

**A bounding box algorithm for  
three-dimensional phase field  
simulations of microstructural evolution  
in polycrystalline materials**

*Liesbeth Vanherpe*

*Nele Moelans*

*Bart Blanpain*

*Stefan Vandewalle*

*Report TW 494, May 2007*

**Katholieke Universiteit Leuven  
Department of Computer Science**

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# A bounding box algorithm for three-dimensional phase field simulations of microstructural evolution in polycrystalline materials

*Liesbeth Vanherpe*

*Nele Moelans*

*Bart Blanpain*

*Stefan Vandewalle*

*Report TW 494, May 2007*

Department of Computer Science, K.U.Leuven

## **Abstract**

Phase field modelling has proven to be a versatile tool for simulating microstructural evolution phenomena, such as grain growth in polycrystalline materials. However, the computing time and computing memory requirements of a phase field model pose severe limitations on the number of phase field variables that can be taken into account in a practical implementation. In this paper, a sparse bounding box method is proposed that allows the use of a large number of phase field variables without excessive memory usage or computational requirements. The algorithm is applied to a three-dimensional model for grain growth in the presence of second-phase particles.

**Keywords :** Grain growth, Zener pinning, Phase field models, Efficient simulation.

**AMS(MOS) Classification :** Primary : 35Q99, Secondary : 35K57, 65Y99.

# A bounding box algorithm for three-dimensional phase field simulations of microstructural evolution in polycrystalline materials

Liesbeth Vanherpe,<sup>1,\*</sup> Nele Moelans,<sup>2,†</sup> Bart Blanpain,<sup>2</sup> and Stefan Vandewalle<sup>1</sup>

<sup>1</sup>*Department of Computer Science, Faculty of Engineering, K.U.Leuven, 3001 Leuven, Belgium*

<sup>2</sup>*Department of Metallurgy and Materials Engineering,  
Faculty of Engineering, K.U.Leuven, 3001 Leuven, Belgium*

(Dated: 12th June 2007)

Phase field modelling has proven to be a versatile tool for simulating microstructural evolution phenomena, such as grain growth in polycrystalline materials. However, the computing time and computing memory requirements of a phase field model pose severe limitations on the number of phase field variables that can be taken into account in a practical implementation. In this paper, a sparse bounding box method is proposed that allows the use of a large number of phase field variables without excessive memory usage or computational requirements. The algorithm is applied to a three-dimensional model for grain growth in the presence of second-phase particles.

PACS numbers: 81.10.Aj, 64.10.+h, 02.60.Cb, 02.70.-c

## I. INTRODUCTION

The microstructure of materials often consists of multiple grains with different crystallographic orientations. The study of the evolution of these grains is of great technological importance, because many material properties, such as strength and toughness, depend on the mean grain size and grain size distribution. A large number of theoretical, experimental and computational studies have been performed on this subject (see for example [1–5]). Although it is well understood that grain growth behavior results from the interplay between curvature driven grain boundary movement and the geometrical requirements at triple junctions, there are still many controversies, especially with respect to the shape and evolution of the grain size distribution. Computer simulations based on mesoscale models such as Monte Carlo, vertex and phase field models and cellular automata, are essential for a better understanding of grain growth. They allow to study separately the roles of different parameters, which is impossible in experimental studies on real materials. Moreover, images of three-dimensional simulations provide more insight in the shape and size of grains than two-dimensional microscopic images of cross-sections of a material.

Phase field modelling has shown to be a versatile tool for simulating microstructural evolution phenomena, such as solidification [6–8], precipitation [9, 10] and grain growth [11–13]. It allows to predict the evolution of complex morphologies considering different thermodynamic driving forces, such as interfacial energy, bulk energy and elastic energy, and different transport processes, such as heat and mass diffusion. Grain growth in single-phase polycrystalline materials has been simulated with the phase field method by different authors [14–16]. The model of [14], was extensively discussed and results were compared with experimental observations and analytical theories in [12, 17]. An extension of the model towards two-phase materials was proposed in [18–20], and applied for simulating Zener pinning by growing second-phase particles in [21], and for particles which are constant in time in [22, 23]. In [14], the phase field model represents the microstructure of a single-phase material by a set of  $p$  non-conserved phase field variables  $\eta_1(\mathbf{r}, t)$ ,  $\eta_2(\mathbf{r}, t)$ ,  $\dots$ ,  $\eta_p(\mathbf{r}, t)$ . These phase field variables are used to distinguish the different crystallographic orientations of the grains. Inside a grain, one phase field variable  $\eta_i$  takes a nonzero constant value, mostly 1 or  $-1$ , while the other phase field variables assume values close to zero. Across the grain boundaries, the corresponding field variables vary continuously from their equilibrium value in the grain to their equilibrium value in the neighboring grains. The spatial and temporal evolution of the phase field variables is governed by the time-dependent Ginzburg-Landau equations. In principle, the number of phase field variables  $p$  should equal or exceed the total number of grains, as in reality the number of possible grain orientations is infinite. When not enough crystallographic orientations are involved in a grain growth simulation, growth can occur by coalescence of two neighboring grains with the same orientation. This leads to incorrect growth kinetics and unphysical grain shapes. Especially in three-dimensional simulations, where grains have on average more neighbors than in two dimensions, a very large number of crystallographic orientations are required to minimize the effect of grain coalescence [13]. Moreover, for anisotropic materials, it is important that the

---

\*Electronic address: [liesbeth.vanherpe@cs.kuleuven.be](mailto:liesbeth.vanherpe@cs.kuleuven.be)

†Electronic address: [nele.moelans@mtm.kuleuven.be](mailto:nele.moelans@mtm.kuleuven.be)

orientation dependence of material properties is resolved well. Also, when the pinning effect of particles is modelled, the spatial resolution of the employed numerical technique has to be fine enough in order to represent the particles, which are much smaller than the grains, and reproduce the shape of grain boundaries at particle-grain boundary intersections correctly. To conclude, since one is mostly interested in the evolution of the grain size distribution, a large amount of grains (and particles) must be considered in grain growth simulations to achieve reliable statistics. As a consequence, realistic three-dimensional computer simulations for grain growth with a phase field model demand very significant amounts of computation power.

Several algorithms have been designed to overcome the computational limitations of the phase field method. In [13, 24], the grain orientations are dynamically reassigned to reduce memory requirements and to avoid frequent grain coalescence. This approach is limited to systems where the only use of the phase field variables is to distinguish unique domains. Incorporating anisotropy or any property depending on the relative or absolute position of a grain or the orientation difference between neighboring grains into this technique is impossible. Adaptive meshing [25–27], and moving mesh techniques [28, 29], have been used in phase field simulations to increase computational efficiency. However, so far these techniques have proven effective only for simulations with a small number of phase field variables. For polycrystalline structures its benefits drastically reduce, because of the amount of interface involved. In [30–32], efficient algorithms using a sparse data structure are proposed. These algorithms are based on the observation that only a few crystallographic orientations are active at a given point in a microstructure. These sparse methods show significant improvements over conventional algorithms as they scale with the size of the microstructure instead of with the number of crystallographic orientations involved. In this paper, these techniques are further explored. An object-oriented bounding box method, based on the same observations that led to [30–32], is presented. The algorithm is designed in the context of the study of grain growth in the presence of second-phase particles. However, it can be applied to other cases as well. Thanks to the object-oriented design, the method can be easily extended to more complicated phase field models, for example for grain growth in materials which contain impurities, either dissolved or undissolved, and for models that show orientation dependence. Moreover, the object-oriented approach has definite advantages in post-processing.

The structure of the paper is as follows. The phase field model for which we developed the bounding box algorithm, is presented in section II. In section III, the bounding box algorithm itself is explained. Then, the implementation and simulation conditions are specified in section IV. The bounding box algorithm is validated for three-dimensional simulations in sections V and VI. In section VII, the method is applied to grain growth in the presence of second-phase particles and simulation results are discussed. Some concluding remarks are formulated in section VIII.

## II. PHASE FIELD MODEL AND DISCRETIZATION

The bounding box algorithm has been implemented for phase field models for grain growth, as described in [14]. According to [14], the microstructure of a single-phase material is represented by a set of  $p$  phase field variables

$$\eta_1(\mathbf{r}, t), \eta_2(\mathbf{r}, t), \dots, \eta_p(\mathbf{r}, t). \quad (1)$$

The phase field variables are used to distinguish the different crystallographic orientations of the grains and are continuous functions of the spatial coordinates and time. The spatial and temporal evolution of the phase field variables is governed by the time-dependent Ginzburg-Landau equations,

$$\frac{\partial \eta_i(\mathbf{r}, t)}{\partial t} = -L_i \frac{\partial F}{\partial \eta_i(\mathbf{r}, t)}, \quad i = 1, \dots, p. \quad (2)$$

The kinetic coefficients  $L_i$  are related to the grain boundary mobility. The free energy  $F$  of the system is described by

$$F = \int_V \left( \sum_{i=1}^p \left( \frac{\eta_i^4}{4} - \frac{\eta_i^2}{2} \right) + \sum_{i=1}^p \sum_{j \neq i}^p \eta_i^2 \eta_j^2 + \sum_{i=1}^p \frac{\kappa_i}{2} (\nabla \eta_i)^2 \right) d^3 \mathbf{r}, \quad (3)$$

with  $\kappa_i$  the gradient energy coefficients.

In [22, 23], this phase field model is expanded for simulating grain growth in materials containing small incoherent second-phase particles with constant properties. To include such particles in the model, a spatially dependent parameter  $\phi$  is added. This parameter  $\phi$  equals 1 inside a particle and 0 elsewhere and remains constant in time. The free energy  $F$  of the system is now described by

$$F = \int_V \left( \sum_{i=1}^p \left( \frac{\eta_i^4}{4} - \frac{\eta_i^2}{2} \right) + \sum_{i=1}^p \sum_{j \neq i}^p \eta_i^2 \eta_j^2 + \phi^2 \sum_{i=1}^p \eta_i^2 + \sum_{i=1}^p \frac{\kappa_i}{2} (\nabla \eta_i)^2 \right) d^3 \mathbf{r}. \quad (4)$$

Substituting (4) to (2) results in a set of reaction-diffusion partial differential equations

$$\frac{\partial \eta_i}{\partial t} = L_i \left( \kappa_i \nabla^2 \eta_i - \eta_i^3 + \eta_i - 2\eta_i \left( \sum_{j \neq i}^p \eta_j^2 + \phi^2 \right) \right), \quad i = 1, \dots, p. \quad (5)$$

For isotropic grain boundary energy and mobility the coefficients are fixed to  $\kappa_i = \kappa$  and  $L_i = L$  for all  $i$ . Periodic boundary conditions are applied, as in [33].

The equations (5) are solved by using a finite difference scheme. The spatial derivative is discretized with second order central differences. For  $\mathbf{r} = (x, y, z)$ , we have

$$\nabla^2 \eta_i(\mathbf{r}) = \sum_{v=x,y,z} \frac{\eta_i(v + \Delta v) - 2\eta_i(v) + \eta_i(v - \Delta v)}{(\Delta v)^2}, \quad (6)$$

where  $\Delta x$ ,  $\Delta y$  and  $\Delta z$  denote the mesh widths in the finite difference grid. The time derivative is discretized using a first order semi-implicit scheme [34]. The diffusion part is treated implicitly, the reaction part explicitly:

$$\frac{\eta_i^{n+1} - \eta_i^n}{\Delta t} = L \left[ \left( \kappa \nabla^2 \eta_i \right)^{n+1} + \left( \eta_i^3 + \eta_i - 2\eta_i \left( \sum_j^p \eta_j^2 + \phi^2 \right) \right)^n \right], \quad i = 1, \dots, p. \quad (7)$$

where the superscript  $n$  indicates the solutions at time step  $n$ . The implicit treatment of the Laplacian, combined with the explicit treatment of the nonlinear coupling of the phase field variables, allows the use of a large time step without the need to solve one very large coupled system of equations. It effectively decouples the system into  $p$  scalar diffusion equations, to be solved at every time step.

### III. BOUNDING BOX ALGORITHM

A phase field simulation is both memory consuming and computationally intensive. There are several ways to speed up the phase field computations. Because of the explicit treatment of the coupling of the phase field variables, the equations can be assigned to different processors and solutions can be computed in parallel [24]. A closer look at a grain growth simulation reveals that the solutions of the model display small regions of high activity surrounded by large regions of inactivity. This property, combined with the semi-implicit discretization, is exploited in this paper by only solving the equations locally. This technique lowers both storage requirements and computing time and computing memory.

To initialize the bounding box algorithm, a polycrystalline microstructure is required. The initial microstructure can be obtained for example from microscopic images, Voronoi calculation or by simulation.

A phase field variable is defined to be active at a given grid point when its absolute value exceeds a small positive threshold value  $\epsilon$ . At every grid point, only a few phase field variables are active and thus contribute to the evolution of the microstructure; inside each grain  $i$ , away from its boundaries, one phase field variable,  $\eta_i$ , is active and near grain boundaries, only those phase field variables corresponding to the neighboring grains, are active. Based on these properties, a grain  $G_i$  is now identified by its type  $i$  and a set of ordered couples  $(\mathbf{r}, \eta_i(\mathbf{r}))$  with grid points  $\mathbf{r}$  that are connected, and with  $|\eta_i(\mathbf{r})| > \epsilon$ :

$$G_i = \{(\mathbf{r}, \eta_i(\mathbf{r})) : |\eta_i(\mathbf{r})| > \epsilon \text{ and } \mathbf{r} \text{ connected}\}. \quad (8)$$

The type  $i$  corresponds to a crystallographic orientation. Grains are allowed to wrap around the grid boundaries so that periodic boundary conditions are automatically taken into account.

For every grain  $G_i$ , a bounding box is determined as the smallest cuboid grid part containing the grid points  $\mathbf{r}$  of  $G_i$ . The bounding box is completely characterized by the coordinates of two delimiting grid points. The set  $B_i$  is then defined as

$$B_i = \{(\mathbf{r}, \eta_i(\mathbf{r})) : \mathbf{r} \text{ lies inside the bounding box of } G_i\}. \quad (9)$$

Having defined the active phase field variables, grains  $G_i$  and sets  $B_i$ , the equations (5) only have to be solved for the values contained in the sets  $B_i$ . From the starting point of the bounding box algorithm on, the phase field values not exceeding  $\epsilon$  in absolute value are assumed to be zero. Therefore, a margin of one grid point, filled with zeros, can be added to the bounding boxes and sets  $B_i$  without violating this assumption. The equations (5) can thus be solved locally using homogeneous Dirichlet conditions and allow for the grains to grow inside the bounding boxes.

Listing 1: Bounding box algorithm

```

<preprocessing step>
for phase field variables  $i = 1, \dots, p$ 
  find grains  $\hat{G}_{i,k}$ 
  for  $k = 1, \dots, n_i$ 
    determine bounding box delimiters and  $\hat{B}_{i,k}$ 
  grain numbering procedure
<computation step>
for time steps  $t = 1, \dots, n$ 
  for all grid points
    compute  $\sum \eta_j^2 + \phi^2$ 
  for  $j = 1, \dots, q$ 
    solve equation (5) for  $B_j$ 
    update bounding box delimiters and  $B_j$ 

```

The resulting systems can be solved with iterative methods, for example with multigrid methods, the successive over relaxation (SOR) method or the Gauss-Seidel method. The latter is the method used in this paper.

When not enough phase field variables are involved in representing the microstructure, grain growth can occur by coalescence. To prevent this coalescence when the bounding box algorithm is employed, a special procedure which assigns a different crystallographic orientation to every grain is applied before solving the equations (5). The bounding box algorithm now proceeds as in Listing 1. The *preprocessing step* of the algorithm is illustrated in Fig. 1. First, for every phase field variable  $\eta_i$ , all corresponding grains  $\hat{G}_{i,k}$ ,  $k = 1, \dots, n_i$  are located, with  $n_i$  the number of grains associated with  $\eta_i$ . Second, the corresponding bounding boxes and the sets  $\hat{B}_{i,k}$  are established. Finally, a grain numbering takes place and a new set of phase field variables is introduced, such that there is a one to one mapping between grains and phase field variables. More precisely, every grain  $\hat{G}_{i,k}$  for  $i = 1, \dots, p$  and  $1, \dots, n_i$ , is given a unique index  $j$ , ranging from 1 to  $q$ , the total number of grains, with  $q = \sum_{i=1}^p n_i$ . That is, grain  $\hat{G}_{i,k}$  is rewritten as  $G_j$ , a grain corresponding to one of the new phase field variables,  $\eta_j$ , for  $j = 1, \dots, q$ , with bounding box information copied from  $\hat{B}_{i,k}$ . Note that this renumbering of grains and phase field variables is allowed, since the model considered in this paper is fully isotropic. The actual value of the grain orientation does not play a role in equation (5). As such, the different phase field variables merely distinguish different grain types. As long as they are sufficient in number, their actual numbering is not important.

In the bounding box data structure, the microstructure is treated as an object with a number of attributes: the grid size, the set of all grains  $G_j$  and, if present in the model, second-phase particles or other features of the polycrystalline microstructure. A grain in turn is also treated as an object. Its attributes are the material properties  $\kappa$  and  $L$ , the bounding box delimiters, the type and the set of values  $B_j$ . Figure 2 shows a UML diagram of the object-oriented data structure. The line connecting the elements of the data structure indicates their association relationship. The notation at each end of the line indicates the multiplicity, which is the number of objects that participate in the association.

After the data structure is determined, the evolution of the microstructure is computed in the *computation step*. Within each time step, two substeps are performed. First, for every grid point, the sum of the squares of all phase field variables active at that point is computed. When simulating grain growth in the presence of second-phase particles, the additional term  $\phi^2$  is included in this sum. Second, for every set  $B_j$  in turn, the solution of the equation (5) is computed using the discretization described in section II. After the computation, the algorithm checks whether the grain  $G_j$  has shrunk or grown and adjusts the bounding box delimiters and the set  $B_j$  accordingly, taking into account the margin of zeros.

As mentioned, a grain is treated as an object with a number of properties. Anisotropy and other properties depending on the absolute or relative positions of the grains can be easily inserted into the object-oriented implementation by adding attributes to a grain or the microstructure (see Fig. 3). Every grain can be attributed different properties. A property is stored once per grain and not for every grid point of the grain. Furthermore, in the same way as grains, the second-phase particles can be treated as objects with specific properties.

The data structure of the bounding box algorithm can save significant time in post-processing, e.g. when the number of grains or the mean grain size have to be determined. The number of grains and their location is known throughout the simulation and this information can be used immediately.

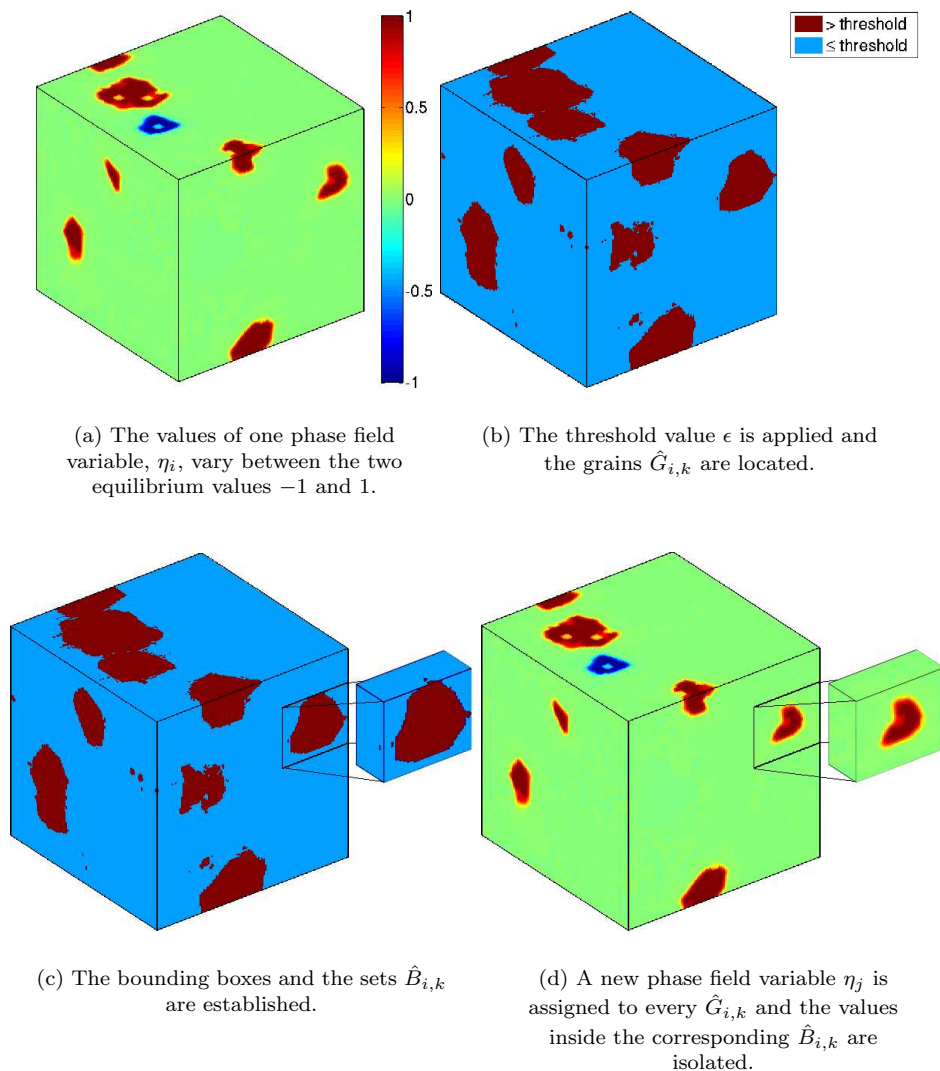


Figure 1: Locating the grains and determining the bounding boxes in the preprocessing step.

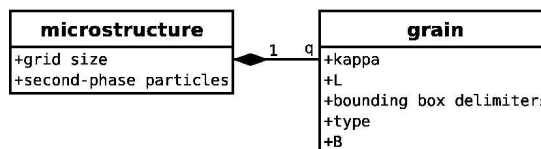


Figure 2: UML diagram of the object-oriented data structure composed in the preprocessing step. The microstructure and grains are treated as objects with a number of attributes.

#### IV. SIMULATIONS

To obtain an initial polycrystalline microstructure for simulations with the bounding box method, a parallel implementation was made of an algorithm solving the equations (5) globally, based on the semi-implicit finite-difference scheme proposed in [35]. This algorithm and its characteristics will be referred to as the conventional algorithm and the conventional characteristics. Simulations with the conventional algorithm were run on a  $256 \times 256 \times 256$  grid. Because of memory limitations, only 100 phase field variables were employed, whereas according to [13], more than 200 phase field variables are required to prevent grain coalescence in a three-dimensional simulation. The parameter values were set to  $\kappa = 0.5$  and  $L = 1$  and no second-phase particles were included. The discretization spacings were

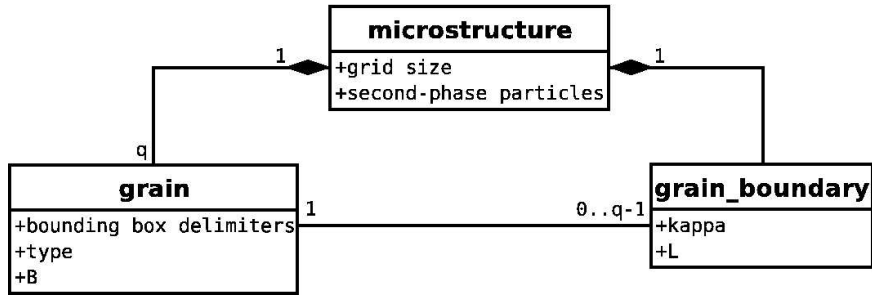


Figure 3: UML diagram of the object-oriented data structure, extended to simulate anisotropic grain growth. The microstructure, grains and grain boundaries are treated as objects with a number of attributes.

$\Delta x = 1$  and  $\Delta t = 0.2$ . At the start of a simulation, small random values between  $-0.001$  and  $0.001$  were assigned to the phase field variables at all grid points. All computations were performed on 20 nodes of a computer cluster, which are interconnected with an Infiniband network. At simulation time  $t_s$ , the conventional algorithm was stopped and the preprocessing of the bounding box algorithm was applied.

When not enough phase field variables are involved in a conventional simulation, the small grains which nucleate and grow during the simulation can undergo significant coalescing. In spite of the grain renumbering procedure in the preprocessing step (see section III), some grains might still coalesce in the first stages of the bounding box algorithm. In the time steps before two grains of the same type  $i$  coalesce, the values of the involved phase field variable start to increase along the contact boundary and eventually exceed  $\epsilon$  in magnitude. At that point, the two grains have become one according to the definition. By using higher values of  $\epsilon$ , the continued coalescence can be prevented. The use of too high threshold values will however disturb the accuracy of the simulation results. Nevertheless, we feel that the method described above is a convenient way to obtain an initial large polycrystalline structure since in the case of ideal grain growth, the typical grain structure with stationary grain size distribution is recovered after a short transition time (see also [36]).

To visualize the microstructural evolution, the function  $\psi$  is defined:

$$\psi(\mathbf{r}, t) = \sum_{j=1}^q \eta_j^2(\mathbf{r}, t) \quad (10)$$

Only the values inside the bounding boxes are taken into account for the computation of (10). The function  $\psi$  takes the value 1 inside grains and considerably smaller values on grain boundaries.

## V. VALIDATION OF THE BOUNDING BOX ALGORITHM

There are two important parameters in the bounding box algorithm: the applied threshold value  $\epsilon$  and the initial mean grain size. When the microstructure is obtained through simulation, the mean grain size is determined by the simulation time point  $t_s$  at which the bounding box algorithm is started (see section IV). The parameters  $\epsilon$  and the mean grain size influence the accuracy, computing memory and computing time. In this section, some implementation independent characteristics of the algorithm are discussed. Section VI treats the characteristics which are implementation dependent.

### A. Preprocessing step

In the preprocessing step, the active regions of the microstructure are identified and isolated. As a result, the storage requirements of the bounding box data structure are significantly lower than those of the conventional grid-based data structure. Whereas the requirements of the conventional data structure equal the number of phase field variables multiplied by the grid size, the requirements of the bounding box data structure are determined by the number of phase field values included in the sets  $B_j$ . This number depends on  $\epsilon$  and the initial mean grain size and, as will be shown below, is typically a small fraction of what would be necessary in a more straightforward implementation.

The preprocessing step was performed on the simulation results specified in section IV at time points  $t_s = 200, 400, 600, \text{ and } 800$ . Different values of  $\epsilon$  were applied:  $10^{-3}, 10^{-4}, 10^{-5}, \text{ and } 10^{-6}$ . For both the bounding box

Table I: Ratio of the number of phase field values included in the data structure to the grid size for the bounding box algorithm and the conventional algorithm as a function of  $\epsilon$  and the initial mean grain size (grid points).

starting $t_s$	mean grain size	threshold value $\epsilon$				conventional
		$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	
$t_s = 200$	$3.5 \times 10^3$	6.15	9.35	14.19	31.12	100
$t_s = 400$	$7.8 \times 10^3$	4.98	7.11	9.74	13.33	100
$t_s = 600$	$1.4 \times 10^4$	4.29	5.84	7.73	10.16	100
$t_s = 800$	$2.0 \times 10^4$	3.85	5.00	6.40	7.94	100

Table II: Ratio (%) of the number of phase field values included in the bounding box computations to the number included in the conventional computations as a function of  $\epsilon$  and the simulation time.

simulation time $t$	threshold value $\epsilon$			
	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
$t_s$	5.75	6.48	7.26	8.10
$t_s + 200$	6.71	6.74	6.90	6.90
$t_s + 400$	6.72	6.73	6.81	6.81
$t_s + 600$	6.68	6.69	6.73	6.73
$t_s + 800$	6.64	6.65	6.68	6.68

algorithm and the conventional algorithm, the number of phase field values was counted. Table I shows the ratio of the number of phase field values included in the data structure to the grid size for the bounding box algorithm and the conventional algorithm as a function of  $\epsilon$  and  $t_s$  and thus the initial mean grain size. It can be seen that the bounding box algorithm is far more efficient than the conventional algorithm. The fine grained topology of the microstructure at the initial evolution stages results in smaller initial mean grain sizes and higher ratios.

## B. Computation step

In the computation step of the bounding box algorithm, equations (5) are only solved for the sets  $B_j$ . A large amount of computing time and computing memory is thus saved. To validate the computation step of the bounding box algorithm, a  $256 \times 256 \times 256$  microstructure was generated containing 67 grains with a mean grain size of  $2.3 \times 10^5$  grid points and a different crystallographic orientation for every grain. During a simulation, the topology of a microstructure becomes more and more coarse grained. While the memory demand of the conventional algorithm is independent of the topology, the bounding box algorithm is more efficient for coarser grained topologies. Table II shows the time evolution of the ratio of the number of phase field values included in the bounding box computations to the number included in the conventional computations for different threshold values  $\epsilon$ . It can be seen that the memory efficiency increases with the simulation time for the lowest threshold values. Also, in the course of the simulation, the memory efficiencies for the different threshold values converge. This is because the different threshold values have no significant influence on the long term topology.

The effect of the use of different threshold values  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$  on the accuracy of the computations is almost independent of  $\epsilon$ . To study this effect, the results of a conventional simulation have been compared with those of a bounding box simulation after 1, 1000 and 7000 time steps, starting from the microstructure described above. After one time step, the point-wise differences between results for the threshold values  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$  are of order  $10^{-5}$ ,  $10^{-6}$ ,  $10^{-6}$ , and  $10^{-6}$  respectively. Furthermore, the mean grain size shows no influence of the use of different threshold values. Figure 4 shows a  $128 \times 128 \times 128$  part of the microstructures computed after 1000, 3000 and 7000 time steps. Three evolving grains are indicated. There are no visual differences between the microstructure computed by the conventional algorithm, and by the bounding box algorithm for  $\epsilon = 10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$  and  $10^{-6}$ . After 1000 time steps, the point-wise differences of the results are only of order  $10^{-3}$ . The mean grain size obtained by the bounding box algorithm deviates less than  $10^{-5}\%$  from the size obtained by the conventional algorithm. The differences between the results obtained for the different values of  $\epsilon$  are even smaller. After 7000 time steps, the point-wise differences of all computed results are only of order  $10^{-2}$  and located at the boundaries of grains which are slightly larger or smaller when compared. The mean grain sizes computed by the bounding box algorithm

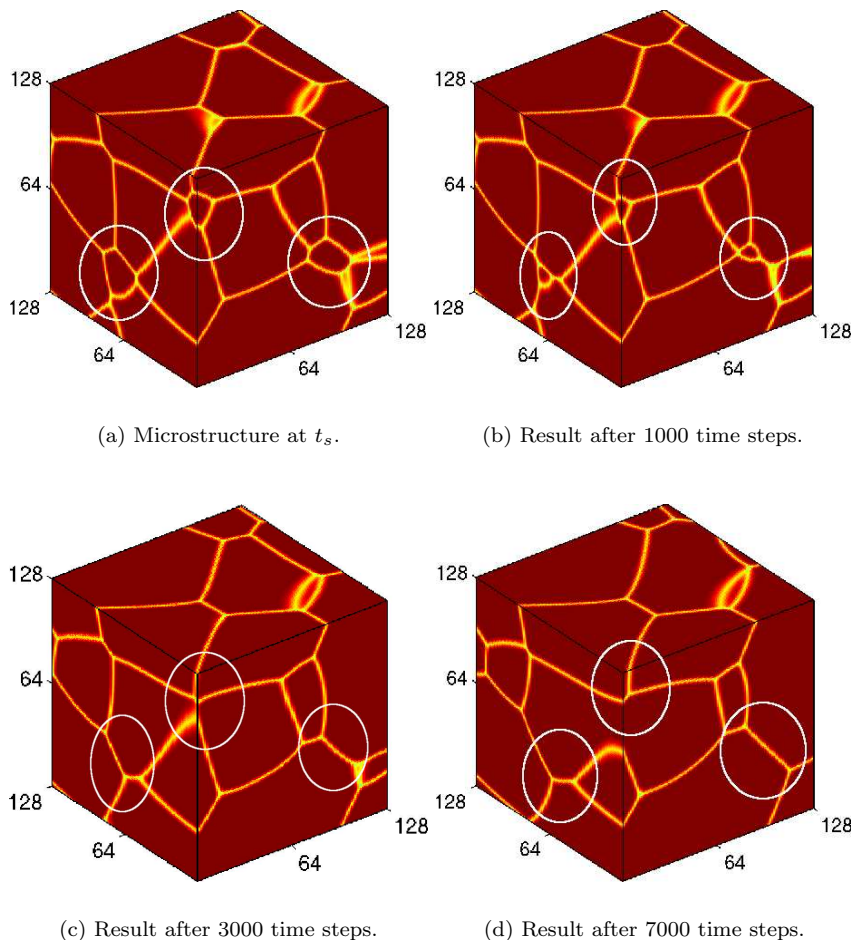


Figure 4:  $128 \times 128 \times 128$  part of the microstructure computed after 1000, 3000 and 7000 time steps. The results for the conventional algorithm and for the bounding box algorithm with  $\epsilon = 10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$  and  $10^{-6}$  are visually indistinguishable.

for the different threshold values differ by less than  $10^{-5}\%$ . The differences between the mean grain sizes computed by the bounding box algorithm and the conventional algorithm are larger but still only 2%.

As described in section IV, when the initial microstructure is obtained by simulation, the value of  $\epsilon$  can influence the growth kinetics: low threshold values can allow for continued coalescence, whereas high threshold values prevent this. To study this influence, the bounding box is applied to the simulation results specified in section IV. Figure 5 shows the evolution of the mean grain size from  $t_s = 200$  to  $t = 3000$ , computed by the conventional algorithm and the bounding box algorithm for several  $\epsilon$  values. It can be seen that the mean grain size is higher for lower threshold values and highest for the conventional algorithm.

The time point  $t_s$  also has an influence on the growth kinetics: when  $t_s$  is chosen earlier, some amount of coalescence is prevented. To study this effect, the bounding box algorithm was started at  $t_s = 200, 400, 600,$  and  $800$  and run until  $t = 3000$ . Figure 6 illustrates the computed evolution of the mean grain size from the different  $t_s$  to  $t = 3000$  for  $\epsilon = 10^{-4}$  and  $\epsilon = 10^{-5}$  and the conventional algorithm. As anticipated, the mean grain size is smaller when  $t_s$  is chosen earlier. For smaller values of  $\epsilon$ , this effect is not clear: the choice of a smaller threshold value also leads to coalescence.

Based on the observations concerning the memory reduction, accuracy and growth kinetics for the different threshold values, we advocate to use  $\epsilon = 10^{-5}$  or  $10^{-6}$ .

## VI. COMPUTATIONAL REQUIREMENTS

In this section, implementation dependent characteristics of the bounding box algorithm are discussed.

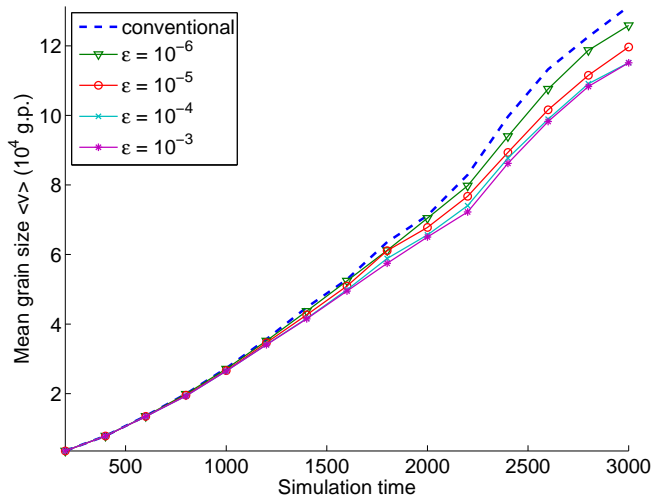


Figure 5: Evolution of mean grain size from  $t_s = 200$  to  $t = 3000$ , computed by the conventional algorithm and by the bounding box algorithm for different values of  $\epsilon$ .

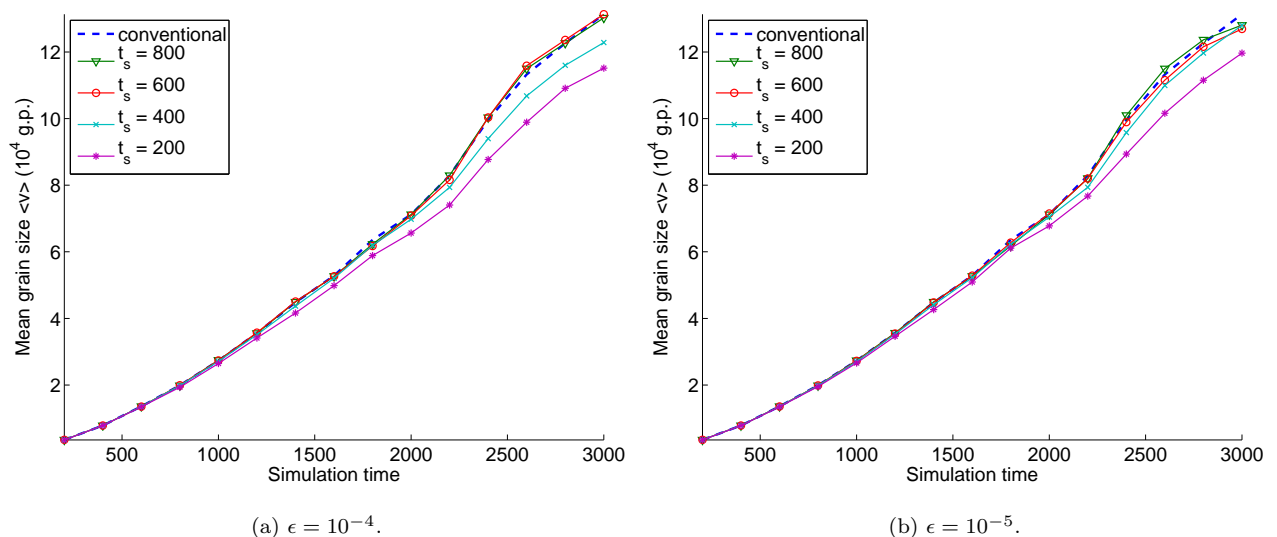


Figure 6: Evolution of mean grain size from  $t_s$  to  $t = 3000$ , computed by the conventional algorithm and by the bounding box algorithm for different starting time points  $t_s$  and different values of  $\epsilon$ .

### A. Preprocessing step

From the preprocessing step of the bounding box algorithm on, all phase field values not exceeding  $\epsilon$  in absolute value are assumed to be zero. They are not included in the sets  $B_j$  and therefore excluded from further computations. As a consequence, these values do not have to be stored and the data resulting from a bounding box simulation require less storage than the data from a conventional simulation. Table III displays the bounding box storage requirements for  $t_s = 200, 400, 600,$  and  $800$ , and thus different initial mean grain sizes, and  $\epsilon = 10^{-3}, 10^{-4}, 10^{-5},$  and  $10^{-6}$  and the conventional storage requirements. For smaller mean grain size, the storage requirements are higher. Furthermore, the threshold value  $\epsilon$  has a larger influence on the storage amount when the mean grain size is smaller.

Table III: Bounding box storage (gigabyte) as a function of  $\epsilon$  and the initial mean grain size (grid points), compared with conventional storage.

starting $t_s$	mean grain size	threshold value $\epsilon$				conventional
		$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	
$t_s = 200$	$3.5 \times 10^3$	0.84	1.27	1.93	4.18	13.4
$t_s = 400$	$7.8 \times 10^3$	0.70	1.00	1.37	1.90	13.4
$t_s = 600$	$1.4 \times 10^4$	0.63	0.86	1.14	1.50	13.4
$t_s = 800$	$2.0 \times 10^4$	0.59	0.77	0.98	1.24	13.4

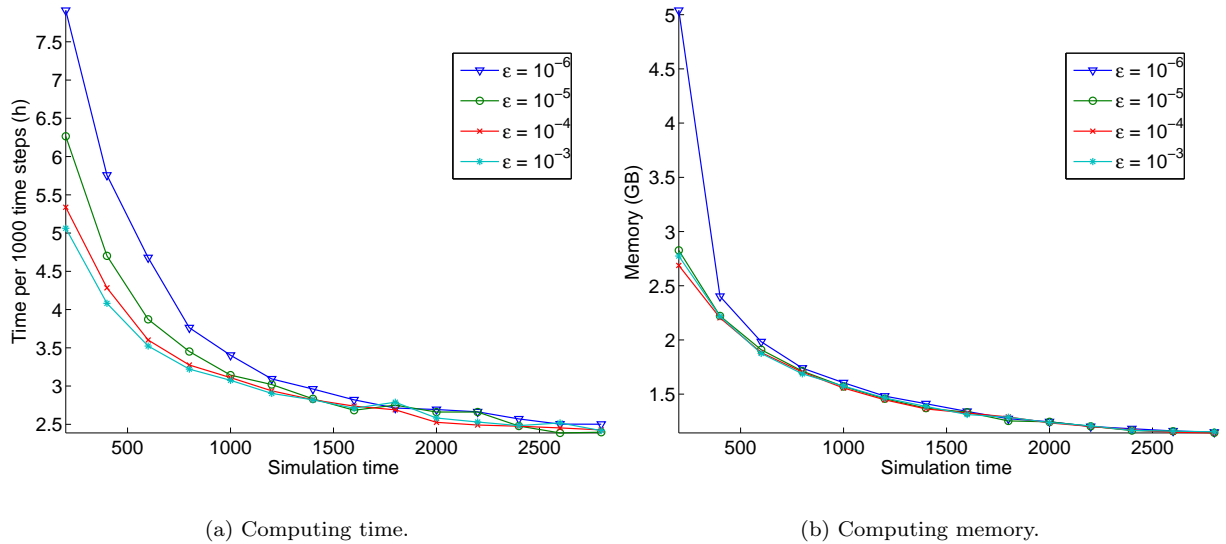


Figure 7: Evolution of computing time and computing memory per 1000 time steps of the bounding box algorithm for  $t_s = 200$  and different values of  $\epsilon$ .

## B. Computation step

If the initial microstructure is obtained by simulation, the total amount of computing resources spent on a grain growth simulation depends on the starting time point  $t_s$ . When  $t_s$  is chosen relatively small, little effort is spent on a time and memory consuming conventional simulation. The computational requirements are also influenced by the threshold value  $\epsilon$ . Figure 7 shows the evolution of the computing time and computing memory per 1000 time steps of the bounding box algorithm for  $t_s = 200$  and different values of  $\epsilon$ . At the start of the bounding box algorithm, considerably more computing time and computing memory is required, because of the smaller mean grain size. Furthermore, for lower threshold values  $\epsilon$ , the computational requirements are higher. Later on in the simulations, the requirements are approximately the same for the different parameters.

The conventional computational requirements are constant in time, whereas the bounding box algorithm requires less resources as simulation time progresses. In Table IV, the computational requirements for the first 5000 time steps of a conventional simulation and a bounding box simulation are shown for  $\epsilon = 10^{-5}$  and  $t_s = 200$ . The conventional algorithm was run on 20 processors (see section IV), in contrast to the bounding box algorithm which could be executed on a single processor.

## VII. APPLICATION

In [31], a sparse algorithm similar to the bounding box algorithm presented in this paper, has been applied to study ideal grain growth in single phase materials. To illustrate the applicability and efficiency of the bounding box algorithm beyond ideal grain growth, we applied the method to grain growth in the presence of second-phase

Table IV: Computational requirements of the first 5000 time steps of the bounding box algorithm with  $\epsilon = 10^{-5}$  as a function of the simulation time and the mean grain size (grid points), compared to the conventional algorithm.

simulation time	mean grain size	bounding box (on 1 processor)		conventional (on 20 processors)	
		time ( $h$ )	memory (gigabyte)	time ( $h$ )	memory (gigabyte)
200 $\rightarrow$ 400	$3.52 \times 10^3 \rightarrow 7.82 \times 10^3$	6.3	2.8	6.8	30.3
400 $\rightarrow$ 600	$7.82 \times 10^3 \rightarrow 1.35 \times 10^4$	4.7	2.2	6.8	30.3
600 $\rightarrow$ 800	$1.35 \times 10^4 \rightarrow 1.95 \times 10^4$	3.8	1.9	6.8	30.3
800 $\rightarrow$ 1200	$1.95 \times 10^4 \rightarrow 2.66 \times 10^4$	3.5	1.7	6.8	30.3
1200 $\rightarrow$ 1400	$2.66 \times 10^4 \rightarrow 3.46 \times 10^4$	3.1	1.6	6.8	30.3

particles. The addition of impurities, which leads to the formation of finely dispersed second-phase particles, is a common technique to control the grain size. The particles pin the grain boundaries and when a limiting grain size is reached, grain growth stops. The limiting grain size depends on the number, size, shape and spatial distribution of the particles. In [22, 23], a phase field model is presented for simulating grain growth in materials containing small incoherent second-phase particles which are constant in time. The interaction between a single particle and a grain boundary is investigated and the results of two-dimensional simulations of the pinning effect of the particles on grain growth are discussed. Unfortunately, the computational requirements of three-dimensional simulations with the conventional algorithm are prohibitive. With the bounding box algorithm, three-dimensional simulation results can be obtained for relevant comparison with experimental data.

Simulations were run on a  $256 \times 256 \times 256$  grid with 100 phase field variables and volume fractions of  $f_V = 5\%$ ,  $8\%$ , and  $12\%$  of second-phase particles with a radius  $r$  of 3 grid points. The parameter values were set to  $\kappa = 0.5$  and  $L = 1$ . The discretization spacings were  $\Delta x = 1$  and  $\Delta t = 0.2$ . First, the conventional algorithm was executed on 20 processors and applied until  $t_s = 1800$ . Second, the bounding box algorithm was applied and run on only one processor until no further growth was observed, for  $\epsilon = 10^{-5}$ . The evolution of the microstructure is illustrated for  $f_V = 8\%$  in Fig. 8. Figure 9 shows the evolution of the radius of the mean grain size for the different volume fractions  $f_V$ . For larger values of  $f_V$ , the grain growth stops earlier and the final grain size is smaller.

In Fig. 10, the evolution of the normalized grain size distribution is shown for  $f_V = 8\%$ . The normalized grain size distribution for a single-phase system is shown on one time point, since for single-phase systems, the distribution is constant in time [1]. As predicted by mean field theories [3, 37, 38], the simulation results indicate that, when second-phase particles are present, the peak of the distribution shifts towards smaller grain sizes.

To validate the computed results, we compare the obtained limiting grain sizes to existing theories and observations. Figure 11 shows the ratio of the limiting mean grain size to the particle radius as a function of the volume fraction as computed by the phase field simulations, the classical Zener relation, the relation obtained through three-dimensional Monte Carlo simulation in [39] and a relation obtained by fitting experimental data from [40]. The data obtained with our simulations correspond well to the Monte Carlo results, but deviate strongly from the experimental data. This indicates that the Monte Carlo model, the phase field model and the Zener relation are sufficient for round-shaped particles and isotropic grain boundary energies. However, to simulate real, anisotropic materials, which contain plate- or lens-shaped particles, the phase field model has to be expanded. The difference between the Monte Carlo results and the phase field model results is probably due to the fact that the particles had a slightly different shape in both simulation experiments, as after discretization the small particles are no longer completely spherical.

In [24], similar results were obtained using a dynamic grain-orientation reassignment. However, the computational requirements of the latter algorithm (approximately 48 gigabyte, on 16 processors) are more than ten times larger than the requirements of the bounding box algorithm (2 gigabyte, on 1 processor).

## VIII. CONCLUSION

In this paper, a bounding box method is presented to solve a phase field model describing grain growth in polycrystalline materials. The bounding box method is based on the observation that the solutions of the model display small regions of high activity surrounded by large regions of inactivity. Because of its object-oriented design, the bounding box algorithm is extensible to more complex models. With the bounding box method, no additional computational requirements are posed when the number of phase field variables is increased to run coalescence-free simulations.

To illustrate that the bounding box method is applicable to grain growth simulations for realistic materials, which usually contain impurities or show orientation dependence, the algorithm was applied to a three-dimensional phase

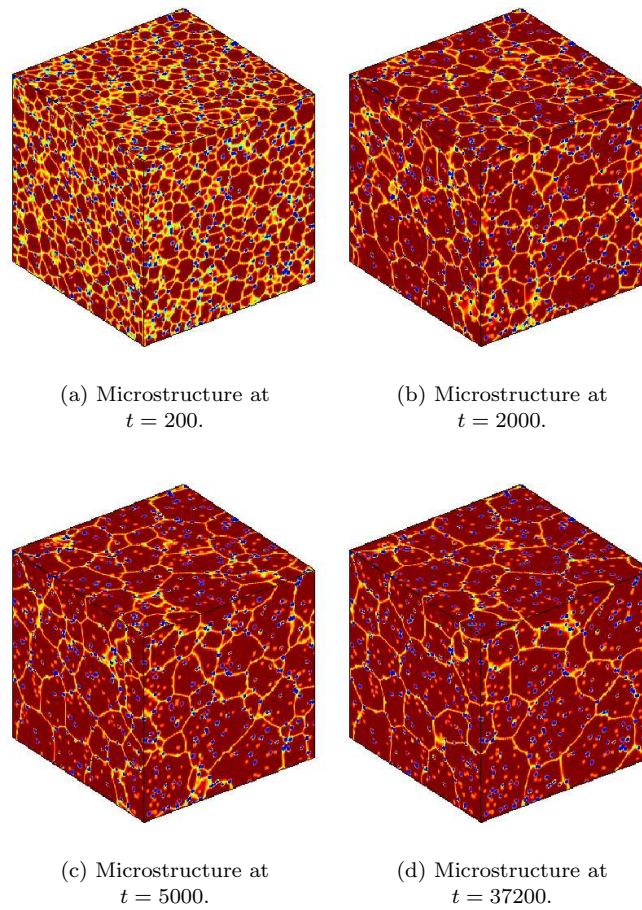


Figure 8: Evolution of  $256 \times 256 \times 256$  microstructure with volume fraction of  $f_V = 8\%$  of second-phase particles, shown at  $t = 200, 2000, 5000$ , and  $37200$ .

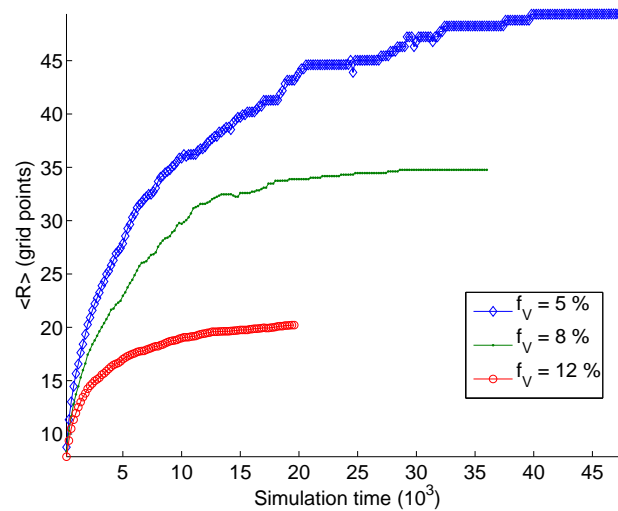


Figure 9: Evolution of the radius of the mean grain size for volume fractions of  $f_V = 5\%$ ,  $8\%$ , and  $12\%$ .

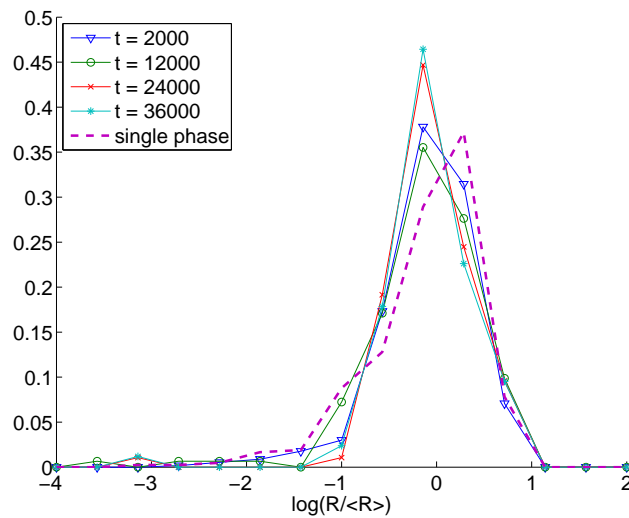


Figure 10: Evolution of the normalized grain size distribution for  $f_V = 8\%$ , compared with the normalized grain size distribution for a single-phase system.

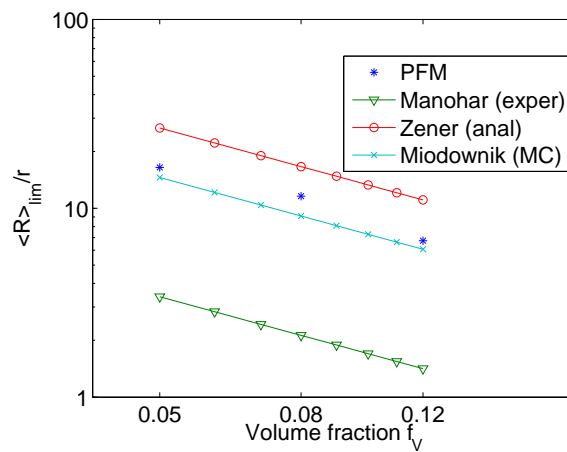


Figure 11: Ratio of the radius of the limiting mean grain size to the particle radius as a function of the volume fraction  $f_V$ , compared for different theories.

field model of grain growth in the presence of second-phase particles. The effect of the particles on the growth kinetics was studied.

### Acknowledgments

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

Nele Moelans is Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

The simulations were performed on the computer cluster of the HPC Computing Initiative, funded by the

- 
- [1] F. J. Humphreys and M. Hatherly, *Recrystallization and related annealing phenomena* (Pergamon, Elsevier Science Ltd., 1996).
- [2] M. A. Miodownik, *Scripta Mater.* **54**, 993 (2006).
- [3] M. Hillert, *Acta Metall.* **13**, 227 (1965).
- [4] M. Guo and H. Suito, *ISIJ Int.* **39**, 1297 (1999).
- [5] J. Gruber, D. George, A. Kuprat, G. Rohrer, and A. Rollett, in *Recrystallization and Grain Growth*, edited by B. Bacroix, J. Driver, R. L. Gall, C. Maurice, R. Penelle, H. Réglé, and L. Tabourot (Materials Science Forum, 2004), vol. 467-470, pp. 733–738.
- [6] D. Danilov and B. Nestler, *Acta Mater.* **54**, 4659 (2006).
- [7] M. Plapp, *J. Cryst. Growth* **303**, 49 (2007).
- [8] B. Böttger, J. Eiken, and I. Steinbach, *Acta Mater.* **54**, 2697 (2006).
- [9] Q. Bronchard, Y. L. Bouar, and A. Finel, *Adv. Eng. Mater.* **8**, 1245 (2006).
- [10] Y. Wen, B. Wang, J. Simmons, and Y. Wang, *Acta Mater.* **54**, 2087 (2006).
- [11] N. Ma, A. Kazaryan, S. A. Dregia, and Y. Wang, *Acta Mater.* **52**, 3869 (2004).
- [12] D. Fan and L.-Q. Chen, *Acta Mater.* **45**, 611 (1997).
- [13] C. E. K. III and L.-Q. Chen, *Acta Mater.* **50**, 3057 (2002).
- [14] L.-Q. Chen and W. Yang, *Phys. Rev. B* **50**, 15752 (1994).
- [15] R. Kobayashi, J. A. Warren, and W. C. Carter, *Physica D* **119**, 415 (1998).
- [16] I. Steinbach, F. Pezzolla, B. Nestler, M. Seeßelberg, R. Prieler, G. J. Schmitz, and J. L. L. Rezende, *Physica D* **94**, 135 (1996).
- [17] D. Fan and L.-Q. Chen, *Acta Mater.* **45**, 1115 (1997).
- [18] L.-Q. Chen and D. Fan, *J. Am. Ceram. Soc.* **79**, 1163 (1996).
- [19] D. N. Fan and L.-Q. Chen, *Acta Mater.* **45**, 3297 (1997).
- [20] D. Fan, S. P. Chen, L.-Q. Chen, and P. W. Voorhees, *Acta Mater.* **50**, 1895 (2002).
- [21] D. Fan, L.-Q. Chen, and S. P. Chen, *J. Am. Ceram. Soc.* **81**, 526 (1998).
- [22] N. Moelans, B. Blanpain, and P. Wollants, *Acta Mater.* **53**, 1771 (2005).
- [23] N. Moelans, B. Blanpain, and P. Wollants, *Acta Mater.* **54**, 1175 (2006).
- [24] Y. Suwa, Y. Saito, and H. Onodera, *Scripta Mater.* **55**, 407 (2006).
- [25] N. Provatas, N. Goldenfeld, and J. Dantzig, *Phys. Rev. Lett.* **80** (1998).
- [26] N. Provatas, N. Goldenfeld, and J. Dantzig, *J. Comput. Phys.* **148**, 265 (1999).
- [27] R. J. Braun and B. T. Murray, *J. Cryst. Growth* **174**, 41 (1997).
- [28] G. Beckett, J. A. Mackenzie, and M. L. Robertson, *Commun. Comput. Phys.* **1**, 805 (2006).
- [29] W. M. Feng, P. Yu, Z. K. Liu, Q. Du, and L.-Q. Chen, *J. Comput. Phys.* **220**, 498 (2006).
- [30] J. Gruber, N. Ma, Y. Wang, A. D. Rollett, and G. S. Rohrer, *Model. Simul. Mater. Sc.* **14**, 1189 (2006).
- [31] S. G. Kim and D. I. Kim, *Phys. Rev. E* **74** (2006).
- [32] S. Vedantam and B. S. V. Patnaik, *Phys. Rev. E* **73** (2006).
- [33] N. Moelans, Ph.D. thesis, K.U.Leuven (2006).
- [34] L.-Q. Chen and J. Shen, *Comput. Phys. Commun.* **108**, 148 (1998).
- [35] M. I. M. Copetti and C. M. Elliot, *Mater. Sci. & Technol.* **6**, 273 (1990).
- [36] D. Zöllner and P. Streitenberger, *Scripta Mater.* **54**, 1697 (2006).
- [37] O. Hunderi and N. Ryum, *Acta Metall.* **30**, 739 (1982).
- [38] G. Abbruzzese, *Acta Metall.* **33**, 1329 (1985).
- [39] M. Miodownik, E. A. Holm, and G. N. Hassold, *Scripta Mater.* **42**, 1173 (2000).
- [40] P. A. Manohar, M. Ferry, and T. Chandra, *ISIJ Int.* **38**, 913 (1998).