modifications has resulted in a significant acceleration of the simulated grain growth kinetics and routine estimates of the grain growth exponent that are close to the theoretical value of 0.5 at smaller grain sizes than is possible with the old algorithm. Comparison of the old and modified algorithms showed that grain coalescence did not have a significant effect on the grain growth exponent. Careful comparison of the simulation results obtained by using the old and new algorithms showed that the slower attainment of the theoretical value of 0.5 for the grain growth exponent with the old algorithm essentially resulted from the slower rate of decay of the average grain boundary mobility with grain size compared with the new algorithm. The absorption of curvature at triple points did not appear to be a significant factor in reducing the grain growth exponent. The analyses also showed that the excellent fits to the Louat function and the log-normal function to the grain size distributions obtained in 2-D and 3-D simulations, respectively, by using the old algorithm resulted from the presence of the coarse grains in the microstructure produced by grain coalescence. The log-normal function gave a better fit than the Louat function for 2-D simulations obtained using the new algorithm. The closeness of fit significantly improved after a certain degree of anisotropy was introduced for the grain boundary energy. The log-normal function also gave an excellent fit to the grain size distribution in 3-D simulations obtained using the new algorithm in the absence of grain boundary anisotropy. However, based on the 2-D results, it is expected that the closeness of fit can be further improved in 3-D simulations also, in the presence of anisotropy in grain boundary energy.

ACKNOWLEDGMENTS

The authors would like to acknowledge Drs. J.M. Vitek and G.M. Ludtka of ORNL for reviewing the manuscript. The research was sponsored by the Division of Materials Sciences, United States Department of Energy, under Contract No. DE-AC05-84OR21400, with Martin Marietta Energy Systems, Inc.

REFERENCES

1. L.D. Fosdick: *Methods in Computational Physics*, Academic Press, New York, NY, 1963, vol. 1.
2. A.B. Bortz, M.H. Calos, and J.L. Lebowitz: *J. Comput. Phys.*, 1975, vol. 17, pp. 10-18.
3. D.J. Srolovitz, M.P. Anderson, G.S. Grest, and P.S. Sahni: *Scripta Metall.*, 1983, vol. 17, pp. 241-46.
4. P.S. Sahni, D.J. Srolovitz, G.S. Grest, M.P. Anderson, and S.A. Safran: *Phys. Rev. B*, 1983, vol. 28, pp. 2705-16.
5. M.P. Anderson, D.J. Srolovitz, G.S. Grest, and P.S. Sahni: *Acta Metall.*, 1984, vol. 32, pp. 783-91.
6. D.J. Srolovitz, M.P. Anderson, P.S. Sahni, and G.S. Grest: *Acta Metall.*, 1984, vol. 32, pp. 793-802.
7. D.J. Srolovitz, M.P. Anderson, G.S. Grest, and P.S. Sahni: *Acta Metall.*, 1984, vol. 32, pp. 1429-38.
8. G.S. Grest, D.J. Srolovitz, and M.P. Anderson: *Acta Metall.*, 1985, vol. 33, pp. 509-20.
9. D.J. Srolovitz, G.S. Grest, and M.P. Anderson: *Acta Metall.*, 1985, vol. 33, pp. 2233-47.
10. M.P. Anderson, G.S. Grest, and D.J. Srolovitz: in *Computer Simulation of Microstructural Evolution*, D.J. Srolovitz, ed., TMS, Warrendale, PA, 1986, pp. 77-93.
11. M.P. Anderson, G.S. Grest, and D.J. Srolovitz: *Scripta Metall.*, 1985, vol. 19, pp. 225-30.
12. F. Righetti, T.M. Liebling, and A. Mocellin: *Acta Stereol.*, 1989, vol. 8, pp. 459-64.
13. M.P. Anderson, G.S. Grest, R.D. Doherty, K. Li, and D.J. Srolovitz: *Scripta Metall.*, 1989, vol. 23, 753-58.
14. G.S. Grest, M.P. Anderson, and D.J. Srolovitz: *Phys. Rev. B*, 1988, vol. 38, pp. 4752-60.
15. M.P. Anderson, G.S. Grest, and D.J. Srolovitz: *Phil. Mag. B*, 1989, vol. 59, pp. 293-329.
16. E.A. Holm, D.J. Srolovitz, and J.W. Cahn: *Acta Metall.*, 1993, vol. 41, pp. 1119-36.
17. Y. Shen, B. Radhakrishnan, and R.G. Thompson: *3rd Int. Conf. on Trends in Welding Research*, Gatlinburgh, TN, ASM, Materials Park, OH, 1992, pp. 259-63.
18. N.P. Louat: *Acta Metall.*, 1974, vol. 22, pp. 721-24.
19. M. Hillert: *Acta Metall.*, 1965, vol. 13, pp. 227-38.
20. B. Radhakrishnan and T. Zacharia: *Int. Conf. on Modeling and Control of Joining Processes*, Orlando, FL, Dec. 8–10, 1993.
21. J. von Neumann: *Metal Interfaces*, ASM, Cleveland, OH, 1952, p. 108.
22. W.W. Mullins: *J. Appl. Phys.*, 1956, vol. 27, pp. 900-04.
23. W.W. Mullins and J. Vinals: *Acta Metall. Mater.*, 1993, vol. 41, pp. 1359-67.