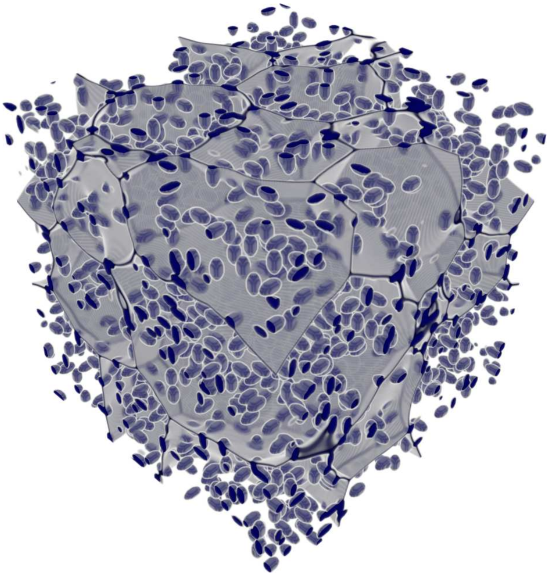
KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science

Acceleration strategies for phase field simulation of grain growth in polycrystalline materials



Liesbeth Vanherpe

Dissertation presented in partial
fulfilment of the requirements for
the degree of Doctor
in Engineering

September 2010

Acceleration strategies for phase field simulation of grain growth in polycrystalline materials

Liesbeth Vanherpe

Jury:
Prof. dr. ir. H. Hens, chair
Prof. dr. ir. S. Vandewalle, supervisor
Prof. dr. ir. B. Blanpain, supervisor
Prof. dr. ir. D. Roose
Prof. dr. ir. P. Wollants
Prof. dr. ir. N. Moelans
Prof. dr. W. Vanroose
(Universiteit Antwerpen)
Prof. dr. I. Steinbach
(Ruhr-Universität Bochum)

Dissertation presented in partial
fulfilment of the requirements for
the degree of Doctor
in Engineering

September 2010

Kasteelpark Arenberg 1 box 2200, BE-3001 HEVERLEE (Belgium)

D/2010/7515/69
ISBN 978-94-6018-230-3

Voorwoord

Met deze doctoraatstekst sluit ik vijf geweldige jaren af. Ik was er niet aan kunnen beginnen als ik de kans ertoe niet had gekregen van Prof. Stefan Vandewalle, en kort daarna ook van Prof. Bart Blanpain. Bedankt voor alle kansen, de uitdagingen en de ondersteuning! Een speciaal woord van dank gaat hierbij naar Prof. Nele Moelans. Zij reikte me het onderwerp van mijn onderzoek aan en zorgde voortdurend voor inspiratie en motivatie.

Graag bedank ik alle juryleden voor hun interesse en waardevolle opmerkingen, en de voorzitter van de jury, Prof. Hugo Hens. I cordially thank all jury members, Prof. Dirk Roose, Prof. Patrick Wollants, Prof. Wim Vanroose, and Prof. Ingo Steinbach for their interest and valuable comments. I also want to thank Prof. Britta Nestler for giving me the opportunity to come to Karlsruhe, and the Karlsruhe group for their cordial reception.

Ik heb tijdens mijn doctoraat kunnen genieten van een fijne werkomgeving. Ik dank graag alle bezoekers, oud-collega's en collega's van het departement Computerwetenschappen, en in het bijzonder van de onderzoeksgroep Technisch-Wetenschappelijk Rekenen voor de kwinkslagen bij de koffie, de behulpzame raad bij onderzoeksproblemen en het dagelijks opvangen van kleine en grote problemen. Bedankt Koen (aan wie ik de mooie prent op de voorpagina van dit boekje te danken heb), Dirk, Wim, Joaquin, Kris, Giovanni, Samuel, en Pieter. Ik wil ook speciaal de collega's van de onderzoeksgroep THERMO, op het departement Metaalkunde en Toegepaste Materiaalkunde, vermelden. Ik kreeg er de kans om in een sympathieke sfeer meer dan gewoon een blik over de muren van andere onderzoeksdisciplines te werpen.

During the yearly PhDays, I discovered many compatriots on the doctoral battlefields. Keep the fight up!

Ik had me overdag niet zo hard kunnen concentreren als ik me daarna niet had kunnen uitleven in de vriendengroep van het Universitair Symfonisch Orkest en later ook die van het Leuvens Alumni Orkest op die talrijke avonden, nachten, weekends, en orkestreizen over de hele wereld, tot zelfs in China. Ik geef graag ook een knipoog naar de klasgenoten van het Centrum voor Levende Talen. Ik kan nog altijd niet vlot Russisch spreken, maar ik hoef er tenminste niet in mijn eentje om te lachen. De teamleden van de FunLovinCWers, een initiatief van het laatste uur, maar daarom niet minder welkom, zorgden voor prettige afleiding en een hoop plezier. Bedankt aan al mijn vrienden voor de steun en de gezelligheid!

Ik wil speciaal de mensen vermelden waarmee ik deze reis gestart ben en waarmee ik die ook min of meer samen zal afsluiten: Yves, Eveline, Joris, en Bart. Zij weten waarom.

Ik bedank heel graag mijn ouders, broer en zussen, die me onvoorwaardelijke steun en de omgeving hebben gegeven waarin ik kon groeien tot wie ik nu ben.

Tot slot wil ik Thomas bedanken voor het grote geduld, de niet-aflatende steun en de liefde van elke dag.

Liesbeth Vanherpe
Augustus 2010

Abstract

The microstructure of many materials consists of multiple grains with different crystallographic orientations. Under certain circumstances, such as increased temperature, the smaller grains will shrink and disappear under the influence of surface tension. One of the modelling techniques that are explored for the simulation of this phenomenon, called grain growth, is phase field modelling. Two phase field models in particular are the main interest of this thesis, namely the continuum field model and the multi-phase field model. Both models represent a polycrystalline microstructure with a large set of phase field variables, where each variable corresponds to a single crystallographic orientation. However, realistic three-dimensional grain growth simulations with these models can demand significant amounts of computation power.

In this thesis, we present a sparse bounding box algorithm designed to perform efficient phase field simulations of grain growth. The algorithm shows significant improvements over existing techniques as its computational requirements scale with the grid size instead of with the number of crystallographic orientations involved. Furthermore, a nonlinear multigrid solver, based on the Full Approximation Scheme, is constructed to solve the multi-phase field model for multiple phase field variables. Experiments with this solver show that its convergence rates are independent of the grid size.

The applicability of the bounding box algorithm is illustrated by three-dimensional simulations of grain growth in the presence of spheroid second-phase particles and of grain growth in a microstructure with anisotropic boundary energy. From the former simulations, it is found that the pinning effect of a particle distribution is stronger for increasing volume fraction, and for increasing aspect ratio of the particles. Furthermore, a generalised Zener type relation is proposed. The second type of simulations is performed in a microstructure whose boundary energy is described by a Read-Shockley type dependence. Simulation results show that the low-angle boundaries are clearly preferred during grain growth. The anisotropic formulation of the boundary energy is furthermore observed to change the individual growth rates of the grains as a function of the number of grain faces.

Samenvatting

De microstructuur van vele materialen bestaat uit korrels met verschillende kristallografische oriëntaties. Onder bepaalde omstandigheden, zoals verhoogde temperatuur, zullen de kleinere korrels krimpen en verdwijnen onder de invloed van de oppervlaktespanning. Een van de modelleringstechnieken die onderzocht worden voor de simulatie van dit fenomeen, ook wel korrelgroei genoemd, is faseveldmodellering. Twee faseveldmodellen in het bijzonder vormen het onderwerp van deze thesis, namelijk het continuümveldmodel en het multifaseveldmodel. Beide modellen stellen een polykristallijne microstructuur voor met behulp van een grote verzameling faseveldvariabelen, waarbij elke variabele overeenkomt met één kristallografische oriëntatie. Realistische driedimensionale korrelgroeisimulaties met deze modellen zijn echter vaak heel rekenintensief.

In deze thesis stellen we een algoritme voor dat op basis van insluitende blokken toelaat om efficiënte faseveldsimulaties van korrelgroei uit te voeren. Het algoritme is een belangrijke verbetering ten opzichte van bestaande technieken, omdat zijn computationele vereisten evenredig zijn met de roostergrootte in plaats van met het aantal kristallografische oriëntaties. Verder wordt een niet-lineaire multiroostermethode geconstrueerd om het multifaseveldmodel op te lossen voor meerdere faseveldvariabelen. Experimenten met deze oplossingsmethode vertonen convergentiesnelheden die onafhankelijk zijn van de roostergrootte.

De toepasbaarheid van het blokalgoritme wordt geïllustreerd door driedimensionale simulaties van korrelgroei in de aanwezigheid van sferoïde tweedefasepartikels en van korrelgroei in een microstructuur met anisotrope korrelgrensenergie. De resultaten van de eerstgenoemde simulaties geven aan dat het pinnend effect van een partikelverdeling sterker wordt voor stijgende volumefractie, en voor stijgende aspectverhouding van de partikels. Na analyse wordt een veralgemeende Zenerrelatie voorgesteld. Het tweede type simulatie wordt uitgevoerd in een microstructuur waarin de grensenergie beschreven wordt door de Read-Shockley-afhankelijkheid. De resultaten tonen dat de lagehoekgrenzen duidelijk verkozen worden tijdens korrelgroei. De anisotrope grensenergie beïnvloedt daarbij de individuele groeisnelheden van de korrels als functie van hun aantal grensvlakken.

Contents

Nomenclature

Grain growth

A	grain area	[m^2]
a, c	axis radii of a spheroid particle	[m]
b	volume fraction exponent in the Zener relation	
f_V	volume fraction of particles	
F_Z	pinning force of a single spheroid particle	[N]
F_Z^S	pinning force of a single spherical particle	[N]
K	prefactor in the Zener relation	
l	long axis radius of a spheroid particle	[m]
n	grain growth exponent	
P_g	driving pressure for grain growth	[J m^{-2}]
r	radius of a spherical particle	[m]
r_a	$= c/a = l/s$, aspect ratio of a spheroid particle	
R	grain radius	[m]
$\langle R \rangle$	mean grain radius	[m]
$\langle R \rangle_\mathrm{lim}$	limiting mean grain radius at arrest of grain growth	[m]
s	short axis radius of a spheroid particle	[m]
V	grain volume	[m^3]
σ_gb	grain boundary energy (grain boundary tension)	[J m^{-2}]

Phase field modelling

F	free energy of a system
p	number of phase field variables
$\mathbf{r}$	$= (x, y, z)$, position vector
t	time

Multi-phase field model

a	gradient free energy density
$\mathcal{G}$	Gibbs-simplex
s	scaling parameter
w_{obst}	multi-obstacle potential
w_{well}	multi-well potential
ϵ	interface width
λ	Lagrange multiplier function
μ	interface mobility
σ_{ij}	surface energy density of the interface created by phase field variables ϕ_i and ϕ_j
σ_{ijk}	surface energy density of the interface created by phase field variables ϕ_i, ϕ_j and ϕ_k
ϕ_i	component of vector-valued order parameter, associated with orientation i
$\boldsymbol{\phi}$	vector-valued order parameter
ω	$= 1/\mu$, mobility parameter

Continuum field model

f_0	homogeneous free energy density
L	kinetic coefficient, related to the grain boundary mobility
m	parameter of the free energy density f_0
α, β, γ	parameters of the free energy density f_0
η_i	phase field variable, associated with orientation i
$\boldsymbol{\eta}$	set of phase field variables
θ	smallest misorientation angle difference between the crystallographic orientations of two neighbouring grains in a fibre-textured microstructure
Θ	ensemble of parameters that represents boundary misorientation
κ	gradient energy coefficient
Φ	spatially dependent parameter representing second-phase particles ($\Phi = 1$ within particles and $\Phi = 0$ elsewhere)

Computational techniques and stability analysis

D	diffusion part of a system of PDEs
D_n	error estimate at time point n
k	wave number
l	interval length of one-dimensional periodic grid
$p(\mathbf{r})$	interpolation function
p	order of time discretisation scheme
$r_{\min}, r_{\max}, \theta$	parameters to control adaptive time step size
r_n	$= \Delta t_n / \Delta t_{n-1}$, ratio between current and previous time step size at time point n
R	reaction part of a system of PDEs
V_r^n	small perturbation, added to an equilibrium value to study stability properties
Δt	time step size
$\Delta x, \Delta y, \Delta z$	spatial discretisation size
ϵ	small threshold value
ι	$= \sqrt{-1}$, imaginary unit
τ	tolerance value
$\Psi(\mathbf{r}, t)$	sum of the squared values of phase field variables, used to compute error estimates and to visualise a microstructure
Ω	system domain
$\hat{\ }$	refers to the Fourier transform

Bounding box algorithm

$B_i(t)$	set of couples $(\mathbf{r}, \eta_i(\mathbf{r}, t))$ associated with bounding box $\mathcal{B}_i(t)$
$\mathcal{B}_i(t)$	bounding box associated with grain region $G_i(t)$
$G_i(t)$	grain region where phase field variable η_i is active
K	number of processors involved in a parallel simulation
M	margin of grid points added in each direction to bounding box during preparation step in bounding box simulation
$\mathbf{r}_{i,\min}(t)$, $\mathbf{r}_{i,\max}(t)$	two opposite grid points delimiting bounding box $\mathcal{B}_i(t)$
t_s	time point at which the conventional algorithm is stopped and the bounding box algorithm is started
ϵ	small threshold value

Multigrid method

A	system matrix
b	right-hand side of a system
e	error vector
J_N	Jacobian of nonlinear operator N
N	nonlinear operator
N	$= N_x = N_y$, number of grid points in every dimension in an equisized grid
N_x, N_y	number of grid points in the dimensions along the x- and the y-axis
r	residual vector
V, W	cycle types
γ	multigrid parameter indicating the number of multigrid cycles carried out on the current grid
ν_1	number of presmoothing iterations
ν_2	number of postsmoothing iterations
$_{\mathrm{c}}$	refers to the coarse grid
$_{\mathrm{f}}$	refers to the fine grid

Simulation results

a	parameter of generalised Zener relation
D	linear dimension of a grain
f	number of grain faces
g.p.	grid point
m	particle measure to replace particle radius in Zener relation
ϕ_2	fraction of particles at grain face
ϕ_3	fraction of particles at triple junctions
ϕ_4	fraction of particles at quadruple junctions
ϕ_n	fraction of particles at junctions where more than four grains meet
ϕ_{tot}	total fraction of particles in contact with a grain boundary

Abbreviations

BDF	Backward Differentiation Formula
DFT	Discrete Fourier Transform
FAS	Full Approximation Scheme
FFT	Fast Fourier Transform
HAB	high-angle boundary
IMEX	Implicit-Explicit
LAB	low-angle boundary
MD	misorientation distribution
MPI	Message Passing Interface
OpenMP	Open Multi-Processing
UML	Unified Modelling Language

Chapter 1

Introduction

1.1 Context

Materials research is often the source of technological progress. Research into the production of new and more advanced materials for example enabled enhanced performance of transport media. A modern bicycle is almost unrecognisable next to the first bicycles at the start of the nineteenth century. Aeroplanes cross the air and even space is gradually explored by aircraft. Without decent knowledge of materials, a rocket would fall to pieces or burn. In medicine, porous and other materials are used to make prostheses through an accurate production process. Architecture likes to employ modern, light materials with special properties with regard to isolation and interaction with light.

Powerful computers and enhanced computing techniques contribute considerably to materials research. Many interesting technological innovations originate from the marriage of materials science and computer science. A recent example are the nanotechnology simulations of [115], which surprisingly provided an explanation for mechanisms that could not be identified by laboratory observation.

The microstructure of many materials consists of multiple grains with different crystallographic orientations. The study of the factors influencing the evolution of a grain structure is of great technological importance, because many material properties, such as corrosion resistance, conductivity, strength and toughness, depend on the mean grain size and the grain size distribution. The ultimate goal of this research is to provide predictive models that can be used to design production processes of materials with a microstructure tailored to any desired property.

1.2 Grain growth

A specific example of why knowledge of microstructural evolution is of importance, is the production process of spokes [82]. Spoke manufacturing involves the drawing of steel rods through a series of dies of decreasing hole size until a wire with the desired diameter is produced. The deformation caused by the tension in the rod has a strengthening effect, but the wire can become so strain-hardened that there is danger of rupture. Therefore, during the drawing process, the wire must be periodically softened. This is accomplished by heating the wire above a threshold temperature in a process known as annealing. Annealing consists of three stages, namely recovery, recrystallisation and grain growth.

At temperatures below the recrystallisation range, the cold-worked metal undergoes the process of recovery, in which excess lattice vacancies caused by the deformation anneal out. The driving force for recovery is the stored energy of the cold deformation work. Most of the softening of the material however occurs in a narrow temperature range by the process of recrystallisation, in which new grains nucleate and grow into the cold-worked microstructure.

At high annealing temperatures, after recrystallisation is completed, the polycrystalline structure is not yet stable. Further growth of the recrystallised grains may occur, which is driven by the interfacial energy associated with grain boundaries. Grain growth produces gradual softening of the material for two reasons. First, the irregularities on grain boundaries are prime sources of dislocations during deformation. Second, grain boundaries act as barriers to slip bands because of the mismatch of slip systems from grain to grain. The rate of grain growth can be decreased by for example the presence of second-phase particles at grain boundaries. This effect tends to diminish as the annealing temperature is raised.

The study of processes such as grain growth will provide better insight into the key factors of microstructural evolution and facilitate the design of specially adapted production processes, such as the wire drawing process. A large number of theoretical, experimental and computational studies have been performed on the subject of grain growth [58, 84, 54, 48, 44]. Although it is well understood that grain growth behaviour results from the interplay between curvature driven grain boundary movement and the geometrical requirements at boundary junctions, there are still many controversies, especially with respect to the shape and evolution of the grain size distribution.

1.3 Numerical modelling and simulation of grain growth

Computer simulations are an essential ingredient in the study of grain growth. They enable parameter studies and provide three-dimensional insight into the evolution of a microstructure, which can be difficult to obtain from experiments. One of the modelling techniques that has been explored for simulating microstructural evolution is phase field modelling. This technique has shown to be a useful tool for the simulation of various phenomena, such as solidification, precipitation and grain growth.

Grain growth in single-phase polycrystalline materials has been investigated with the phase field method by different authors [25, 86, 89, 71, 120, 42, 99]. The models of both [120, 42, 99] and [25, 86, 89, 90] describe a polycrystalline microstructure with a large set of phase field variables. In principle, the number of phase field variables included in a grain growth simulation should equal or exceed the total number of grains, as in reality the number of possible grain orientations is infinite. As a consequence, realistic three-dimensional computer simulations of grain growth with a phase field model can demand significant amounts of computation power.

Several algorithms have been designed to overcome the computational limitations of the phase field method. In [74, 122], 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 neighbouring grains into this technique is very difficult. Adaptive meshing [104, 105, 17], and moving mesh techniques [11, 38], have been used in phase field simulations to increase computational efficiency. However, so far these techniques were always applied for simulations with a small number of phase field variables. For polycrystalline structures their benefits are drastically reduced, because of the amount of interface involved.

1.4 Motivation and goals

At the Department of Metallurgy and Materials Engineering of the K.U.Leuven, much research is performed to develop predictive phase field models of grain growth [86, 87, 89, 90]. A similar type of phase field models is employed at the Institute of Materials and Processes of the University of Applied Sciences in Karlsruhe, Germany [42, 99, 97]. As mentioned above, realistic three-dimensional grain growth simulations with these models demand significant amounts of computation power. The purpose of this thesis is to design, analyse and implement efficient, numerical solvers for both types of phase field models. In particular, different techniques are explored to reduce the computational requirements of the phase field models, such as the use of sparse algorithms, multigrid methods, and parallel computing. Those are techniques that have been studied and developed for many years already in the Scientific Computing group at the Department of Computer Science of the K.U.Leuven.

The goals of this thesis are not limited to the construction and analysis of computational methods, but also cover aspects of materials research. With the help of the newly developed software, we will address certain questions that deal with the effect of second-phase particles and the presence of texture on grain growth.

1.5 Outline of the text

The text starts in **Chapter 2** with an introduction to the mechanisms of grain growth. Two of the influencing factors of grain growth are discussed, namely the effect of second-phase particles at grain boundaries, and the presence of texture. Next, several types of grain growth models are described, and the main two phase field models that are explored in the rest of the thesis are presented, namely the multi-phase field model [120] and the continuum field model [25, 86, 87, 89, 90].

Chapter 3 gives an overview of the computational techniques that have been applied to alleviate the computational requirements of the phase field models described in Chapter 2. To obtain more insight in the benefits of the application of IMEX time integration methods in particular, the stability properties of the continuum field model are theoretically and numerically studied in the subsequent **Chapter 4**.

In **Chapter 5**, a sparse bounding box algorithm is developed that exploits the characteristics of the solutions of the phase field models for grain growth. The new algorithm is tested for the continuum field model and compared to a conventional method, and proves to reduce the computational requirements of simulations with the latter model drastically.

Chapter 6 covers the construction of a nonlinear multigrid solver based on the Full Approximation Scheme (FAS) for two-dimensional phase field simulations with the multi-phase field model. The implemented solver shows the desired grid size independent properties.

The sparse algorithm developed in Chapter 5 is applied to grain growth in the presence of spheroid particles in **Chapter 7**, and to grain growth in a microstructure with anisotropic boundary energy properties in **Chapter 8**. In both chapters, the simulation results are analysed and discussed.

We end in **Chapter 9** with a summary of the contributions of this thesis, and some suggestions and directions for future research.

Chapter 2

Phase field modelling of grain growth

2.1 Introduction

The purpose of this chapter is to provide an introduction to the phenomenon of grain growth. Furthermore, the construction of computer models for grain growth simulation is discussed. In particular, the main two phase field models that are explored in the rest of the thesis are presented.

The microstructure of many materials consists of multiple grains with different crystallographic orientations. Under certain circumstances, such as increased temperature, the smaller grains will shrink and disappear under the influence of surface tension. This phenomenon, called grain growth, is thus characterised by an increase of the mean grain size.

Grain growth may be divided into two types, normal grain growth and abnormal grain growth [58]. During normal grain growth, a polycrystalline microstructure changes in a rather uniform way. There is a relatively narrow range of grain sizes and shapes, and the form of the grain size distribution is usually independent of time and scale. During abnormal grain growth, a few grains in the microstructure grow and consume the smaller grains. Consequently, a bimodal grain size distribution develops, with one peak corresponding to the smaller grains and the second peak corresponding to the larger grains. Eventually the latter grains will impinge and normal grain growth may then resume. The phenomenon of abnormal grain growth will not be considered here.

Computer simulations based on mesoscale models such as Monte Carlo Potts, front-tracking and phase field models, are essential for a better understanding of grain growth. They allow to study the roles of different parameters separately, which can be very complex 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. Consequently, there have been many efforts to develop models with predictive accuracy for grain growth under realistic conditions.

This chapter starts with an introduction on grain growth in Section 2.2. In this section, some of the factors that influence grain growth are discussed, namely the effect of second-phase particles, and the presence of texture. The text continues with a short discussion on several types of grain growth models and introduces phase field modelling in Section 2.3. Next, we describe two different phase field models in detail, namely the multi-phase field model of [120] in Section 2.4 and the continuum field model of [25] in Section 2.5. The latter section also covers the extensions of the continuum field model proposed in [86, 87] to simulate grain growth in the presence of second-phase particles, and the extensions proposed in [89, 90] to simulate grain growth in a microstructure with anisotropic boundary energy. Section 2.6 ends this chapter with conclusive remarks.

2.2 Grain growth

2.2.1 Normal grain growth

Grain growth mechanism

As mentioned in Chapter 1, grain growth is usually one step in the production process of a polycrystalline material. The driving force for grain growth is the reduction of the free energy, which is stored in the material in the form of grain boundaries. The boundary between one grain and its neighbour is in fact a defect in the crystal structure. Therefore, with every grain boundary, there is an associated surface energy, or a surface tension σ_{gb} [82]. Due to the curvature of a grain boundary, there is a resulting driving pressure P_g that drives the boundary to move towards its centre of curvature. This is illustrated in Fig. 2.1.

During grain boundary movement, the surface tensions of intersecting grain boundaries must balance each other [82]. Figure 2.2 illustrates this geometrical requirement at a triple junction in two dimensions. The three involved surface energies σ_{gb}^{ij} and the corresponding angles α_k have to satisfy

$$\frac{\sigma_{\text{gb}}^{12}}{\sin\alpha_3} = \frac{\sigma_{\text{gb}}^{13}}{\sin\alpha_2} = \frac{\sigma_{\text{gb}}^{23}}{\sin\alpha_1}. \tag{2.1}$$

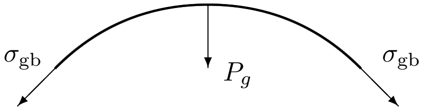


Figure 2.1: The driving pressure P_g resulting from the grain boundary curvature forces the boundary to move towards its centre of curvature.

For isotropic materials, the equality $\sigma_{\text{gb}}^{12} = \sigma_{\text{gb}}^{23} = \sigma_{\text{gb}}^{13}$ holds, and the equilibrium angle between each pair of intersecting grain boundaries is 120°.

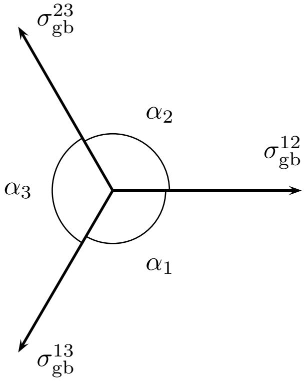

σ_{gb}^{23}
α_2
σ_{gb}^{12}
α_3
α_1
σ_{gb}^{13}

Figure 2.2: Surface tension balance at a triple junction.

Equilibrium condition (2.1) has important consequences for the topology of a grain structure. In two dimensions, grains with less than six sides will be mostly convex, while grains with more than six sides will be concave. In general, larger grains have more sides than smaller grains. Therefore, during grain growth, on average, the larger grains will grow and the smaller grains will shrink and disappear. The interplay of curvature-driven grain boundary movement and geometrical constraints thus results in a decrease of grain boundary area and an increase of the mean grain size.

Grain growth kinetics

The kinetics of grain growth can be expressed by relating the mean grain size to time. In [20], a growth law is deduced that describes the temporal evolution of the mean grain radius $\langle R\rangle(t)$ as

$$\langle R\rangle(t) = \left(\langle R\rangle_0^2 + k't\right)^{\frac{1}{2}} \tag{2.2}$$

where $\langle R\rangle_0$ is the mean grain radius at $t = 0$ and k' a constant value. In the limit $t \to \infty$, where $\langle R\rangle^2(t) \gg \langle R\rangle_0^2$, a more general power law is obtained, namely:

$$\langle R\rangle(t) = kt^{\frac{1}{n}}, \tag{2.3}$$

with grain growth exponent n. Only few measurements of growth kinetics have produced a grain growth exponent of $n = 2$. Instead, n is generally found to be well above 2. This discrepancy has been attributed to processes such as solute drag and precipitate pinning [58].

Concluding remarks

It is a complex matter to construct an analytical model of grain growth. A comprehensive model should include the curvature-driven movement of the grain boundaries, the geometrical requirements at the boundary junctions, as well as the fact that the grains have to be space-filling. Computer simulations offer a way out of this problem. They consider the growth and shrinkage of every grain separately, and allow to study the different parameters of grain growth separately, which is often difficult 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.

2.2.2 The pinning effect of second-phase particles

Zener pinning

Since the driving pressure for grain growth is relatively weak, even extremely small amounts of impurities may have a drastic effect on the grain growth behaviour of a material. Small second-phase particles, such as precipitates and insoluble inclusions, exert a strong pinning effect, also called Zener pinning, on the grain boundaries. They restrain the mobility of grain boundaries and eventually inhibit grain growth, limiting the final mean grain size of the microstructure. As briefly mentioned in Section 2.1, a characteristic of normal grain growth in ideal materials is that the grain size distribution has a shape that is self-similar in time. Processes such as Zener pinning will affect the shape of the grain size distribution such that the evolution of the grain structure is no longer self-similar.

Insight into the pinning effect of particles is of great technological importance, since for many applications a tailored grain size is required to obtain materials with the desired properties. Examples are the addition of a small amount of alloying elements to HSLA (High Strength Low Alloyed) steels and Ni-based super alloys in order to obtain materials with a small grain size and high strength [23, 100, 117], the use of precipitates in thin films to induce abnormal grain growth in order to obtain films with a large grain size for high electrical conductivity and reduced electromigration damage [40, 76, 70], and, recently also the use of particles (such as precipitates, carbon nanotubes and organic or amorphous particles) to stabilise nanocrystalline materials [52, 77, 72, 26].

In most analytical studies [96, 32, 55, 81] on Zener pinning it is assumed that normal grain growth is arrested when a critical mean grain radius $\langle R \rangle_{\mathrm{lim}}$ is reached. Most often a relation of the form

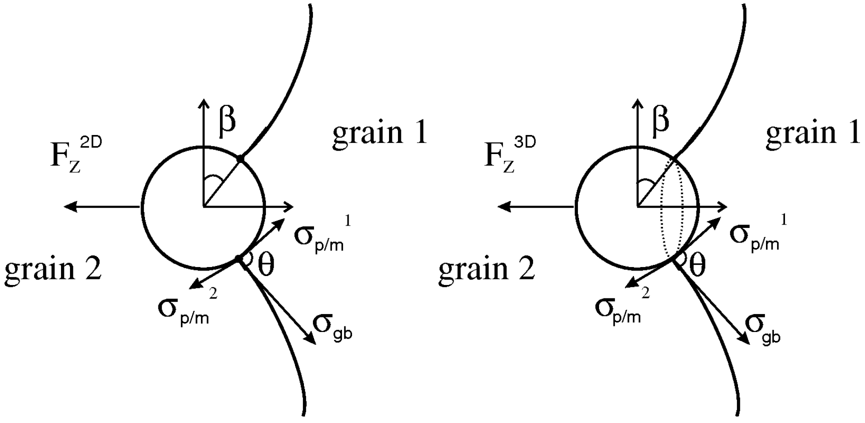
$$\frac{\langle R \rangle_{\mathrm{lim}}}{r} = K \frac{1}{f_{\mathrm{V}}^{b}} \tag{2.4}$$

is obtained, where r is the radius of the second-phase particles and f_{V} the volume fraction. The values of the parameters K and b vary among the different studies, depending on which assumptions are made on the shape and properties of the particles and boundaries, and on the assumptions made on the position of the particles with respect to the boundaries.

The pinning force exerted by one particle on a grain boundary can be analytically calculated based on the position of the boundary, the shape of the particle and the properties of the particle-matrix interface and the grain boundary [81, 96, 75, 110]. Figure 2.3 shows schematic representations of a grain boundary pinned by respectively a circular and a spherical particle. In two dimensions, the particle-matrix interface and the grain boundary intersect at two distinct points, which is indicated on Fig. 2.3(a) by two dots. In three dimensions, the particle-matrix interface and the grain boundary intersect along a circular line with length $2\pi r \cos\beta$, indicated by the dotted line on Fig. 2.3(b). In both cases, the grain boundary obtains a dimple shape in order to balance the grain boundary tension σ_{gb} and interfacial tensions $\sigma^1_{p/m}$ and $\sigma^2_{p/m}$ of the particle-matrix interface. The contact angle θ depends on the proportions between $\sigma^1_{p/m}$, $\sigma^2_{p/m}$, and σ_{gb}, and is constant during the passage of the grain boundary. The angle β increases while the grain boundary moves from left to right. The grain boundary breaks free from the particle for $\beta = \theta$ in two dimensions and for $\beta = \theta/2$ in three dimensions. Figure 2.4 illustrates grain boundaries pinned in a Fe-0.2%C-0.02%P alloy by Ce_2O_3 and CeS inclusions. The dimple shape mentioned above can be recognised on Fig. 2.4(a).

Computational studies of Zener pinning

The calculation of the total pinning force of a distribution of multiple particles is more complex. The number of particles that lie at a grain boundary, as well as the geometry of the grain boundary at and near each grain boundary-particle intersection, has to be known. This appears to be extremely difficult to describe

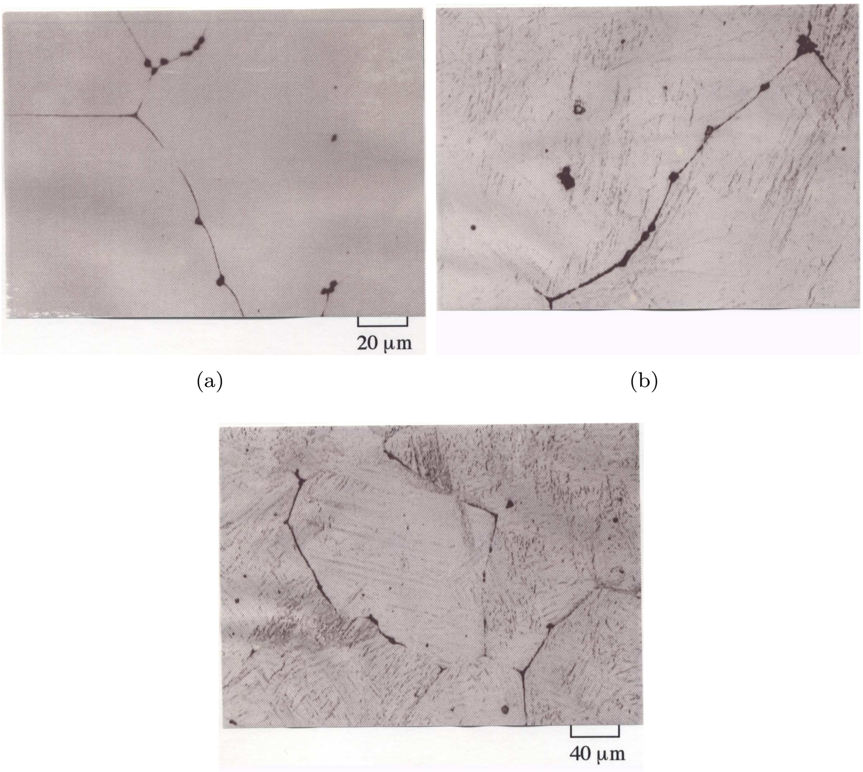


(a) 2D: $F_Z^{2\mathrm{D}} = 2\sigma_{\mathrm{gb}}\cos(\theta - \beta)$ (b) 3D: $F_Z^{3\mathrm{D}} = 2\pi r\sigma_{\mathrm{gb}}\cos\beta\cos(\theta - \beta)$

Figure 2.3: Schematic representations of a grain boundary pinned by (a) a circular particle and (b) a spherical particle [88].

analytically. In this respect, computer simulations turn out to be helpful. They are a practical tool, not only to determine the number and the geometry of boundary-particle intersections, but also to study the role of different characteristics of the second-phase particles separately. Together with existing analytical theories and experimental findings, they provide valuable insights.

The pinning effect of second-phase particles has mostly been studied by two-dimensional computer simulations using Monte Carlo Potts models [41, 118, 83], front-tracking-type models [109, 138, 29] and phase field models [37, 86, 87, 122, 88, 129, 22, 5]. It is found that for two-dimensional systems, relation (2.4) is obeyed, with $b = 0.5$, and most particles are in contact with a grain boundary in the pinned microstructure. Only few studies consider three-dimensional systems. Simulations for three-dimensional systems [83, 122, 88] show that the fraction of particles in contact with a grain boundary is significantly lower than in two-dimensional systems. Furthermore, fitting of relation (2.4) to the results obtained from three-dimensional simulations gives values for b and K that are very different from those obtained for two-dimensional simulations. In [83], $b = 1.02$ and $K = 0.728$ are obtained; in [29], the value $b = 1.0$ is extracted. In contrast, in the work of [122], the parameter values $b = 0.870$ and $K = 1.42$ are found. The study in [88] shows that even for columnar grain structures, the pinning effect is very different from that observed in two-dimensional simulations. In two dimensions, the grain boundaries are basically lines interacting with particles, which split into two different boundary segments when they meet a particle. In contrast, in three dimensions, the grain boundaries are surfaces that remain a single entity when meeting a particle. In order to balance the interfacial tensions at



(a) (b)

(c)

Figure 2.4: Grain boundaries pinned in a Fe-0.2%C-0.02%P alloy by Ce_2O_3 and CeS inclusions: (a) with Ce_2O_3 quenched at 1673K, [ppm insol. Ce] = 680, [ppm T.O] = 140; (b) with Ce_2O_3 held at 1673 K for 60 min, [ppm insol. Ce] = 607, [ppm T.O] = 104; and (c) with CeS quenched at 1673 K, [ppm insol. Ce] = 780, [ppm T.S] = 260 [47].

a particle-boundary intersection, the grain boundary assumes a dimple shape. The extra curvature thus created contributes to the driving force of grain growth [58]. Furthermore, in two dimensions, the pinning force of one particle is maximal when the grain boundary meets the particle at an angle which is twice as large as is the case in three dimensions. Therefore, the pinning effect in two-dimensional systems is in general much stronger than in three-dimensional systems [88]. It is thus important that predictive computer models reflect the three-dimensional nature of Zener pinning for bulk material systems as well as for thin films.

The effect of particle shape

Although the values obtained for the coefficients in the Zener relation in the different simulation studies seem to vary considerably, it was shown in [122] and [129] that the limiting mean grain sizes obtained with different simulation techniques, in fact, agree quite well. Experimentally determined limiting mean grain sizes are however in general much smaller than those predicted by simulations. Based on experimental data, it was also concluded that the Zener relation should have different parameter values for respectively low and high volume fractions [55, 81]. Such a transition between two regimes has not been observed in computer simulations. These discrepancies between experimental and theoretical results indicate that one or several essential aspects were not considered in previous computer simulations. A possible hypothesis is the effect of particle shape. Except for [22], all the above mentioned studies only performed simulations of grain growth in the presence of spherical particles. In reality, however, particles are generally not spherical. Experimental microstructures show that, even in the case where the particle-matrix interface is incoherent and has properties that are independent of orientation, the particle shape deviates from spherical because of inhomogeneities in the surrounding matrix during formation of the particle [121]. For example, particles formed on a former boundary are lance-shaped, the solute flux for particles formed near another particle or a grain boundary is not spherically symmetric, particles may change shape in an anisotropic way during deformation processes. It has been shown that the pinning force of a single particle is strongly dependent on the geometry at the grain boundary-particle section and the particle shape [96, 75, 83, 49]. The effect of particle shape seems to be even far more important than the effect of the anisotropy of the particle-matrix interfacial energy [96, 75].

According to [114], the maximal pinning force F_Z of a spheroid particle is, for the case where the boundary intersects the particle perpendicular to its major axis,

$$F_Z = F_Z^S \frac{2}{(1+r_a)r_a^{1/3}} \tag{2.5}$$

and, for the case where the boundary intersects the particle along a plane containing the major axis,

$$F_Z = \frac{F_Z^S}{\pi} \frac{(1+2.14\ r_a)}{r_a^{1/3}}, \qquad r_a \geq 1, \tag{2.6}$$

where r_a is the aspect ratio of the particle and F_Z^S the drag of spherical particle of equal volume. The last equation shows that if the boundary intersects the particle along a plane containing the major axis, its pinning force is considerably larger than that of a spherical particle with equal volume, even for small aspect ratios. If the boundary intersects the particle perpendicular to the major axis, the pinning force of the particle is however much smaller than that of a spherical particle. As a consequence, the effect of particle shape on the overall pinning effect of a particle distribution depends on the possibility that the boundary-particle intersection contains the major axis. In [75], it was analytically calculated that within well-defined conditions, particle dispersions of mono-orientation can be very effective in pinning migrating boundaries. This was confirmed by the results of two-dimensional phase field simulations in [22], which show that a dispersion of mono-oriented ellipse-shaped particles with a high aspect ratio is more effective in pinning than a dispersion of circular particles of the same size.

Based on the work of [114, 75], it is generally assumed that dispersions of ellipsoid particles have a stronger pinning effect than dispersion of spherical particles, although to our knowledge, the effect of random dispersions of ellipsoid particles on grain growth has not been verified by three-dimensional mesoscale simulations yet, mainly because of computational limitations. To gain more insight in the dependence of the pinning force of a particle distribution on the shape of the particles, we performed three-dimensional phase field simulations of grain growth in systems with spheroid particles for different aspect ratios and volume fractions of the particles [131]. They are described and analysed in Chapter 7.

2.2.3 Anisotropic grain boundary properties

Anisotropic grain boundary energy formulation

During grain growth, grains with favourable properties grow at the expense of other grains, which results in an increase of the mean grain size. In particular, the evolution of the grain structure is influenced by the misorientation between the crystallographic orientations of neighbouring grains and the inclination of the grain boundaries with respect to the reference lattice. Grain boundary characteristics such as boundary mobility and boundary energy strongly depend on these properties. A predictive model for realistic evolution of polycrystalline materials should therefore include orientation dependent microstructural properties and interactions to be able to study this interplay of texture and grain growth [31, 57, 67, 7, 8, 15].

In this thesis, we distinguish one specific grain boundary type from all other boundary types, namely subgrain boundaries. According to [13], a subgrain boundary is formed by the network of discrete dislocations at the interface between two joined crystals of the same kind, but slightly rotated with respect to one another. A subgrain boundary is defined by:

- the orientation of a rotation vector $\boldsymbol{\theta}$,
- the value of a rotation angle θ and

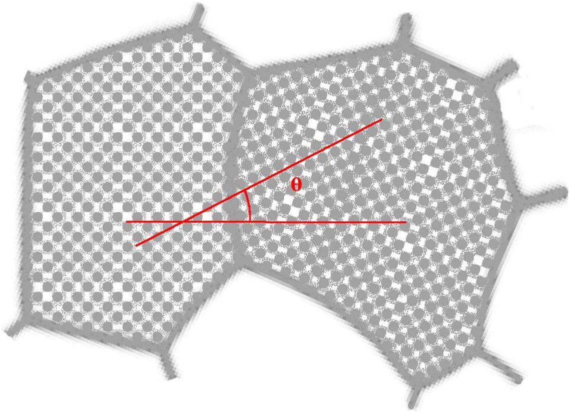


Figure 2.5: Cross-section of a microstructure with fibre texture, revealing the schematic representation of a tilt boundary. The rotation angle θ, which indicates the misorientation between the crystallographic orientations of two neighbouring grains, is indicated.

- the position of the boundary within the crystal.

In a polycrystalline microstructure with fibre texture, the crystallographic orientations of the constituting grains are nearly identical in one direction, a chosen axis, and random in the plane perpendicular to this axis. Consequently, the latter axis coincides with the rotation vector $\boldsymbol{\theta}$ of the occurring subgrain boundaries. Assuming that a subgrain boundary is planar, the orientation of its boundary plane with respect to the rotation axis can be perpendicular, in arbitrary orientation, or parallel. Accordingly, the subgrain boundary is respectively called a pure twist boundary, a partial twist boundary, or a tilt boundary. Figure 2.5 shows the cross-section of a microstructure with fibre texture, revealing the schematic representation of a tilt boundary. The rotation angle θ, which measures the misorientation between the crystallographic orientations of two neighbouring grains, is indicated.

In [13], the energy of a subgrain boundary is calculated for the case of a tilt boundary, essentially following the work of Read and Shockley [108, 107]. The dependence of the boundary energy on the tilt angle θ is derived as:

$$\sigma_{\mathrm{gb}}(\theta) = \sigma_0 \theta (A - \ln \theta), \tag{2.7}$$

with σ_0 and A constant values. This equation is valid for all types of subgrain boundaries if the constants σ_0 and A are modified correspondingly. The maximal value σ_m of this boundary energy formulation is attained for the angle

$$\theta_m = \exp(A - 1), \tag{2.8}$$

and is calculated as

$$\sigma_m = \sigma_0 \exp(A - 1). \tag{2.9}$$

Dividing equation (2.7) by σ_m, equation (2.7) thus becomes

$$\frac{\sigma_{\mathrm{gb}}(\theta)}{\sigma_m} = \frac{\theta}{\theta_m}\left(1 - \ln\left(\frac{\theta}{\theta_m}\right)\right). \tag{2.10}$$

This formulation of the boundary energy, also called the Read-Shockley dependence, is no longer valid for large angles θ, when the boundary becomes a high-angle boundary [13].

Computational studies on the effect of anisotropic boundary properties

Several models exist to simulate the evolution of anisotropic polycrystalline systems: anisotropic grain growth has been studied with Monte Carlo Potts [101, 56, 127, 50, 67], front-tracking [44, 69] and phase field models [63, 127, 78, 124, 90]. In most of these works, two-dimensional simulations are performed [101, 56, 63, 127, 78, 69], although some also include three-dimensional results [44, 67, 124], and in [101, 56, 50, 78], a three-dimensional characterisation of the crystallographic orientation is applied. Most studies start from randomly oriented microstructures [101, 127, 50, 44, 69], although some include a specific initial texture [67]. For example in [56], it is found that grain growth kinetics are significantly slower for a strongly textured structure than for a randomly textured structure, where kinetics are recovered that are consistent with isotropic grain growth. Based on simulation results of microstructures containing two texture components, in [124], grain growth kinetics are divided into different stages of fast and slow growth of one of the components. In [78], a fraction of cube texture (resp. 12.5% and 27%) is included in an otherwise randomly textured microstructure. In the subsequent simulations, the fraction of textured grains increases or decreases continuously in time, depending on the interplay between grain boundary energy and mobility. The anisotropy of the grain boundary characteristics seems to slow down grain growth kinetics [127, 67], while in other studies, normal grain growth kinetics are recovered [50].

When studying the influence of anisotropic properties, the employed models include anisotropic boundary energy [101, 56, 50, 69, 90] or anisotropic mobility [124], or both [127, 63, 78, 67, 44]. It is generally found that the effect of energy anisotropy on grain growth is much larger than the effect of mobility anisotropy [127, 63, 78, 44]. In some studies, the applied boundary energy function and mobility function depend on the misorientation between neighbouring grains solely [101, 56, 127, 50, 78, 67, 124, 90], while in other studies, they depend on the inclination of the grain boundaries as well [63, 44, 69]. For the anisotropic boundary energy function, most [63, 127, 56, 50, 78, 67, 90] assume the Read-Shockley dependence (2.10) for low-angle boundaries:

$$\sigma_{\mathrm{gb}}(\theta) = \sigma_m \frac{\theta}{\theta_m} \left(1 - \ln \left(\frac{\theta}{\theta_m} \right) \right), \qquad \theta < \theta_m, \tag{2.11}$$

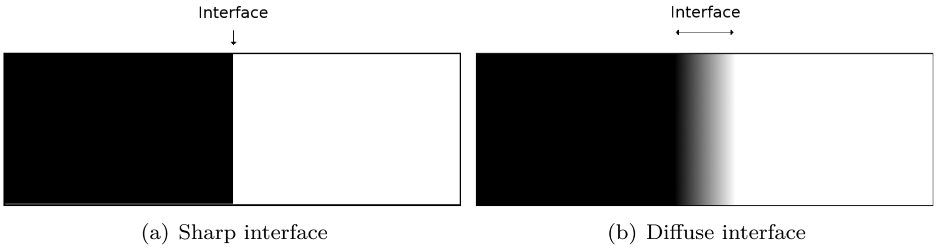
with θ the smallest rotation angle between the orientations of two neighbouring grains, θ_m a threshold angle to distinguish the low angles from the high angles and σ_m the boundary energy associated with misorientation θ_m. In [56, 63, 78, 67], a constant boundary energy σ_m is assumed for the higher-angle boundaries. In [127], the misorientation dependence of grain boundary energy and mobility for higher-angle boundaries is derived from molecular dynamics simulations, including extra low-energy boundaries at specific high-angle misorientations. The latter are also included in [101] and [50], where a three-dimensional crystallography is employed. In all mentioned studies, the fraction of low-angle boundaries is found to increase with time. While [101] also reports an increase of the length fraction for all low energy misorientations, in [50], this increase is only observed for low-angle boundaries, possibly because high-angle boundaries with low energy are more difficult to form in their model. In [127], a two-dimensional crystallography is employed, and all lower-energy boundary fractions are seen to increase with time.

2.3 Different models for grain growth

2.3.1 On grain growth modelling

A polycrystalline microstructure consists of several grains. A model of such a microstructure thus has to be able to distinguish between different domains, each corresponding to a grain. Ideally, the number of possible crystallographic orientations in a grain growth model should equal or exceed the total number of grains, as in reality the number of possible orientations is infinite. The interfaces between the grain domains represent the grain boundaries.

With respect to the representation of the grain boundaries, there exist two types of approaches to construct multi-domain models. The first approach is based on a sharp interface description, in which properties are discontinuous at the interfaces, as is illustrated in Fig. 2.6(a). The second approach employs a diffuse interface description in which properties vary continuously within a narrow interface region, as in Fig. 2.6(b). An example of the first approach are Monte Carlo Potts models, which are surveyed in Section 2.3.2. In simulations with this type of models, the grain domains evolve in time, while the grain boundaries are implicitly defined as discontinuities and thus follow grain evolution. Another example of sharp interface models are front-tracking models, described in Section 2.3.3. This type of



(a) Sharp interface

(b) Diffuse interface

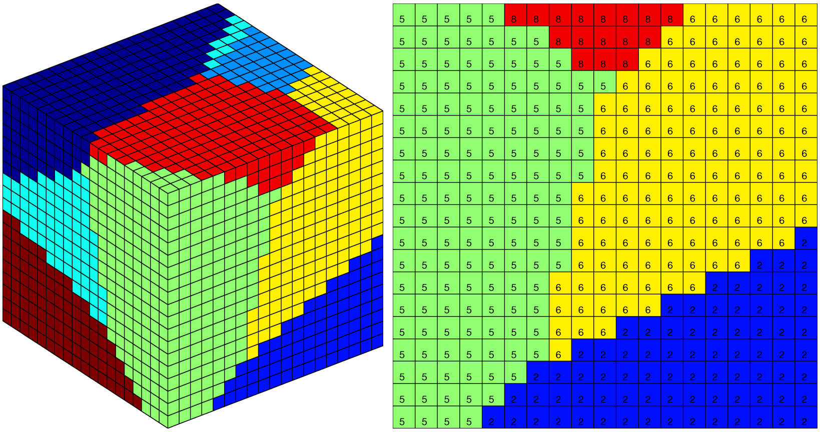
Figure 2.6: There exist two types of models: (a) sharp interface models, with properties that are discontinuous at the interfaces, and (b) diffuse interface models, with properties that vary continuously within a narrow interface region.

models describes the grain boundaries with vertices, edges and faces, and explicitly tracks their evolution. Grains are then defined as the domains delineated by the moving boundaries. Phase field models follow the second approach of modelling: they represent interfaces in a diffuse way. This approach allows to simulate complex morphologies and morphological changes without the need for any prior assumptions on shape. Moreover, the interfaces do not have to be explicitly tracked. Simulations with the three models mentioned above are found to be in good agreement [127, 49].

Another type of models that has been applied to grain growth is level set models [12, 34]. In the level set formulation, grain boundaries are defined as zero-level sets of characteristic functions or signed distance functions that are evolved by solving partial differential equations or by convolution. Phase field models are related to level set models in the sense that they represent grain boundaries as level sets of the phase field variables. As phase field models, level set models naturally handle the difficulties associated with topological changes [34]. However, according to [34], level set models require a lower resolution than phase field models to obtain the same accuracy. So far, little research has been performed on the application of level set models to grain growth simulation. The same is true for cellular automata models. An introductory review of the latter type of models to grain boundary movement in polycrystalline materials can be found in [61]. We will not go into further detail on both types of models.

2.3.2 Monte Carlo Potts models

A Monte Carlo Potts model maps a continuum polycrystalline microstructure to a discrete lattice, which can be rectangular [3, 57, 140] or triangular [56, 50, 67]. Each lattice site represents a Monte Carlo unit (MCU). To initialise the lattice, which contains N MCUs, an integer value corresponding to a crystallographic



5	5	5	5	5	8	8	8	8	8	8	8	8	6	6	6	6	6	6
5	5	5	5	5	5	5	8	8	8	8	8	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	8	8	8	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	2
5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	2	2	2
5	5	5	5	5	5	5	6	6	6	6	6	6	6	2	2	2	2	2
5	5	5	5	5	5	5	6	6	6	6	6	2	2	2	2	2	2	2
5	5	5	5	5	5	5	6	6	6	2	2	2	2	2	2	2	2	2
5	5	5	5	5	5	5	6	2	2	2	2	2	2	2	2	2	2	2
5	5	5	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2
5	5	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2
5	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Figure 2.7: Microstructure on a three-dimensional rectangular lattice. Each lattice site is assigned an integer value corresponding to a grain orientation O_i.

orientation O_i is assigned to every lattice site. This is illustrated for a rectangular grid in Fig. 2.7. All sites within a grain have the same orientation and grain boundaries are represented by interfaces between neighbouring sites of different orientation. A large number p of different grain orientations is used to avoid grain growth by coalescence between neighbouring grains with the same orientation, instead of by grain boundary movement. This effect would result in unphysical grain shapes and incorrect growth kinetics, and is therefore undesired.

The energy of a microstructure is defined in terms of the lattice site energy, which depends on the shell of nearest neighbours of a particular lattice site. The energy $E_{s,i}$ at lattice site s with orientation O_i is computed as

$$E_{s,i} = \sigma_{\mathrm{gb}} \sum_{j=1}^{n} (1 - \delta_{ij}), \tag{2.12}$$

where σ_{gb} is the grain boundary energy, which is constant for isotropic grain growth. The summation in (2.12) is taken over all n sites in the neighbour shell of site s. The study of [67] shows that this neighbour shell has to include a sufficient number of nearest lattice sites to obtain correct grain growth kinetics. The parameter δ_{ij} is the Kronecker delta function, i.e.

$$\delta_{ij} = \begin{cases} 1 & \text{if } O_i = O_j, \\ 0 & \text{if } O_i \neq O_j, \end{cases} \tag{2.13}$$

and O_j the orientation of the jth neighbouring lattice site. Each pair of nearest neighbours thus only contributes an energy amount of magnitude σ_{gb} to the system energy if they do not have the same orientation.

Simulation with the Monte Carlo Potts model relies on repeated random sampling of crystallographic orientations. The smallest time unit of a Monte Carlo Potts simulation is one Monte Carlo step (MCS) and consists of N reorientation attempts. Each reorientation attempt consists of several steps. First, one MCU is chosen according to a specified probability distribution. Second, a new orientation O_j is selected from the set of neighbour orientations. This means that the microstructure can only change through grain boundary motion and that nucleation of new grains is not allowed. The new state of the microstructure now differs from the old state by the change of a single MCU. Third, the difference in energy $\Delta E = E_{s,j} - E_{s,i}$ between the new state j and the old state i is calculated. If the new state has a lower energy than the old state, the change of orientation is accepted with probability P_0. If not, the change of orientation is accepted with a probability of

$$P = P_0 \exp(-\frac{\Delta E}{k_B T}), \tag{2.14}$$

where $P_0 = 1$ for isotropic grain growth. The denominator $k_B T$, with k_B Boltzmann's constant and T the temperature, is an energy term defining the thermal fluctuation of the simulation. In practice, it determines the amount of noise present in the system. Algorithm 2.1 summarises a simulation with the Monte Carlo Potts model.

Algorithm 2.1: Monte Carlo Potts algorithm

```
Generate initial state by assigning an orientation $O_i$ to each lattice site;
for $t = t_1, t_2, \dots, t_{end}$ do
    for $s = 1, 2, \dots, N$ do
        Select MCU according to specified distribution;
        Select new orientation $O_j$ from the set of neighbour orientations;
        Compute $\Delta E = E_{s,j} - E_{s,i}$ with Formula (2.12);
        if $\Delta E \leq 0$ then
            Accept change from state $i$ to state $j$ with probability $P_0$;
        else
            Accept change from state $i$ to state $j$ with probability
            $P = P_0 \exp(-\frac{\Delta E}{k_B T})$;
        end
    end
end
```

In [3], the Monte Carlo Potts model is extended to simulate grain growth in the presence of immobile second-phase particles. During initialisation of the lattice, N_Φ particles are brought into the microstructure by selecting lattice sites according to a uniform distribution. The particles sites are assigned an additional orientation O_{p+1}. The remaining $(N - N_\Phi)$ lattice sites are filled by distributing the other p orientations. The particle-matrix interfacial energy is set to the same value as the grain boundary energy. During a simulation, no attempt is made to reorient the particle sites. Also, they are not allowed to move. One MCS now consists of $(N - N_\Phi)$ reorientation attempts. The model is further modified in [56, 57, 50, 15] to include anisotropic properties. A common factor of these modifications is that P_0 is rewritten such that a reorientation is accepted with a probability proportional to the anisotropic boundary properties.

Important parameters of the Monte Carlo Potts model are the neighbour shell in calculation (2.12) of the lattice site energy and the lattice type, which can be triangular or rectangular. Naturally, the latter parameter is related to the former, since the lattice geometry determines the shape of the neighbour shell. In the case of three-dimensional simulation, the type of stacking of the two-dimensional lattice planes into a three-dimensional lattice is important as well. In [67] for example, a study is made of the effect of the stacking mode and the number of nearest neighbour lattice sites included in the calculation of the system energy on the simulation of normal grain growth.

Simulations with the Monte Carlo Potts model are sensitive to lattice effects: the grain boundaries mimic the underlying lattice geometry and grow faceted. In the case of simulation on a rectangular lattice, grain boundaries tend to meet at 90° or 180°, which results in rectangular grain shapes. On triangular lattices, grain growth is inhibited by lattice pinning due to the formation of triple junctions where three boundaries meet at 120° [56]. Since these lattice effects are non-physical, it is necessary to eliminate them from grain growth simulations. Two possible solutions are an increase of the neighbour sampling in the lattice site energy calculation (2.12), and an action on the simulation temperature T, which alters the transition probability function (2.14). The latter approach thus introduces noise into the system [56]. Although these interventions generally have little effect on the microstructural evolution, it is known that choosing an inappropriate value of T can lead to undesirable effects such as lattice pinning in the case of a low temperature T, or disordering in the case of a high T value [46].

2.3.3 Front-tracking models

Front-tracking models explicitly describe a three-dimensional microstructure by a network of area segments constructed from a set of vertices and edges. The vertices are points located in Cartesian space and the edges are lines connecting the vertices. When edges are connected to one another, a surface is produced, which represents a surface of a grain boundary. Figure 2.8 shows the same microstructure as depicted on Fig. 2.7, but now from the viewpoint of a front-tracking model. The colouring of the former illustration is similar to the latter to indicate the correspondences between the figures. In two dimensions, the representation of the grain boundaries in a front-tracking model is limited to vertices and edges.

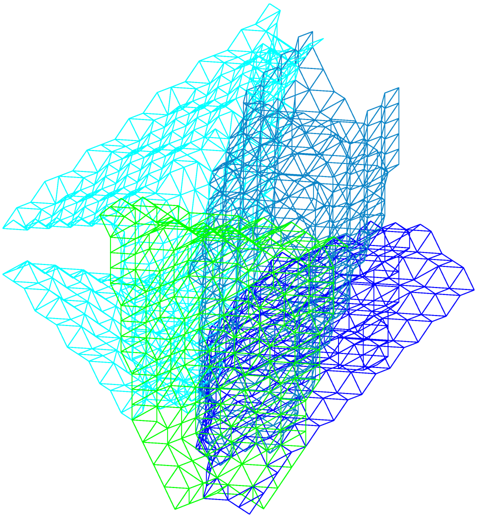


Figure 2.8: Front-tracking models describe a microstructure by a network of area segments constructed from a set of vertices and edges.

All front-tracking models follow a similar approach: the grain boundary movement is curvature-driven, and subject to spatial or geometrical constraints. The velocities of the moving vertices are calculated by minimising a functional that depends on the local geometry and the properties of the grain boundaries. Examples of front-tracking models of grain growth are described in [44, 69] and [49].

We describe one front-tracking model in more detail, namely the two-dimensional model presented in [69]. The data structure of the latter model consists of grain boundaries, triple junctions, and grains, and is managed using standard linking lists. A grain boundary $\Gamma^k(t)$ is defined by a set of nodal points as

$$\Gamma^k(t) = \{x_j^k(t) : j = 1, \ldots, N_k(t)\}, \quad k = 1, \ldots, K, \tag{2.15}$$

with K the number of grain boundaries. Grain boundaries are thus approximated using linear elements and uniformly discretised with a distance between neighbouring nodal points that satisfies

$$\frac{3}{4}h < |x_j^k - x_{j-1}^k| < \frac{3}{2}h, \quad j = 2, \ldots, N_k(t). \tag{2.16}$$

The global mesh size h is defined at the beginning of a simulation and remains constant through a simulation. The number of nodal points $N_k(t)$ for each grain boundary changes during a simulation, as the nodal points are redistributed in order to keep the distance between neighbouring nodal points in the range defined above. The minimal number of nodal points for a grain boundary is three, which guarantees the presence of at least one interior nodal point per grain boundary.

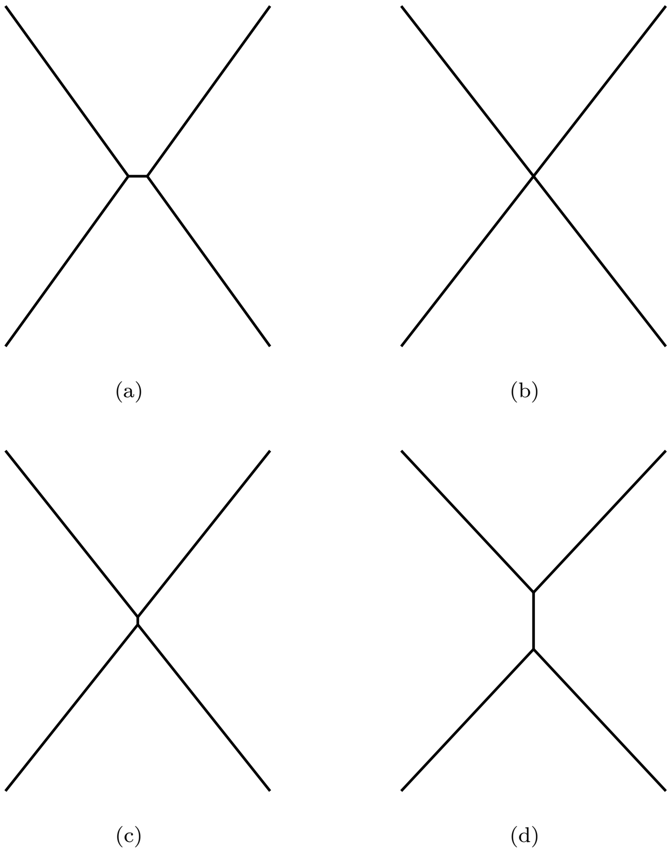
During every time step, first, the grain boundaries are evolved according to the kinetic equation describing grain movement, and second, the new positions of the triple points are updated through an iterative process such that a force balance equation is obeyed. As grain growth proceeds, two types of critical events occur. When the length of a grain boundary falls below a certain threshold value, the grain boundary is removed and the two triple junctions previously delimiting the boundary become one quadruple junction. Since triple junctions typically are the only stable junctions, the newly formed quadruple junction is instantly split into two new triple junctions, connected by an infinitesimally small new grain boundary. The latter grain boundary is most likely to grow during the next time step. This process is called grain boundary flipping and is illustrated on Fig. 2.9. It is designed to preserve the energy decrease of the microstructural evolution. The second type of critical event occurs when small grains shrink below a certain size. This event triggers a process called grain disappearance. As a small grain shrinks, it eventually becomes a junction at which multiple grain boundaries meet. Again, this multiple junction is instantly split into multiple triple junctions.

When implementing a front-tracking model, care has to be taken to the construction of a correct and accurate discretisation scheme and to the treatment of critical events [69]. Critical events are significant changes in the grain boundary network, for example when a grain boundary or an entire grain disappears as described above. To implement these changes and the responses to these changes, assumptions on shape evolution have to be made in advance, which is far from trivial, especially in three dimensions. An advantage of the front-tracking approach is the reduced dimensionality of the data structure, which permits the simulation of large-scale systems.

2.3.4 Phase field models

A phase field model describes a microstructure with a set of phase field variables that are continuous in space and time. These phase field variables represent microstructural properties such as the concentration of one of the components, and the structure or the crystallographic orientation of a phase. When the phase field variables represent structural properties, they are also called order parameters. The spatial and temporal evolution of the phase field variables is governed by a set of coupled partial differential equations. Based on the values assumed by the phase field variables, a microstructure can be decomposed into several domains, separated by diffuse interfaces. The values of the phase field variables are constant within a domain, but vary continuously across the interfaces.

Phase field modelling has shown to be a versatile tool for simulating microstructural evolution phenomena, such as solidification [136, 99, 14, 30, 103], precipitation [19, 137] and grain growth [25, 120, 35, 71, 74, 78]. It allows to predict



(a)

(b)

(c)

(d)

Figure 2.9: When short grain boundaries shrink below a certain threshold value, a process called grain boundary flipping is triggered: (a) the grain boundary length falls below the critical threshold value, (b) the boundary is removed and the previously delimiting triple junctions are joined into one quadruple junction, (c) the quadruple junction is split and a new grain boundary is created with an orientation orthogonal to the disappeared boundary, and (d) the new grain boundary grows.

the evolution of complex morphologies without the need of prior assumptions on shape, while considering different thermodynamic driving forces, such as interfacial energy, bulk energy and elastic energy, and different transport processes, such as heat and mass diffusion.

In the next two sections, we will describe two phase field models for grain growth in detail, namely the multi-phase field model of [120] and the continuum field model presented in [25] and extended in [86, 89, 90]. These models are the main interest of this thesis. Both models describe a polycrystalline microstructure with a large set of phase field variables, with one phase field variable corresponding to each crystallographic orientation. The most obvious difference between these models is that in the multi-phase field model, the phase field variables are interpreted as fractions of orientation. The model therefore imposes the constraint that these phase field variables have to sum up to one, while in the continuum field model, there is no such restriction. There exists no straightforward relation to transform one of these models into the other. However, it is possible to derive relationships between the model parameters for given grain boundary properties and grain growth kinetics [92]. After a detailed comparison of the multi-phase field model and the continuum field model in [92], it is concluded that both models essentially give the same results, except for differences near small shrinking grains, which are most often local and temporary for large grain structures.

2.4 Multi-phase field model for grain growth

In [120], a multi-phase field model is proposed which models a polycrystalline microstructure by a vector-valued order parameter $\boldsymbol{\phi}$ with p components:

$$\boldsymbol{\phi}(\mathbf{r},t) = (\phi_1(\mathbf{r},t), \phi_2(\mathbf{r},t), \ldots, \phi_p(\mathbf{r},t))\,. \tag{2.17}$$

Each component ϕ_i is a phase field variable and represents a different crystallographic orientation. The values of the p components are continuous in space and time. Inside a grain, one component ϕ_i takes the value 1, while the other phase field variables assume the value zero. Across the grain boundaries, the component values vary continuously from their equilibrium value within the grain to their equilibrium value in the neighbouring grains. In accordance with [120], the condition is imposed that the components ϕ_i must lie in the Gibbs-simplex $\mathcal{G}$, which is defined as

$$\mathcal{G} = \{\boldsymbol{\phi} \in \mathbb{R}^p : \phi_i \geq 0,\ \sum_{i=1}^{p} \phi_i = 1\}. \tag{2.18}$$

The free energy $\tilde{F}$ of the microstructure is a function of the order parameter $\boldsymbol{\phi}$ and is described by

$$\tilde{F}(\boldsymbol{\phi}) = \int_V \left(\epsilon a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi}) + \frac{1}{\epsilon} w(\boldsymbol{\phi}) \right) dV, \tag{2.19}$$

with $a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi})$ the gradient free energy density and $w(\boldsymbol{\phi})$ the bulk potential. The parameter ϵ is a measure for the width of the diffuse interfaces. The term $a(\phi, \nabla\phi)$ models the interfacial contribution to the free energy and is given by:

$$a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi}) = s \sum_{\substack{i,j=1\\ i<j}}^{p} \sigma_{ij} \|\phi_i \nabla\phi_j - \phi_j \nabla\phi_i\|^2, \tag{2.20}$$

with σ_{ij} the surface energy density of the corresponding i/j interface. A scaling parameter s is introduced for technical reasons: it allows to formulate the phase field model on the unit square [132]. Note that for $p = 2$, formulation (2.20) reduces to $a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi}) = s\sigma_{12}\|\nabla\phi_1\|^2$, taking into account that $\phi_1 + \phi_2 = 1$.

For the bulk potential $w(\boldsymbol{\phi})$, we describe two possible choices [42]. The first one is the multi-obstacle potential, which is formulated as

$$w_{\text{obst}}(\boldsymbol{\phi}) = \frac{16}{\pi^2} \sum_{i<j} \sigma_{ij}\phi_i\phi_j + \sum_{i<j<k} \sigma_{ijk}\phi_i\phi_j\phi_k \quad \text{for} \quad \boldsymbol{\phi} \in \mathcal{G}, \tag{2.21}$$

where $w_{\text{obst}}(\boldsymbol{\phi})$ is defined to be infinite whenever $\boldsymbol{\phi}$ is not in the Gibbs-simplex $\mathcal{G}$. The second possible choice is the multi-well potential:

$$w_{\text{well}}(\boldsymbol{\phi}) = 9 \sum_{i<j} \sigma_{ij}\phi_i^2\phi_j^2 + \sum_{i<j<k} \sigma_{ijk}\phi_i^2\phi_j^2\phi_k^2. \tag{2.22}$$

To ensure that the sum constraint in (2.18) is preserved, a Lagrange multiplier function is added to the formulation of the free energy in (2.19). The free energy functional is now described by

$$F(\boldsymbol{\phi}) = \int_V \left(\epsilon a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi}) + \frac{1}{\epsilon} w(\boldsymbol{\phi}) \right) dV + \lambda \left(\sum_{i=1}^{p} \phi_i - 1 \right). \tag{2.23}$$

The evolution of the phase field variables ϕ_i is governed by gradient flow dynamics of the form

$$\omega\epsilon \frac{\partial \phi_i}{\partial t} = -\frac{\delta F(\boldsymbol{\phi})}{\delta \phi_i}, \qquad i = 1, \ldots, p, \tag{2.24}$$

where the parameter ω is related to a physical grain boundary property: $\frac{1}{\omega}$ equals the interfacial mobility μ. The notation $\frac{\delta}{\delta\phi_i}$ is used to denote the variational derivative with respect to ϕ_i and is computed as

$$\frac{\delta}{\delta\phi_i} = \frac{\partial}{\partial\phi_i} - \nabla \cdot \frac{\partial}{\partial(\nabla\phi_i)} \tag{2.25}$$

in accordance with [120]. The second part of the right-hand side of this formulation is the divergence of a vector, which can be written as

$$\nabla \cdot \frac{\partial}{\partial(\nabla\phi_i)} = \nabla \cdot \begin{pmatrix} \frac{\partial}{\partial(\partial\phi_i/\partial x)} \\ \frac{\partial}{\partial(\partial\phi_i/\partial y)} \end{pmatrix}. \tag{2.26}$$

Substitution of (2.20) and (2.21) in the free energy functional (2.23), and elaborating the kinetic equations (2.24) results in:

$$\omega\epsilon\frac{\partial\phi_i}{\partial t} = \epsilon\left(\nabla\cdot\frac{\partial a}{\partial(\nabla\phi_i)} - \frac{\partial a}{\partial\phi_i}\right) - \frac{1}{\epsilon}\frac{\partial w}{\partial\phi_i} - \lambda, \qquad i = 1,\ldots,p. \tag{2.27}$$

Here, we shall not follow the approach suggested in [99], where the function λ is eliminated explicitly. Instead, the system of equations (2.27) is extended with the sum constraint of the Gibbs-simplex (2.18). Hence the model consists of p partial differential equations plus one algebraic equation. The gradient free energy component in (2.27) is derived as follows. For the first component in the right-hand side of (2.27) we find:

$$\begin{aligned}\nabla\cdot\frac{\partial a}{\partial(\nabla\phi_i)} = {} & -2s\sum_{j\neq i}\sigma_{ij}\nabla\phi_j\cdot(\phi_i\nabla\phi_j - \phi_j\nabla\phi_i) \\ & - 2s\sum_{j\neq i}\sigma_{ij}\phi_j(\phi_i\nabla^2\phi_j - \phi_j\nabla^2\phi_i).\end{aligned} \tag{2.28}$$

The second component gives:

$$\frac{\partial a}{\partial\phi_i} = 2s\sum_{j\neq i}\sigma_{ij}\nabla\phi_j\cdot(\phi_i\nabla\phi_j - \phi_j\nabla\phi_i). \tag{2.29}$$

As a result, we have that:

$$\begin{aligned}\nabla\cdot\frac{\partial a}{\partial(\nabla\phi_i)} - \frac{\partial a}{\partial\phi_i} = {} & -4s\sum_{j\neq i}\sigma_{ij}\nabla\phi_j\cdot(\phi_i\nabla\phi_j - \phi_j\nabla\phi_i) \\ & - 2s\sum_{j\neq i}\sigma_{ij}\phi_j(\phi_i\nabla^2\phi_j - \phi_j\nabla^2\phi_i).\end{aligned} \tag{2.30}$$

The equations obtained by filling out (2.30) in (2.27) will be the subject of Chapter 6. There, we will develop a fast multigrid solver for two-dimensional grain growth simulations with multi-phase field model (2.27).

2.5 Continuum field model for grain growth

2.5.1 Continuum field model for isotropic grain growth

The phase field model of Chen and Yang [25] represents the microstructure of a single-phase polycrystalline material by a set of p phase field variables

$$\boldsymbol{\eta}(\mathbf{r},t) = (\eta_1(\mathbf{r},t), \eta_2(\mathbf{r},t), \ldots, \eta_p(\mathbf{r},t)) \,. \tag{2.31}$$

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. Inside a grain, one phase field variable η_i takes a non-zero constant value, 1 or -1, while the other phase field variables assume values close to zero. Across the grain boundaries, all phase field variables vary continuously from their equilibrium value within the grain to their equilibrium value in the neighbouring grains. This is illustrated in Fig. 2.10, where a microstructure is visualised by transforming the sum of the squares of the phase field variables into grey scale.

The spatial and temporal evolution of the phase field variables (2.31) is governed by the equations

$$\frac{\partial \eta_i}{\partial t} = -L_i \frac{\delta F(\boldsymbol{\eta})}{\delta \eta_i}, \qquad i = 1, \ldots, p, \tag{2.32}$$

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

$$F(\boldsymbol{\eta}) = \int_V \left(f_0(\boldsymbol{\eta}) + \sum_{i=1}^{p} \frac{\kappa_i}{2} (\nabla \eta_i)^2 \right) dV, \tag{2.33}$$

with κ_i the gradient energy coefficients and f_0 the homogeneous free energy density. The gradient term in formula (2.33) is always positive and thus penalises the presence of interfaces. The free energy density f_0 is

$$f_0(\boldsymbol{\eta}) = \sum_{i=1}^{p} \left(-\frac{\alpha}{2} \eta_i^2 + \frac{\beta}{4} \eta_i^4 \right) + \gamma \sum_{i=1}^{p} \sum_{j>i}^{p} \eta_i^2 \eta_j^2, \tag{2.34}$$

with α, β, and γ positive constants and $\gamma > \frac{\beta}{2}$, as proposed in [25, 36]. For $\gamma > \frac{\beta}{2}$, f_0 has $2p$ degenerate minima with equal depth at

$$\begin{aligned} (\eta_1, \eta_2, \ldots, \eta_p) = {} & (1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1), \\ & (-1, 0, \ldots, 0), (0, -1, \ldots, 0), \ldots, (0, 0, \ldots, -1). \end{aligned} \tag{2.35}$$

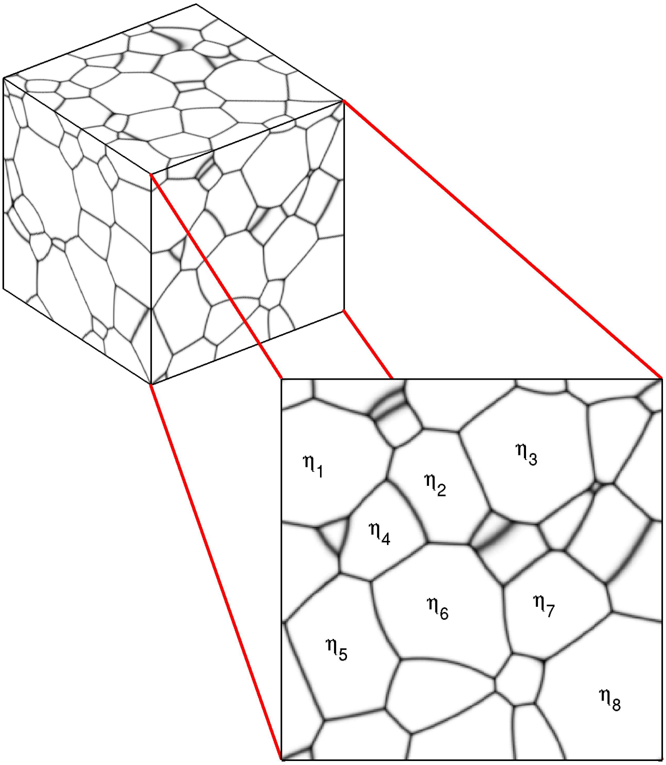


Figure 2.10: The continuum field model represents a single-phase polycrystalline microstructure by a set of phase field variables $(\eta_1(\mathbf{r},t), \eta_2(\mathbf{r},t), \ldots, \eta_p(\mathbf{r},t))$.

As in [85], we assume that the set of values $(\ldots, \eta_i, \ldots) = (\ldots, 1, \ldots)$ and $(\ldots, \eta_i, \ldots) = (\ldots, -1, \ldots)$ represent the same orientation. The $2p$ minima of the free energy density f_0 thus reflect the p orientations a grain can have. For a particular phase field variable η_i, f_0 is minimal for $\eta_i = 1$ or for $\eta_i = -1$ when all other phase field variables equal zero. If one of the other phase field variables equals 1 or -1, f_0 only reaches its minimum when $\eta_i = 0$, because of the cross-term in (2.34). The latter term makes it energetically unfavourable to have two phase field variables different from zero at the same position in the system.

In accordance with Moelans in [85], the parameters α, β and γ in (2.34) are renamed to the same parameter $m = \alpha = \beta = \gamma$. The constraint that $\gamma > \beta/2$ is still fulfilled. The free energy density expression f_0 can then be written as

$$f_0(\boldsymbol{\eta}) = m\tilde{f}_0(\boldsymbol{\eta}) = m\left(\sum_{i=1}^{p}\left(-\frac{\eta_i^2}{2} + \frac{\eta_i^4}{4}\right) + \sum_{i=1}^{p}\sum_{j>i}^{p}\eta_i^2\eta_j^2 + \frac{1}{4}\right). \tag{2.36}$$

The constant term $m/4$ is added so that the homogeneous free energy $f_0(\boldsymbol{\eta})$ equals zero within grains.

Substituting (2.33) into (2.32) and applying formula (2.25) to compute the variational derivative in (2.32) results in a set of reaction-diffusion partial differential equations:

$$\frac{\partial \eta_i}{\partial t} = L_i \left(\kappa_i \nabla^2 \eta_i + m \left(\eta_i^3 + \eta_i - 2\eta_i \sum_{j=1}^{p} \eta_j^2 \right) \right), \qquad i = 1, \ldots, p. \tag{2.37}$$

For isotropic grain boundary energy and mobility, the kinetic coefficients and the gradient energy coefficients as well as the parameter m are constant values, with $\kappa_i = \kappa$ and $L_i = L$.

2.5.2 Continuum field model for grain growth in the presence of second-phase particles

In [86, 87], phase field model (2.37) is extended for the simulation of grain growth in materials containing small incoherent second-phase particles with constant properties. To include such particles in the model, a spatially dependent phase field variable Φ is added. This parameter Φ equals 1 inside a particle and 0 elsewhere, and remains constant in time. The free energy density f_0 of the system is now described by

$$f_0(\boldsymbol{\eta}, \Phi) = m \left(\sum_{i=1}^{p} \left(-\frac{\eta_i^2}{2} + \frac{\eta_i^4}{4} \right) + \sum_{i=1}^{p} \sum_{j>i}^{p} \eta_i^2 \eta_j^2 + \Phi^2 \sum_{i=1}^{p} \eta_i^2 + \frac{1}{4} \right). \tag{2.38}$$

The extra term in the free energy expression forces all phase field variables to be zero inside a particle. For $\Phi = 1$, the free energy density has one minimum, namely at $(\eta_1, \eta_2, \ldots, \eta_p) = (0, 0, \ldots, 0)$. For $\Phi = 0$, the free energy density reduces to expression (2.36) with its $2p$ minima representing the orientations of the grains.

Substituting (2.38) into (2.33) and subsequently into (2.32) results again in a set of reaction-diffusion partial differential equations, this time extended to include the presence of second-phase particles:

$$\frac{\partial \eta_i}{\partial t} = L_i \left(\kappa_i \nabla^2 \eta_i + m \left(\eta_i^3 + \eta_i - 2\eta_i \left(\sum_{j=1}^{p} \eta_j^2 + \Phi^2 \right) \right) \right), \qquad i = 1, \ldots, p. \tag{2.39}$$

In Chapter 5, a sparse bounding box method is developed to perform efficient large-scale simulations with model (2.39). The applicability of the bounding box algorithm is illustrated in Chapter 7, where model (2.39) is used to study the pinning effect of spheroid second-phase particles on grain growth.

2.5.3 Continuum field model for grain growth in a microstructure with anisotropic grain boundary energy

The boundaries that delineate the grains represent misorientation between the crystallographic orientations of neighbouring grains. The evolution of a grain structure is influenced by the misorientation and the inclination of the grain boundaries with respect to the reference lattice. Grain boundary characteristics such as mobility and boundary energy strongly depend on these properties. A predictive model for realistic evolution of polycrystalline materials should therefore include orientation dependent microstructural properties and interactions to be able to study the interplay of texture and grain growth [31, 57, 67, 7, 8, 15].

In [89, 90], a procedure is derived to generate the model parameters of a generalised phase field model for anisotropic grain growth, which is based on the work of [36, 63]. This procedure computes the model parameters to represent the grain boundary energy and mobility of a material for arbitrary misorientation and inclination dependence. Its formulation allows to perform quantitative simulations with uniform stability and accuracy conditions. In this thesis, we will restrict ourselves to misorientation dependence of the boundary energy function.

We reconsider the formulation of the free energy F and introduce misorientation dependence through the gradient energy coefficient κ:

$$F(\boldsymbol{\eta}) = \int_V \left(f_0(\boldsymbol{\eta}) + \frac{\kappa(\Theta)}{2} \sum_{i=1}^{p} (\nabla \eta_i)^2 \right) dV, \tag{2.40}$$

where Θ corresponds to the ensemble of parameters that defines the boundary misorientation (see also Section 2.2.3). The misorientation dependence is brought into the homogeneous free energy f_0 through the parameter $\gamma(\Theta)$:

$$f_0(\boldsymbol{\eta}) = m \left(\sum_{i=1}^{p} \left(-\frac{\eta_i^2}{2} + \frac{\eta_i^4}{4} \right) + \sum_{i=1}^{p} \sum_{j>i}^{p} \gamma(\Theta) \eta_i^2 \eta_j^2 + \frac{1}{4} \right). \tag{2.41}$$

Note that we have previously eliminated the parameter γ (see formula (2.34) and (2.36)). As mentioned, the inclination dependence of the grain boundary properties is ignored. The parameters $\kappa(\Theta)$ and $\gamma(\Theta)$ are defined as

$$\kappa(\Theta) = \frac{\sum_{i=1}^{p} \sum_{j>i}^{p} \kappa_{ij} \eta_i^2 \eta_j^2}{\sum_{i=1}^{p} \sum_{j>i}^{p} \eta_i^2 \eta_j^2} \tag{2.42}$$

and

$$\gamma(\Theta) = \frac{\sum_{i=1}^{p}\sum_{j>i}^{p}\gamma_{ij}\eta_i^2\eta_j^2}{\sum_{i=1}^{p}\sum_{j>i}^{p}\eta_i^2\eta_j^2}. \tag{2.43}$$

Recall that inside a grain, one phase field variable η_i takes the constant value 1, while the other phase field variables assume values close to zero. Across the grain boundaries, the corresponding field variables vary continuously to their equilibrium value in the neighbouring grains. Consequently, the parameters κ and γ respectively equal κ_{ij} and γ_{ij} at the boundary between two neighbouring grains represented by η_i and η_j.

In [89, 90], a procedure is developed that allows to compute the values of κ_{ij} and γ_{ij} in such a way that any given grain boundary energy function can be reproduced, as well as a constant diffuse interface width for uniform accuracy stability. Substituting formulations (2.42) and (2.43) into the free energy F (2.40), and in turn into the kinetic equations (2.32) yields model

$$\frac{\partial\eta_i}{\partial t} = L\left(\kappa(\Theta)\nabla^2\eta_i - m\left(\eta_i^3 - \eta_i + 2\eta_i\sum_{j\neq i}^{p}\gamma_{ij}\eta_j^2\right)\right), \qquad i = 1,\ldots,p. \tag{2.44}$$

To take into account the coupling between the phase field variables through the parameters $\kappa(\Theta)$ and $\gamma(\Theta)$, dedicated features are added to the sparse bounding box algorithm developed in Chapter 5.

In Chapter 8, we present preliminary results obtained from three-dimensional simulations with model (2.44) of grain growth in a microstructure with fibre texture. In this case, the parameter Θ reduces to the misorientation angle θ. Furthermore, a Read-Shockley boundary energy dependence is assumed for the energy of the low-angle boundaries, while a constant, higher boundary energy is attributed to the other grain boundary types.

2.6 Conclusion

In this chapter, we have introduced the mechanisms of grain growth in polycrystalline microstructures, as well as some of its influencing factors. After a short discussion of the different types of models that have been developed to simulate the latter phenomenon, we have described two different phase field models for grain growth, namely the multi-phase field model of [120], and the continuum field model of [25], which was extended to simulate grain growth in the presence of second-phase particles in [86, 87] and to simulate grain growth in a microstructure with anisotropic boundary properties in [89, 90]. Both models describe a polycrystalline microstructure with a large set of phase field variables, with one phase field variable corresponding to each crystallographic orientation. There is no straightforward relation between the two models, but for given system properties, it is possible to derive relationships between the model parameters. Furthermore, both models essentially give the same results [92].

In principle, the number of phase field variables p included in these phase field models of grain growth 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, grain growth can occur by coalescence between neighbouring grains with the same orientation, instead of by boundary migration. This leads to unphysical grain shapes and incorrect growth kinetics. This was verified in two-dimensional phase field simulations of coarsening in [35]: as the number of phase field variables was decreased, the growth rate of the mean grain size increased steadily. Especially in three-dimensional simulations, where grains have on average more neighbours than in two dimensions, a very large number of crystallographic orientations are required to minimise the effect of grain coalescence [74].

For anisotropic materials, it is particularly important that the orientation dependencies of material properties are resolved accurately. Furthermore, 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 to reproduce the shape of grain boundaries at grain boundary-particle 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 significant amounts of computation power as well as data storage. In the next chapter, we will give an overview of the computational techniques that have been applied to overcome these constraints.

Chapter 3

Acceleration techniques for numerical simulation

As discussed in the previous chapter, realistic three-dimensional computer simulations of grain growth with multi-phase field model (2.27) and continuum field models (2.37), (2.39) and (2.44) demand significant amounts of computation power as well as data storage. This chapter gives an overview of several algorithms designed to overcome the computational limitations of the phase field method. Sections 3.1 and 3.2 describe different discretisation schemes that have been applied to phase field models. Subsequently, Section 3.3 discusses acceleration techniques that intervene in the data structure, while Section 3.4 describes techniques that take action on the level of the solver. Another possibility is to use parallel computing, which is briefly discussed in Section 3.5. This chapter ends in Section 3.6 with a summary and conclusive remarks.

3.1 Discretisation in space

Before simulating grain growth with model (2.37) or (2.27), it is important to think of the appropriate spatial discretisation. A finer discretisation in space will increase the resolution in space and augment accuracy. Unfortunately, a finer discretisation also results in a larger system that has to be solved at each time step: there is a larger number of unknowns involved. Also, depending on the discretisation in time, a fine spatial discretisation can impose severe restrictions on the possible time step size. For reasons of clarity, in this section, the spatial discretisation will only be described in one dimension.

3.1.1 Finite differences

We consider a uniform grid $[x_1 \ \dots \ x_N]$ on the interval $[0, l]$ with $\Delta x = \frac{l}{N}$, and the vector of corresponding sample values $[\eta_1 \ \dots \ \eta_N]^{\mathrm{T}}$. The standard second-order finite difference approximation of $\eta_r' = \eta'(x_r)$ is

$$\eta_r' = \frac{\eta_{r+1} - \eta_{r-1}}{2\Delta x}. \tag{3.1}$$

When periodic boundary conditions are assumed, as is the case for phase field model (2.37), equalities $\eta_0 = \eta_N$ and $\eta_1 = \eta_{N+1}$ hold. We can represent this discrete differentiation as a matrix-vector product:

$$\begin{pmatrix} \eta_1' \\ \\ \vdots \\ \\ \eta_N' \end{pmatrix} = \frac{1}{\Delta x} \begin{pmatrix} 0 & \frac{1}{2} & 0 & \dots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} & & & 0 \\ 0 & -\frac{1}{2} & 0 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & & 0 \\ 0 & & & & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \dots & 0 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \\ \vdots \\ \\ \eta_N \end{pmatrix}. \tag{3.2}$$

An alternative way to derive (3.1) is by a succession of local interpolation and differentiation [126]:

Algorithm 3.1: Second-order finite difference approximation of η_r' by local interpolation and differentiation

```
for $r = 1, 2, \dots, N$ do
    Let $p_r$ be the unique polynomial of degree $\leq 2$ with $p_r(x_{r-1}) = \eta_{r-1}$,
    $p_r(x_r) = \eta_r$, and $p_r(x_{r+1}) = \eta_{r+1}$;
    Set $\eta_r' = p_r'(x_r)$;
end
```

This derivation by local interpolation can be generalised to higher orders. The polynomial p_r will then be of higher degree and the bandwidth of the corresponding differentiation matrix will be larger.

The differentiation matrix in equation (3.2) has second-order accuracy. This means that for data η_r obtained by sampling a sufficiently smooth function $\eta(x)$, the corresponding discrete approximations to η_r' will converge at rates $\mathcal{O}((\Delta x)^2)$ as $\Delta x \to 0$.

After second-order finite differentiation of the Laplacian, model (2.37) becomes:

$$\frac{\partial \eta_i}{\partial t} = L\kappa \frac{\eta^n_{i_{r+1}} - 2\eta^n_{i_r} + \eta^n_{i_{r-1}}}{(\Delta x)^2} + Lm \left(\eta^3_{i_r} + \eta_{i_r} - 2\eta_{i_r} \sum_{j=1}^{p} \eta^2_{j_r} \right), \qquad r = 1, \ldots, N, \qquad i = 1, \ldots, p. \tag{3.3}$$

3.1.2 Spectral differences

The order of accuracy of spectral differences is much higher than that of finite differences [126]. This property is called spectral accuracy. The design principle for spectral difference approximation of η'_r is as follows:

Algorithm 3.2: General algorithm for spectral difference approximation of η'_r by interpolation and differentiation

```
Let $p$ be a single function (independent of $r$) such that $p(x_r) = \eta_r$
for $r = 1, \ldots, N$;
Set $\eta'_r = p'(x_r)$;
```

For a periodic domain, the natural choice for the interpolant p is a trigonometric polynomial on an equispaced grid. Before we construct interpolant p, we review the concepts of the Discrete Fourier Transform (DFT).

We regard the basic periodic grid $[x_1 \ \ldots \ x_N]$ on the interval $[0, l]$ as one cycle extracted from an infinite grid with sample values satisfying

$$\eta_{r-mN} = \eta_r, \qquad r = 1, \ldots, N; \quad m \in \mathbb{Z}. \tag{3.4}$$

The Fourier domain is discrete as well as bounded, because waves in physical space must be periodic over the interval $[0, l]$. Only waves $e^{\iota k \frac{2\pi}{l} x}$ with integer wave numbers k have the required period l, with ι the imaginary unit. We limit our attention to $k \in \{-\frac{N}{2} + 1, -\frac{N}{2} + 2, \ldots, \frac{N}{2}\}$, since wave numbers differing by an integer multiple of $N = \frac{l}{\Delta x}$ are indistinguishable on the grid. The DFT of the sample values η_r is now computed with formula

$$\hat{\eta}_k = \Delta x \sum_{r=1}^{N} e^{-\iota k \frac{2\pi}{l} x_r} \eta_r, \qquad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}, \tag{3.5}$$

while the inverse DFT is computed as

$$\eta_r = \frac{1}{l} \sum_{k=-N/2+1}^{N/2} e^{\iota k \frac{2\pi}{l} x_r} \hat{\eta}_k, \qquad r = 1, \ldots, N. \tag{3.6}$$

The DFT of the derivative η'_r, with $r = 1, \ldots, N$, is therefore given by

$$\widehat{\eta'}_k = \iota k \frac{2\pi}{l} \hat{\eta}_k, \qquad k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}. \tag{3.7}$$

Trefethen explains in [126] how the interpolant p in Algorithm 3.2 can be constructed as

$$p(x) = \frac{1}{l} \sum_{k=-N/2}^{N/2}{}' \, e^{\iota k \frac{2\pi}{l} x} \hat{\eta}_k, \tag{3.8}$$

where the prime indicates that the terms $k = \pm N/2$ are multiplied by 1/2 for reasons of symmetry. One of the algorithms developed in [126] to construct an approximation for the derivative $\eta'_r = p'(x_r)$ uses formulas (3.5) and (3.6) as follows:

Algorithm 3.3: Spectral difference approximation of η'_r

```
Given $\eta_r$, compute $\hat{\eta}_k$ with formula (3.5);
Define $\widehat{\eta'}_k = \iota k \frac{2\pi}{l} \hat{\eta}_k$, except $\widehat{\eta'}_{\frac{N}{2}} = 0$ for reasons of symmetry [126];
Compute $\eta'_r$ from $\widehat{\eta'}_k$ with formula (3.6);
```

For higher derivatives, we multiply $\hat{\eta}_k$ by the appropriate power of $\iota k \frac{2\pi}{l}$, taking special care of the $\widehat{\eta'}_{\frac{N}{2}}$ term. In summary, to approximate the νth derivative of the function η on the grid, the following algorithm can be used:

Algorithm 3.4: Spectral difference approximation of $\eta_r^{(\nu)}$

```
Given $\eta_r$, compute $\hat{\eta}_k$ with formula (3.5);
Define $\widehat{\eta^{(\nu)}}_k = (\iota k \frac{2\pi}{l})^\nu \hat{\eta}_k$, with $\widehat{\eta^{(\nu)}}_{\frac{N}{2}} = 0$ if $\nu$ is odd;
Compute $\eta_r^{(\nu)}$ from $\widehat{\eta^{(\nu)}}_k$ with formula (3.6);
```

The computation of the DFT can be accomplished by the Fast Fourier Transform (FFT) [27].

Application of spectral differentiation in space to model (2.37) results in

$$\frac{\partial \eta_{i_r}}{\partial t} = L\kappa \eta_{i_r}^{(2)} + Lm \left(\eta_{i_r}^3 + \eta_{i_r} - 2\eta_{i_r} \sum_{j=1}^{p} \eta_{j_r}^2 \right),$$

$$r = 1, \ldots, N, \qquad i = 1, \ldots, p, \tag{3.9}$$

with $\eta_{i_r}^{(2)}$ computed as described above.

3.1.3 Finite element method

Some authors have made use of the finite element method (FEM) to approximate the solution of a phase field model [98, 104, 105, 106]. To our knowledge, no work has been performed where the FEM was applied to phase field models (2.37) and (2.27) for grain growth. Therefore, we will not go into detail on this discretisation method.

3.2 Discretisation in time

Many authors that simulate grain growth with phase field simulations use first-order, explicit time integration formulas [74, 97, 65, 45, 133], while some apply implicit formulas [24, 129, 33]. Because phase field models (2.37) and (2.27) of grain growth require a lot of computing memory, to our knowledge, no work has been performed on the application of multi-step or multi-stage schemes. This type of schemes relies on the solution values at different time steps or stages. Keeping track of the solution values at these different stages requires additional computing memory. On the other hand, multi-step and multi-stage schemes allow fairly naturally for adaptive time step strategies.

3.2.1 IMEX schemes

For PDE systems such as phase field model (2.37), it is inefficient to use one single time integration formula for the different parts of the system. Applying an implicit formula to the latter model results in a large nonlinear system because of the simultaneous coupling in space and between the phase field variables. Model (2.37) naturally splits up into two parts: a diffusion part and a reaction part. These different parts can be integrated over time with appropriate formulas: the reaction part is suitable for explicit treatment, while the diffusion part requires an implicit treatment. This type of splitting methods is known as IMEX methods, which consist of an appropriate combination of an implicit and explicit method [60]. IMEX methods are no universal cure, but can be very effective in many situations [6]. In [113], the performance of several linear multi-step IMEX schemes is analysed for reaction-diffusion problems in pattern formation.

In the case of our model, the application of an IMEX scheme has many advantages. For example, it allows the diffusion part to be treated by a linear multigrid solver, without the reaction part influencing the solver's convergence properties in a negative way. Also, the decoupling of the different phase field variables clears the way for parallel computing. This possibility is briefly discussed in Section 3.5.

First-order semi-implicit BDF (SBDF1)

The most straightforward IMEX scheme is the first-order semi-implicit BDF method. It consists of the Forward Euler method as the explicit method, and the Backward Euler method as the implicit method. The Backward Euler method is in fact a first-order Backward Differentiation Formula (BDF), hence the name SBDF1 [113]. For clarity, we first simplify model (2.37) to

$$\frac{\partial \eta_i}{\partial t} = D(\eta_i) + R(\boldsymbol{\eta}), \qquad i = 1, \ldots, p, \tag{3.10}$$

with $D(\eta_i)$ the diffusion part, $R(\boldsymbol{\eta})$ the reaction part, and $\boldsymbol{\eta}$ as defined in (2.31).

Application of SBDF1 to equations (3.10) results in

$$\frac{\eta_i^{n+1} - \eta_i^n}{\Delta t} = D(\eta_i^{n+1}) + R(\boldsymbol{\eta}^n), \qquad i = 1, \ldots, p. \tag{3.11}$$

Combining SBDF1 with the finite differences in equations (3.3) results in the following system:

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^n}{\Delta t} = L\kappa \frac{\eta_{i_{r+1}}^{n+1} - 2\eta_{i_r}^{n+1} + \eta_{i_{r-1}}^{n+1}}{(\Delta x)^2} + Lm \left((\eta_{i_r}^n)^3 + \eta_{i_r}^n - 2\eta_{i_r}^n \sum_{j=1}^{p} (\eta_{j_r}^n)^2 \right),$$

$$r = 1, \ldots, N, \qquad i = 1, \ldots, p. \tag{3.12}$$

SBDF1 combined with the spectral differences of equations (3.9) results in

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^n}{\Delta t} = L\kappa \left(\eta_{i_r}^{n+1}\right)^{(2)} + Lm \left((\eta_{i_r}^n)^3 + \eta_{i_r}^n - 2\eta_{i_r}^n \sum_{j=1}^{p} (\eta_{j_r}^n)^2 \right),$$

$$r = 1, \ldots, N, \qquad i = 1, \ldots, p, \tag{3.13}$$

Second-order semi-implicit BDF (SBDF2)

The second-order semi-implicit BDF method, or SBDF2, is given by

$$\frac{3\eta_i^{n+1} - 4\eta_i^n + \eta_i^{n-1}}{2\Delta t} = D(\eta_i^{n+1}) + 2R(\boldsymbol{\eta}^n) - R(\boldsymbol{\eta}^{n-1}), \qquad i = 1, \ldots, p. \tag{3.14}$$

One of the properties of this method is that it results in a strong decay of high-frequency error components [6], which is useful when applying multigrid methods. Moreover, it allows relatively large time steps and is recommended for most problems when using second-order central differences for the diffusive term [113].

Third-order semi-implicit BDF (SBDF3)

Another method that gives a strong decay of higher-frequency error components, is third-order semi-implicit BDF (SBDF3) [6]. This scheme is constructed as

$$\frac{1}{\Delta t}\left(\frac{11}{6}\eta_i^{n+1} - 3\eta_i^n + \frac{3}{2}\eta_i^{n-1} - \frac{1}{3}\eta_i^{n-2}\right) =$$

$$D(\eta_i^{n+1}) + 3R(\boldsymbol{\eta}^n) - 3R(\boldsymbol{\eta}^{n-1}) + R(\boldsymbol{\eta}^{n-2}), \qquad i = 1, \ldots, p. \quad (3.15)$$

SBDF3 is useful when higher-order approximations to the diffusive term are used [113].

3.2.2 Adaptive time stepping

For the solution of systems with different time scales, variable step size schemes are often essential to obtain computationally efficient and accurate results. In the case of grain growth, the initial large amount of small competing grains requires a relatively small time step, while larger time steps are desirable to capture the subsequent slowly changing, long-term evolution of the microstructure.

In [111], an adaptive time stepping scheme is applied to a phase field model of binary alloy solidification, following the variable step size strategy described in [60]. The idea behind this adaptive scheme is to compute a time step size Δt such that the (estimated) error is smaller than a tolerance value τ. Consider an attempted computed time step from time point t_n to time point t_{n+1}, with $t_{n+1} = t_n + \Delta t_n$, and an error estimate D_n. The new time step size Δt_{new} is then determined as

$$\Delta t_{\text{new}} = r\Delta t_n, \qquad r = \left(\frac{\tau}{D_n}\right)^{\frac{1}{\tilde{p}+1}}, \quad (3.16)$$

with $\tilde{p} = p$ the order of the time discretisation scheme if D_n is an appropriate estimate of the error of the method. Often, the estimate D_n is quite rough and $\tilde{p}$ may be less than p [60]. The time step is adapted throughout a simulation roughly as described in Algorithm 3.5.

Algorithm 3.5: Adaptive time stepping

```
foreach time step $t_n$ do
    while time level $t_{n+1}$ is not computed do
        Apply time integration formula with time step size $\Delta t_n$;
        Compute error estimate $D_n$;
        if $D_n \leq \tau$ then
            Accept the attempted time step from $t_n$ to $t_{n+1}$;
            Compute $\Delta t_{n+1}$ such that $D_{n+1}$ will be close to $\tau$;
        else
            Reject the attempted time step;
            Compute a new time step size $\Delta t'_n$ such that $D'_n$ is closer to $\tau$;
            $\Delta t_n \longleftarrow \Delta t'_n$;
        end
    end
end
```

Algorithm 3.5 is based on estimates, and often additional control on the decrease and increase of step sizes is desirable. In most codes, the expression for the new trial step size therefore has the form

$$\Delta t_{\text{new}} = \min(r_{\max}, \max(r_{\min}, \theta r))\Delta t_n, \tag{3.17}$$

where $r_{\max}$ and $r_{\min}$ are a maximal and minimal growth factor, and the parameter $\theta < 1$ makes the estimate conservative to avoid repeated rejections. The values of these parameters depend on the type of integration method. The implementation of [111] uses the parameter set $r_{\min} = 0.5$, $r_{\max} = 2.0$, and $\theta = 0.8$.

Multi-step methods, such as SBDF2 and SBDF3 use information from at least two previous time levels. Consequently, when variable step sizes are used with a multi-step method, the formula coefficients need to be adjusted to maintain the order of consistency [60]. In [135], easily implementable Variable Step size Implicit-Explicit (VSIMEX) linear multi-step methods are designed, analysed, and numerically investigated for time-dependent PDEs.

Variable step size implicit-explicit (VSIMEX)

In [135], a number of order-p VSIMEX schemes is developed that apply BDFs to the implicitly treated part, and extrapolate the explicitly treated part to time step t_{n+p}. Those schemes are referred to as order-p Variable Step size Semi-implicit BDF (VSSBDFp) schemes.

The SBDF1 scheme (3.11) is in fact a first-order VSIMEX scheme, since it allows for variable time stepping, and is the same as the VSSBDF1 scheme. The VSSBDF2 scheme is given by

$$\frac{1}{\Delta t_{n+1}}\left(\frac{1+2r_{n+1}}{1+r_{n+1}}\eta_i^{n+2}-(1+r_{n+1})\eta_i^{n+1}+\frac{r_{n+1}^2}{1+r_{n+1}}\eta_i^n\right)=$$

$$D(\eta_i^{n+2})+(1+r_{n+1})R(\boldsymbol{\eta}^{n+1})-r_{n+1}R(\boldsymbol{\eta}^n),$$

$$i=1,\dots,p. \quad (3.18)$$

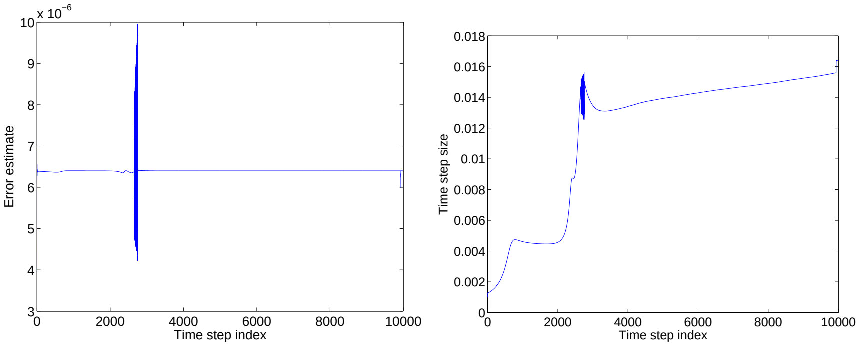
and corresponds to the constant step size SBDF2 scheme (3.14), with $r_n = \Delta t_n/\Delta t_{n-1}$.

To test the adaptive time stepping strategy of Algorithm 3.5, we solve system (2.37) by using the VSSBDF1 method, in combination with finite differences to approximate the spatial derivative. The error estimate D_n is obtained by comparing the solution of VSSBDF1 to the solution obtained by applying the second-order VSSBDF2 scheme. This allows us to use $\tilde{p} = p$ in formula (3.16) to determine the new time step size. Formula (3.17) is included in the implementation to provide extra control on the time step size. The same parameter values as in [111] are used.

Figure 3.1 shows the history of the error estimate D_n and the time step size Δt_n on a $32\times32\times32$ grid with 5 phase field variables and a relative tolerance $\tau = 10^{-5}$, during 10000 time steps. The other parameter values are $\kappa = 0.5$, $L = 1$, $m = 1$, and $\Delta x = 1$, and the initial time step size is set to $\Delta t_0 = 0.001$. The microstructure is initialised with uniformly distributed values on the interval $(-0.001, 0.001)$. It can be seen that the time step size increases during the start of the simulation and then oscillates around an equilibrium value.

The graph of the time step size in Fig. 3.1(b) shows different regimes. Some of these changes correspond to the changing characteristics of the solution of system (3.12). After a transitional period during which small grains nucleate and compete with each other (see Fig. 3.2(a)), larger grains start to appear and impinge on each other at, approximately, time step 2650, as depicted in Fig. 3.2(b). Once these grains have impinged and the microstructure is covered as in Fig. 3.2(c), grain growth sets in after approximately 2800 time steps. During grain growth, the largest values of the error estimate D_n are located at the grain boundaries and not within the grains.

Adaptive time stepping strategies have definite advantages, but in order to be able to apply a strategy with the parameters described as above, the values of all phase field variables at at least two time levels have to be kept in computing memory. For large-scale simulations, this may well turn out to be infeasible. Before a similar



(a) History of the error estimate D_n

(b) History of the time step size Δt_n

Figure 3.1: History of (a) the error estimate D_n and (b) the time step size Δt_n for the VSSBDF1 scheme (3.12) on a $32 \times 32 \times 32$ grid with 5 phase field variables, a relative tolerance $\tau = 10^{-5}$, $\kappa = 0.5$, $L = 1$, $m = 1$, and $\Delta x = 1$, for 10000 time steps.

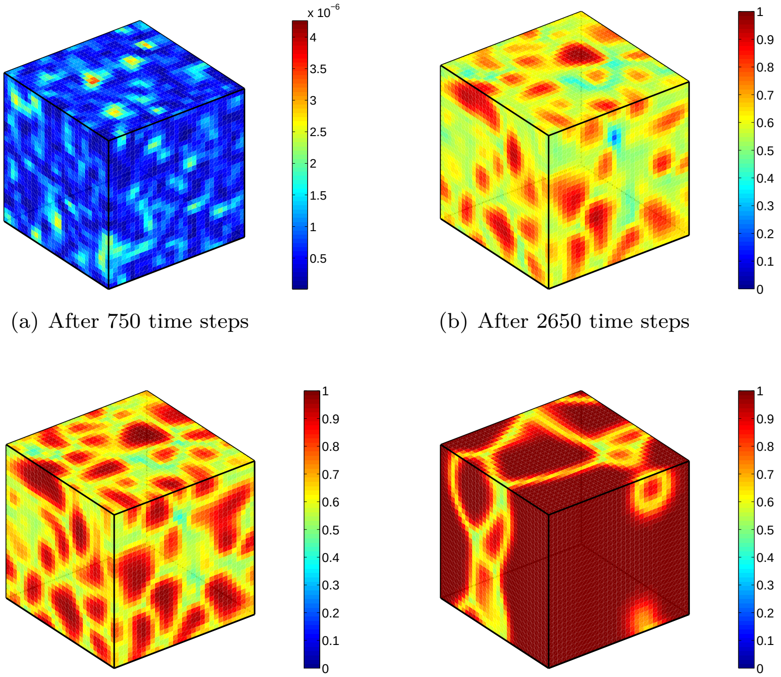
time strategy can be applied in realistic simulations with phase field models (2.37) and (2.27), the number of involved unknowns has to be reduced.

3.3 Sparse data structures

3.3.1 Adaptive mesh techniques

It can be very advantageous to adapt the resolution of the grid to the features of the solution of a system of partial differential equations, especially when these features are very small-scale when compared to the overall scale of the system. Possible advantages of this approach are a decrease in the amount of computing time and the storage requirements, since it can lead to a huge reduction of the number of involved unknowns. Moreover, it is possible to achieve high accuracy in space without excessive computational burden. When the features of the solution in addition change with time, it is appropriate to adapt the grid dynamically during a simulation in order to track the active regions of the solution.

One of the characteristics of phase field models is that they partition the system into domains within which phase field variables are relatively constant. The domain interfaces are represented by gradients in the values of the phase field variables. When the phase field models do not include any type of nucleation inside the domains, all evolution of the solution will take place in these interface regions. This locality of the evolution intuitively makes phase field simulations



(a) After 750 time steps

(b) After 2650 time steps

(c) After 2800 time steps

(d) After 10000 time steps

Figure 3.2: Microstructure on a $32 \times 32 \times 32$ grid with 5 phase field variables after (a) 750, (b) 2650, (c) 2800, and (d) 10000 time steps.

very appropriate for the application of adaptive mesh techniques. An adaptive mesh approach uses dense sets of grid points in the interfacial regions. The need to resolve these interfaces accurately, imposes the minimal grid spacing.

There are two types of adaptive meshing techniques that have been applied to phase field models. The first type of grid adaptation is the so-called h-adaptive method, where the mesh is locally refined or coarsened by adding or deleting grid points, in accordance with the values of the phase field variables. In [104, 105], it is described how a two-dimensional grid is adapted dynamically based on an error estimator that uses information from the two involved phase field variables. The grid elements are organised into a quadtree, which is a tree-like data structure with branches up to a prespecified level. Branches of the quadtree are in turn quadtrees. The execution time of this adaptive method scales with the arc length of the simulated interfaces, which allows simulation of much larger systems [104, 105]. The need for adaptive mesh refinement is even more acute in three dimensions. The grid elements can now be stored in an octree data structure, which is a logical extension of the quadtree structure used in two dimensions [106]. The mesh refinement algorithm and grid data structure used in [111, 112] are very similar to that of [104, 105, 106]. The software used in [17] contains a different type of data structure, which is constructed to achieve good vector performance.

The second type of mesh adaptivity that has been applied to phase field models is the r-adaptive method, also called the moving mesh method, where mesh points are moved throughout the domain while the connectivity of the mesh is kept fixed [79, 11, 38, 125]. The main idea of the moving mesh approach is to construct a time-dependent mapping from the computational domain to the physical domain, such that the representation of the physical solution in the computational domain is "better behaved" [38]. The computational mesh is thus obtained by equidistribution of a monitor function tailored to the variation of the phase field variables in the interfacial region [79]. An advantage of this approach is that it preserves the same number of Fourier modes, as opposed to a mesh refinement method. As a consequence, spectral differences can be used, allowing for high accuracy in space. Simulation results show that for a prescribed accuracy, the moving mesh method can lead to an order of magnitude improvement in efficiency [38]. A thorough overview of the theory and application of the moving mesh technique is given in [128].

The works of [104, 105, 106], [111, 112] and [17] are in the context of phase field simulation of dendritic solidification. In [79, 11, 38], different phase field models are tested. A common factor of these studies is that the adaptive mesh techniques are applied to simulations with only a small number of phase field variables. For polycrystalline structures, their benefits drastically reduce because of the amount of interface involved. The coupling between the many phase field variables, each with their associated interface regions, makes the application of an adaptive mesh technique a complex matter, both in theory and in implementation.

3.3.2 Dynamic grain orientation assignment

For isotropic grain growth, the free energy density (2.34) of the continuum field model is symmetric with respect to the local exchange of the values of phase field variables, representing the grain orientations. Local here refers to the fact that the exchange of values takes place on a subset of grid points of the full grid. If the subset where this exchange is performed is chosen such that the gradients of the phase field variables vanish at the edges of the subset, the operation will leave the total free energy (2.33) unchanged as well [74]. It is thus possible to reassign grain orientations without affecting the thermodynamic driving force for coarsening. An impending coalescence between two neighbouring grains can be avoided by simply reassigning the orientation of one of the grains to one not associated with any nearby grain. This strategy of dynamic grain orientation assignment is found both easy to implement and effective in suppressing the rate of coalescence, thus reducing the number of required phase field variables from more than 200 to about 20 in a three-dimensional grain growth simulation [74].

Unfortunately, the dynamic grain orientation assignment algorithm of [74] 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 neighbouring grains into this technique is difficult. The algorithm of [74] is made suitable for parallel computing in [122], and extended for parallel computing in an anisotropic system in [124]. However, in a subsequent paper, the authors switch over to the use of a different, sparse algorithm, developed in [133], which is “quite easy compared with that of the dynamic grain orientation reassignment” [123].

3.3.3 Sparse algorithms

When an obstacle potential such as equation (2.21) is included in a phase field model of grain growth, the values of the different phase field variables are driven to their equilibrium values within finite grain boundary regions. In [97], this property is exploited by solving the model equations only within these interface regions using activation flags. In contrast, when a double-well potential such as equation (2.22) or (2.36) is chosen, for phase field models of grain growth especially, the number of involved unknowns can be cumbersome. In [45, 133, 65], efficient algorithms using a sparse data structure are proposed to overcome this problem. These algorithms are related to the computational algorithm of [97] and are all based on the observation that only a few crystallographic orientations are active at a given point in a microstructure. A close look at the solutions of model (2.27) or (2.37) during a grain growth simulation indeed reveals that the values of the phase field variables display small regions of high activity surrounded by large regions of inactivity. This is clearly the case for the polycrystalline microstructure depicted on Fig. 2.10.

Both [45] and [133] developed a sparse algorithm for the model of [25], which is the same as model (2.37). The algorithm of [65] was constructed in the context of the multi-phase field models of [119, 66], which correspond to model (2.27). These sparse algorithms show significant improvements over conventional algorithms as they scale with the size of the microstructure instead of with the number of crystallographic orientations involved.

Sparse data structure and algorithm (Gruber et al. [45])

In [45], the microstructural evolution is only computed for non-zero phase field values. A simple data structure is proposed, which maintains a dynamic vector of the indices and the values of the non-zero phase field variables at each grid point. All phase field variables that exceed a small threshold value ϵ are considered non-zero. During a simulation, the updates of the phase field variables at every grid point require a different treatment, depending on the newly computed values:

1. the value of an existing non-zero phase field variable is changed without further action;
2. when the number of non-zero phase field variables at a grid point increases, new storage is added to the data vector of that particular grid point;
3. the data associated with a phase field variable that has dropped below the threshold value ϵ is deleted.

Throughout a simulation, any phase field variable with a value less than ϵ is considered to be zero and thus not retained. For sufficiently coarse systems, the number of non-zero phase field variables in any neighbourhood is usually one within a grain and seldom larger than three or four at a two-dimensional triple junction [45]. The memory required by this sparse data structure thus scales with the number of grid points.

As mentioned above, in [45], the computation of model equations (2.37) is restricted to the non-zero phase field variables at each grid point. The new value for a phase field variable at any grid point is therefore only calculated if it is non-zero at some neighbouring grid point. The sparse algorithm now proceeds as follows:

Algorithm 3.6: Sparse algorithm according to Gruber et al. [45]

```
for $t = t_1, t_2, \ldots, t_{end}$ do
    foreach grid point do
        Generate a list of unique phase field variables that are non-zero in
        the neighbourhood of the grid point;
        Solve the model equations only for those phase field variables in the
        list;
    end
end
```

As with the memory requirements, the sparse data structure of the algorithm causes the simulation time to scale with the number of grid points, independently of the number of phase field variables. Furthermore, the algorithm becomes increasingly efficient as the grain size increases, since the average number of non-zero phase field variables per grid point decreases with time. Simulation results obtained with the sparse algorithm are compared with those obtained by a conventional simulation algorithm in [45]. For a two-dimensional system of 512×512 grid points with $p = 100$ and $\epsilon = 0.001$, for 10000 iterations, the sparse algorithm uses approximately 7% of the computing memory required by the conventional algorithm on the full grid, and only 4% of its total simulation time. The sparse data structure and simulation algorithm described above are presented and tested in [45] for two-dimensional systems, but can easily be extended to grain growth in three dimensions.

Active parameter tracking algorithm (Vedantam et al. [133])

In [133], a very similar data structure is employed. Furthermore, a simple test condition is used, which distinguishes the evolving phase field values at grain boundaries from the constant phase field values within the grains. The model equations are then only solved for the evolving phase field values. At later times in the simulation, when the grain boundary area is only a small fraction of the overall system, additional computational savings are thus made.

Algorithm for effective phase field computation (Kim et al. [65])

In [65], an efficient algorithm is constructed in the context of the models of [119, 66], which correspond to multi-phase field model (2.27). The algorithm uses a data structure that is similar to those of the two algorithms that are described above, but puts a restriction N_p on the maximal number of non-zero phase field variables at every grid point and only stores the indices and the values of the N_p phase field variables with the largest magnitudes.

To support the argument that the phase field variables do not have to be kept in memory at all grid points, preliminary computations of grain growth are performed in [65]. Based on the resulting data, it is concluded that during a simulation, at almost all grid points, the number of phase field variables with positive values is less than 6 in three dimensions and less than 5 in two dimensions. Consequently, a restriction can be put on the maximal number N_p of positive phase field variables coexisting at one grid point. This maximal number is found to have a negligible effect on the grain growth dynamics and means a significant reduction of the memory requirements.

At each moment in time, the state of the microstructure at a grid point (x, y, z) is described by a list of N_p pairs (p_1, q_1), (p_2, q_2), $\ldots$, (p_{N_p}, q_{N_p}). The real variables $p_i(x, y, z)$ are the values of the phase field variables at grid point (x, y, z), ordered as $1 \geq p_1 \geq p_2 \geq \ldots \geq p_{N_p} \geq 0$. The integer variables $q_i(x, y, z)$ are the corresponding indices of the phase field variables. The sparse algorithm of [65] now proceeds as follows:

Algorithm 3.7: Algorithm for effective phase field computation according to Kim et al. [65]

```
for $t = t_1, t_2, \dots, t_{end}$ do
    foreach grid point do
        Retrieve all pairs $(p_i, q_i)$ and those of the nearest grid points;
        Solve the model equations only for the retrieved phase field variables;
        Reorder the new pairs such that $1 \geq p_1^{\text{new}} \geq \dots \geq p_{N_{\text{new}}}^{\text{new}} \geq 0$;
        Update the list of pairs;
        if $N_{new} > N_p$ then
            Keep only the indices and values of the $N_p$ largest phase field
            variables in memory;
        end
        Perform $p_i \longleftarrow p_i / \sum_{j=1}^{\min(N_{\text{new}}, N_p)} p_j$ to ensure that $\sum p_i = 1$;
    end
end
```

The correction operation in the last step of Algorithm 3.7 is essential: omitting this step would result in unstable computations [65].

Concluding remarks

The three algorithms described above are designed for use with an explicit time integration scheme. Explicit time integration schemes do not involve the solution of coupled equations, but can require large amounts of computing time due to impractically small time steps, imposed by strong stability conditions. In contrast, implicit and semi-implicit schemes require more computations due to the involved coupled equations, but have better stability properties and therefore, larger time steps can be used. In [24], a semi-implicit scheme is shown to allow much larger time step sizes than explicit schemes for the model of [25]. Unfortunately, it is difficult to apply this semi-implicit scheme to the sparse data structures of [45], [133], and [65].

Each of the described sparse algorithms employs a data structure that maintains besides the values of the active phase field variables at every grid point, also their indices. There is some overhead associated with this storage scheme, since the same index values are stored at multiple locations. Simulations with a very small total number of phase field variables should therefore prefer a conventional phase field simulation method on a full grid [45]. For simulations starting with random noise or with a microstructure whose grain size is comparable to the grain boundary thickness, a conventional simulation algorithm should be applied until a well-defined polycrystalline microstructure is developed. At this point the sparse algorithm can be used. For models that include anisotropic boundary properties, a logical extension to the sparse data structures would be to store additional material parameters at each grid point, which would only add to the overhead.

One of the main goals of grain growth simulations is to study the evolution of the mean grain size and the grain size distribution. For this purpose, the grains of a microstructure have to be resolved, and their volume has to be computed. This information is not readily available in the sparse data structures of [45, 133, 65]. Therefore, it can take a relatively large amount of time to obtain the necessary information, especially when this post-processing is repeated for multiple microstructures at different moments in their evolution.

In Chapter 5, a sparse bounding box data structure is developed that offers a solution to many of these issues.

3.4 Efficient solvers

When an implicit time integration scheme is applied to a set of partial differential equations, a system has to be solved at each time step in the simulation. Depending on the characteristics of the applied discretisation scheme, this system will have a different size and complexity. To make sure that the possibility of a large time step compensates the execution time required to solve the discretised equations, it is necessary to look for dedicated solvers.

3.4.1 Multigrid methods

The convergence rate of standard iterative solvers such as the Jacobi or the Gauss-Seidel method typically stalls after a small number of iterations. The latter methods remove the high-frequency modes of the error after a few iterations already, while the low-frequency modes remain much longer present, which causes a decrease of the convergence rate after a few iteration steps. Multigrid algorithms accelerate the convergence of standard iterative solvers. They are based on the idea that the remaining smooth, low-frequency error on the fine grid can be removed efficiently by certain computations on a coarser grid. Since coarse-grid calculations are significantly less expensive than calculations on a fine grid, the overall efficiency of an iterative solver can be improved this way.

To solve the system on the coarser grid, the two-grid idea mentioned above can be applied recursively: the low-frequency error on the coarser grid in turn can be removed by computations on a grid that is even more coarse. This way, a multigrid cycle is obtained that uses an entire grid hierarchy. The smallest system, on the coarsest grid of the hierarchy, can be solved by a direct solver. In this work, we will apply standard coarsening to construct the grid hierarchy, which means that coarser grids are obtained by doubling the grid spacing in all dimensions. Figure 3.3 illustrates the grid hierarchy thus obtained for a two-dimensional system.

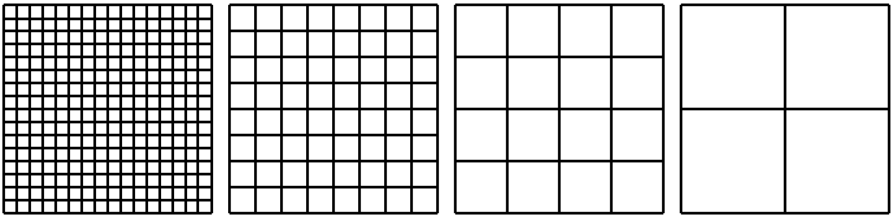


Figure 3.3: Multigrid methods use an entire grid hierarchy to solve a system. In this work, we apply standard coarsening to construct the grid hierarchy, which means that coarser grids are obtained by doubling the grid spacing in all dimensions.

When the components of a multigrid solver are carefully selected, the solver's convergence properties can be independent of the grid size. Moreover, the computational work of a multigrid solver can scale linearly with the number of spatial unknowns. These features make multigrid solvers very appropriate for large-scale simulations. A number of studies has been performed on the development of multigrid methods for phase field simulations. In [111], a nonlinear multigrid solver is implemented to solve a phase field model for binary alloy solidification. This solver is based on the Full Approximation Scheme (FAS) [16] and follows an adaptive grid approach. Monotone multigrid solvers are constructed in the work of [73] to solve vector-valued Allen-Cahn equations, which are extensively used for phase field modelling. In [62], an extension of the method of [73] is presented to solve the Cahn-Hilliard equation, which is often employed in conservative phase field models. The latter equation type also has been solved by Uzawa-type multigrid algorithms [10], linear multigrid methods [21] and FAS methods [64, 139]. However, in all of these studies, only a few phase field variables are considered.

Chapter 6 goes more deeply into the application of multigrid methods to phase field models. In that chapter, the principles of multigrid methods are introduced and a nonlinear FAS multigrid solver is constructed, implemented and tested for the solution of multi-phase field model (2.27).

3.4.2 Fast Fourier transform

In [24], an efficient and accurate numerical method, also-called the Fourier spectral method, is implemented to solve model (2.37). It is demonstrated that for a specified accuracy of 0.5%, the speed-up of using this method, when compared with an explicit finite difference scheme, is at least two orders of magnitude in two dimensions, and close to three orders of magnitude in three dimensions. A similar solver was used in [28] to solve a related phase field model, which was discretised with finite differences. The reason why these solvers are so fast, is because they make use of the Fast Fourier Transform (FFT) to solve the system of equations, which scales as $N\log(N)$, with N the number of involved unknowns [27]. Of course, there are very efficient codes available that implement the FFT, see for example [39].

Fast Fourier transform applied to a finite difference scheme

Applying the Discrete Fourier Transform (DFT) to equations (3.12) results in a decoupled system of equations in Fourier space:

$$\frac{\hat{\eta}_i^{n+1}(k)-\hat{\eta}_i^{n}(k)}{\Delta t} = L\kappa\lambda(k)\hat{\eta}_i^{n+1}(k)+\hat{R}_i(k),$$
$$k=-\frac{N}{2}+1,\dots,\frac{N}{2},\quad i=1,\dots,p. \qquad (3.19)$$

The term $\hat{R}_i$ is the DFT of the reaction part of equations (3.12) and equals

$$\hat{R}_i = Lm\mathcal{F}\left(\left(\eta_{i_r}^n\right)^3+\eta_{i_r}^n-2\eta_{i_r}^n\sum_{j=1}^{p}\left(\eta_{j_r}^n\right)^2\right),$$
$$r=1,\dots,N,\quad i=1,\dots,p, \qquad (3.20)$$

where $\mathcal{F}$ denotes the Fourier transform. The factor $\lambda(k)$ can be computed by using the shift property of the DFT, namely

$$\mathcal{F}(\eta_{i_r})_k=\hat{\eta}_i(k) \quad\Longleftrightarrow\quad \mathcal{F}(\eta_{i_{r-m}})_k=\exp\left(-\iota\frac{2\pi km}{N}\right)\hat{\eta}_i(k), \qquad (3.21)$$

with m a chosen integer. Because the Laplacian in equation (3.12) is discretised with second-order finite differences, in this case $m=-1,0,$ or 1. The factor $\lambda(k)$

thus equals

$$\lambda(k) = \frac{1}{(\Delta x)^2} \mathcal{F}\left(\eta_{i_{r+1}} - 2\eta_{i_r} + \eta_{i_{r-1}}\right) \tag{3.22a}$$

$$= \frac{2}{(\Delta x)^2}\left(\cos\left(\frac{2\pi k}{N}\right) - 1\right). \tag{3.22b}$$

System (3.19) is now easily solved by performing a division:

$$\hat{\eta}_i^{n+1}(k) = \frac{\hat{\eta}_i^n(k) + \Delta t \hat{R}_i(k)}{1 - \Delta t L\kappa\lambda(k)}, \qquad k = -\frac{N}{2}+1, \dots, \frac{N}{2}, \quad i = 1, \dots, p. \tag{3.23}$$

The solution in physical space is obtained by computing the inverse DFT of $\hat{\eta}_i^{n+1}(k)$. This solution method was applied to the Cahn-Hilliard equation in [28].

Fast Fourier transform applied to a spectral difference scheme

To solve scheme (3.13), we transform the equations to the Fourier space:

$$\frac{\hat{\eta}_i^{n+1}(k) - \hat{\eta}_i^n(k)}{\Delta t} = -k^2 \left(\frac{2\pi}{N\Delta x}\right)^2 L\kappa\hat{\eta}_i^{n+1}(k) + \hat{R}_i(k),$$

$$k = -\frac{N}{2}+1, \dots, \frac{N}{2}, \quad i = 1, \dots, p. \tag{3.24}$$

Again, this system is solved easily by performing a division, namely:

$$\hat{\eta}_i^{n+1}(k) = \frac{\hat{\eta}_i^n(k) + \Delta t \hat{R}_i(k)}{1 + \Delta t k^2 \left(\frac{2\pi}{N\Delta x}\right)^2 L\kappa}, \qquad k = -\frac{N}{2}+1, \dots, \frac{N}{2}, \quad i = 1, \dots, p. \tag{3.25}$$

The solution in the spatial domain is obtained by computing the inverse DFT of $\hat{\eta}_i^{n+1}(k)$.

Concluding remarks

The application of the DFT is limited to systems on a uniform grid. In Section 3.3.1, it is described how this method nevertheless can be combined with adaptive mesh refinement.

We have made an implementation of both Fourier schemes by making use of the FFTW library, which is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data [39]. The implementation of solver scheme (3.23) with the FFTW library is used in Chapter 5 to generate initial microstructures, as well as as a reference point for comparison with the sparse bounding box algorithm developed in the latter chapter.

3.5 Parallel computing

Parallel computing offers a solution to the high computational requirements of realistic three-dimensional phase field simulations, both in memory and in time. So far, however, only a few authors have used parallel computing to perform large-scale phase field simulations of grain growth [97, 122, 124, 129].

In [97], a three-dimensional parallel simulator of crystal growth and solidification in complex alloy systems is described. No details are given on the specific parallelisation scheme, but it is stated that the solver is parallelised for distributed as well as shared memory computer architectures using both Message Passing Interface (MPI) libraries and Open Multi-Processing (OpenMP) concepts. The Message Passing Interface, or MPI, is the specification of a portable, high-performance application programming interface designed for parallel programming on distributed memory machines [94]. MPI provides functions for e.g. process management, communication, and synchronisation. In contrast, OpenMP is an API that supports shared memory programming, which relies on the availability of directly addressable global memory [102]. The API of OpenMP consists of compiler directives that indicate to the compiler how parallelism can be extracted from the source code.

As mentioned in Section 3.3.2, the dynamic grain orientation assignment algorithm of [74] was parallelised in [122, 124]. In their work, both MPI and an automatic parallelisation scheme of the FORTRAN compiler are used to parallelise the solver of the phase field model. The data representing the phase field variables is divided over the different involved processors, while specific information that is required for the dynamic orientation assignment, is present on all processors. The orientation assignment algorithm itself is parallelised through MPI.

The use of an IMEX time discretisation scheme in the context of model (2.37), effectively decouples the systems into p smaller systems such that parallel computing is possible. This property is exploited in Section 5.3.1 to parallelise an efficient algorithm for the solution of phase field model (2.39) on a full grid, as well as in Section 5.5, to parallellise the sparse bounding box method developed in Chapter 5.

3.6 Conclusion

The goal of this thesis is to design, analyse and implement efficient numerical algorithms for grain growth simulation with the multi-phase field model (2.27) and the continuum field model (2.37). Both models describe a polycrystalline microstructure with a large set of phase field variables, with one phase field variable corresponding to each crystallographic orientation. Realistic three-dimensional computer simulations of grain growth with these phase field models require the use of a large number of phase field variables and therefore demand significant amounts of computation power as well as data storage.

In this chapter, we have given an overview of the different computational techniques that have been applied to overcome the computational constraints of the multi-phase field model and the continuum field model. The strategies tested so far range from the application of different discretisation schemes to the use of sparse data structures and the development of efficient solvers. In particular, the sparse algorithms of [45], [133] and [65] show a significant improvement over the existing techniques as they scale with the size of the microstructure instead of with the number of crystallographic orientations involved. In Chapter 5, we will develop a sparse bounding box data structure in the context of the continuum field model that offers a solution to many of the unresolved issues with the latter three algorithms. Another promising strategy for large-scale phase field simulation is the application of multigrid solvers, as they can scale linearly with the number of spatial unknowns. However, all multigrid solvers developed until now only consider a few phase field variables. In Chapter 6, we will construct a nonlinear multigrid method to solve the multi-phase field model for multiple phase field variables.

Chapter 4

Stability of the continuum field model and its discretisation

4.1 Introduction

Continuum field model (2.37) naturally splits into two parts, namely a reaction part and a diffusion part. In Section 3.2.1, we described how IMEX schemes treat these two parts differently: the reaction part is discretised by an explicit scheme, while the diffusion part is discretised by an implicit scheme. This approach effectively decouples the model equations into several smaller systems that can be solved efficiently, which creates possibilities for parallel computing.

Explicit discretisation schemes are very efficient in computation, but unfortunately, their stability properties limit the time step size. In contrast, implicit or semi-implicit discretisation methods allow the choice of a larger time step. In this chapter, we compare the stability properties of the first-order IMEX scheme SBDF1 with those of the Forward Euler scheme and the Backward Euler scheme. We restrict ourselves to first-order methods, which require storage of just a single solution. This is indeed the only feasible alternative in the case of large-scale three-dimensional phase field simulations.

This chapter starts with a brief study of the semi-discretised continuum field model in Section 4.2, after which an analysis of the two parts of the fully discretised continuum field model separately is performed in Section 4.3. Both parts are shown to have a different effect on the stability properties of the model. In the subsequent Sections 4.4, 4.5, and 4.6, we perform a stability analysis of respectively the Forward Euler scheme, the Backward Euler scheme and the SBDF1 scheme.

For each discretisation scheme, the resulting stability conditions are checked with numerical experiments. Section 4.7 ends this chapter with a summary and concluding remarks.

4.2 Analysis of the semi-discretised model

In this section, the stability properties of the semi-discretised equations (3.3) of continuum field model (2.37) are briefly examined. In particular, we want to find out whether it is possible that numerical solutions of the continuum field model arrive in an unstable configuration during a typical grain growth simulation, and what will happen in such an event. Furthermore, for the use of multigrid solvers, which employ a hierarchy of grids with different spacings (see Section 3.4.1), it is important to know if and how these stability properties depend on the value of the grid spacings.

To study the stability properties, at each time step of a grain growth simulation, we linearise equations (3.3) about the current solution by composing the Jacobian. Next, we examine the eigenvalues of the Jacobian, which are in fact the growth rates of the associated eigenvectors. Two numerical experiments are performed on a one-dimensional grid, with $p = 2$. The other model parameters are chosen as $\kappa = 0.5$, $L = 1$, and $m = 1$. This choice of parameter values is justified in [87] and returns in Section 7.2, where the set-up of simulations of grain growth in the presence of second-phase particles is described.

In the first experiment, the phase field variables are initialised as

$$\eta_1^0(r) = \exp\left[-\left(5\left(\frac{r}{l}\Delta x - \frac{1}{2}\right)\right)^6\right], \tag{4.1a}$$

$$\eta_2^0(r) = 1 - \eta_1^0(r), \qquad r = 1, \ldots, N, \tag{4.1b}$$

where the interval length l equals 64 and the grid spacing takes different values $\Delta x = 4$, 2, 1, and 0.5. With these initial values, the one-dimensional equivalent of equations (3.12) is solved for 600 time steps, with $\Delta t = 0.2$. Figure 4.1(a) illustrates the values of the phase field variables at $t = 0$ and at $t = 120$. The evolution of the maximal eigenvalue λ_1 of the Jacobian, evaluated at each time step, is depicted as a function of the time step index on Fig. 4.1(b). At the start of the simulation, the value of λ_1 is positive, which indicates the presence of unstable modes. For $\Delta x = 4$, after a short transition time, the value of λ_1 drops below zero and no unstable modes are left. For decreasing values of Δx, the stability properties are similar, but the transition time after which all instabilities have disappeared becomes longer.

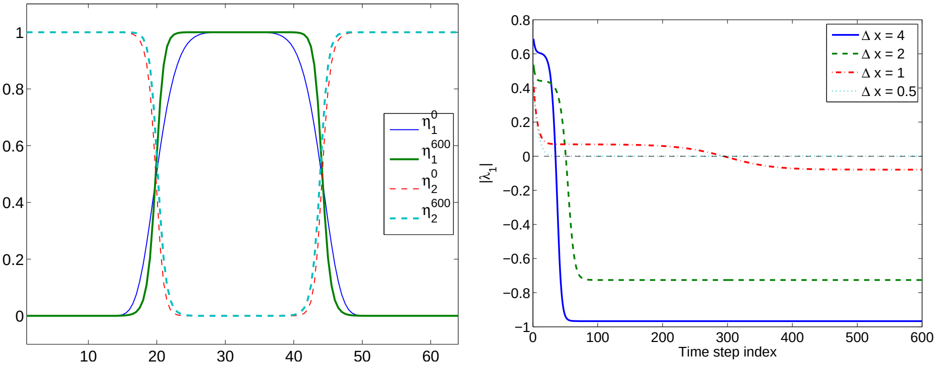


Figure 4.1: (a) Values of the phase field variables η_1 and η_2 at time points $t = 0$ and $t = 120$ for $\Delta x = 0.5$ and (b) the time evolution of the maximal eigenvalue λ_1 of the linearised operator, for different values of the grid spacing Δx.

For the second numerical experiment, a one-dimensional system of 128 grid points is initialised with values that are generated according to a uniform distribution over the interval $(0, 0.001)$. The same model parameter values are applied as in the first experiment, but now with a fixed value of the grid spacing, namely $\Delta x = 1$. With this initial state, the one-dimensional equivalent of equations (3.12) is solved again for 600 time steps, with $\Delta t = 0.2$. Figure 4.2(a) shows the values of the phase field variables η_1 and η_2 both at $t = 0$ and at $t = 120$. For clarity, the complete spectrum of the Jacobian is only depicted at specific time points on Fig. 4.2(b). The evolution of the maximal eigenvalue λ_1 is indicated at all time steps. At the start of the simulation, the eigenvalues are spread over both the positive and the negative side of the zero-axis. After a transition time, the maximal eigenvalue drops below zero and all modes of the Jacobian are stable. The sharp drops of the value of λ_1 correspond to changes in the polycrystalline structure. Once the microstructure is covered with grains and a well-defined polycrystalline structure has appeared, no unstable modes remain present.

Both experiments described above indicate that at the start of a simulation, many unstable modes are present, which disappear after a certain transition time, depending on the formation of a well-defined polycrystalline microstructure. As we are interested in the study of the influencing factors of grain growth in an existing polycrystalline structure, rather than the modelling and simulation of the formation of new grain structures, the unstable modes at the start of the simulations pose no problem. Furthermore, the use of a coarser grid spacing Δx does not increase instabilities. In [9], a theoretical stability study is performed of the Cahn-Hilliard equation, often used in conservative phase field models, and of a related phase field equation. In the latter work, the occurrence of unstable modes is connected to the coarsening of the system.

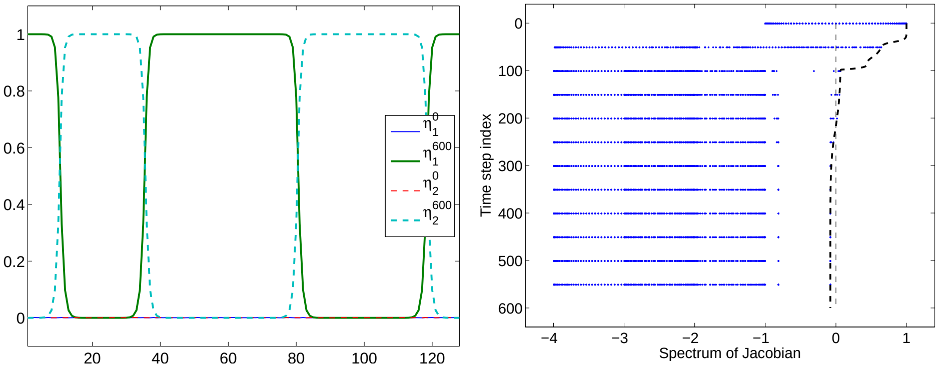


Figure 4.2: (a) Values of the phase field variables η_1 and η_2 at time points $t = 0$ and $t = 120$ and (b) the time evolution of the complete spectrum of the linearised operator. The evolution of the maximal eigenvalue λ_1 is indicated.

4.3 Analysis of the fully discretised model

Continuum field model (2.37) consists of two parts, namely a reaction part and a diffusion part. In this section, we analyse the stability properties of the Forward Euler scheme applied to both parts separately. For reasons of clarity, we will restrict the theoretical derivations in this section and in the next sections to the one-dimensional case.

4.3.1 Analysis of the diffusion part

The first part of the right-hand side of the isotropic continuum field model is a diffusion term. Dropping the reaction part, we obtain p decoupled diffusion equations:

$$\frac{\partial \eta_i}{\partial t} = L\kappa \nabla^2 \eta_i, \qquad i = 1, \ldots, p. \tag{4.2}$$

Application of the Forward Euler method to equations (4.2) and of second-order central finite differences to the Laplacian yields the following system:

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^{n}}{\Delta t} = L\kappa \frac{\eta_{i_{r+1}}^{n} - 2\eta_{i_r}^{n} + \eta_{i_{r-1}}^{n}}{(\Delta x)^2}, \quad r = 1, \ldots, N, \quad i = 1, \ldots, p. \tag{4.3}$$

In accordance with the stability analysis performed in [93], where the amplification of Fourier modes is studied, the stability condition of the discretised diffusion equations (4.3) is

$$\Delta t < \frac{(\Delta x)^2}{2L\kappa} \; . \tag{4.4}$$

In three dimensions, this condition becomes

$$\Delta t < \frac{(\Delta x)^2}{6L\kappa} \; . \tag{4.5}$$

4.3.2 Analysis of the reaction part

The second part of the right-hand side of the isotropic continuum field model is a reaction part. To study the latter part, we drop the diffusion part and obtain the set of equations

$$\frac{\partial \eta_i}{\partial t} = Lm \left(\eta_i^3 + \eta_i - 2\eta_i \sum_{j=1}^{p} \eta_j^2 \right), \qquad i = 1, \ldots, p. \tag{4.6}$$

Discretisation of equations (4.6) with the Forward Euler scheme results in the following system:

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^{n}}{\Delta t} = Lm \left(\left(\eta_{i_r}^{n}\right)^3 + \eta_{i_r}^{n} - 2\eta_{i_r}^{n} \sum_{j=1}^{p} \left(\eta_{j_r}^{n}\right)^2 \right), \qquad r = 1, \ldots, N, \qquad i = 1, \ldots, p. \tag{4.7}$$

Because of the nonlinearity in system (4.7), we cannot study the amplification of Fourier modes as such. Instead, we will examine the amplification of small perturbations in the form of scaled Fourier modes on the equilibrium values of system (4.6). The latter values correspond to the zeros of the right-hand side of equations (4.6), namely $\eta_i(x) = 1$, $\eta_i(x) = -1$ and $\eta_i(x) = 0$.

First, we analyse equilibrium solution $(\eta_1, \ldots, \eta_i, \ldots, \eta_p) = (0, \ldots, 1, \ldots, 0)$ by adding a small perturbation to the equilibrium value $\eta_i(x) = 1$ of the form

$$V_r^n = \epsilon \lambda(k)^n \exp\left(\iota \pi k \frac{r\Delta x}{l} \right), \qquad r = 1, \ldots, N, \qquad k = -(N-1), -(N-2), \ldots, N, \tag{4.8}$$

with l the length of the domain interval, Δx the grid spacing, and ϵ a small value. The Forward Euler scheme (4.7) is stable with regard to this perturbation if the amplification factor $\lambda(k)$ in (4.8) satisfies the condition

$$|\lambda(k)| < 1, \tag{4.9}$$

for every wavenumber k.

Substitution of $\eta_{i_r}^n$ by $1 + V_r^n$ into system (4.7) results in:

$$\frac{V_r^{n+1} - V_r^n}{\Delta t} = Lm\left((1 + V_r^n)^3 + (1 + V_r^n) - 2\,(1 + V_r^n)^3\right), \qquad r = 1, \ldots, N. \tag{4.10}$$

We neglect the higher-order terms in ϵ in equations (4.10), which yields:

$$\frac{V_r^{n+1} - V_r^n}{\Delta t} = -2LmV_r^n, \qquad r = 1, \ldots, N, \tag{4.11}$$

and extract the amplification factor $\lambda(k)$ of the perturbation as:

$$\lambda(k) = 1 - 2\Delta t Lm. \tag{4.12}$$

The condition $\lambda(k) < 1$ is automatically fulfilled. Computations with scheme (4.7) are only stable with respect to the perturbation if $-1 < \lambda(k)$ is also true. The stability condition for the time step size Δt is thus

$$\Delta t < \frac{1}{Lm}\ . \tag{4.13}$$

The same stability condition is obtained when we add perturbation (4.8) to the value $\eta_i(x) = -1$ of the equilibrium solution $(\eta_1, \ldots, \eta_i, \ldots, \eta_p) = (0, \ldots, -1, \ldots, 0)$.

Next, we study the effect of perturbation (4.8) on the value $\eta_i(x) = 0$ of the equilibrium solutions $(\eta_1, \ldots, \eta_j, \ldots, \eta_p) = (0, \ldots, \pm 1, \ldots, 0)$. Substituting $\eta_{i_r}^n = 0 + V_r^n$ into equations (4.7) and neglecting the higher-order terms in ϵ yields the amplification factor

$$\lambda(k) = 1 - \Delta t Lm. \tag{4.14}$$

Computations with scheme (4.7) are stable if $\lambda(k)$ fulfils condition (4.9) and hence if Δt satisfies the inequality

$$\Delta t < \frac{2}{Lm}, \tag{4.15}$$

which is less strict than condition (4.13) found above. In order to perform stable computations with the Forward Euler scheme (4.7), the time step size Δt therefore has to obey stability condition (4.13).

4.3.3 Conclusion

In this section, we have analysed the stability properties of the Forward Euler scheme applied to the diffusion part and the reaction part of the isotropic continuum field model separately. The stability restriction on the time step size is found to be much stricter for the diffusion part than for the reaction part: condition (4.5) contains a dependency on the square of the grid spacing Δx. This is an indication that it is much more important to treat the former part implicitly than the latter part.

In the next three sections, we bring the two parts of the continuum field model back together and analyse the stability properties of three different discretisation schemes applied to the model: the fully explicit Forward Euler scheme and the fully implicit Backward Euler scheme, as well as the IMEX scheme that combines the latter two methods, namely the SBDF1 scheme (see also Section 3.2.1). Many elements of the analysis used in this section will return in the next sections.

4.4 Explicit discretisation

Application of the explicit Forward Euler scheme in combination with second-order central finite differences to the Laplacian in the isotropic continuum field model (2.37) results in the following system:

$$\frac{\eta_{i_r}^{n+1}-\eta_{i_r}^{n}}{\Delta t} = L\kappa\frac{\eta_{i_{r+1}}^{n}-2\eta_{i_r}^{n}+\eta_{i_{r-1}}^{n}}{(\Delta x)^2} + Lm\left(\left(\eta_{i_r}^{n}\right)^3+\eta_{i_r}^{n}-2\eta_{i_r}^{n}\sum_{j=1}^{p}\left(\eta_{j_r}^{n}\right)^2\right), \qquad r=1,\dots,N, \qquad i=1,\dots,p. \tag{4.16}$$

4.4.1 Analysis

A perturbation of the form (4.8) on the equilibrium value $\eta_i(x)=\pm 1$, substituted into scheme (4.16) yields the equations

$$\frac{V_r^{n+1}-V_r^{n}}{\Delta t} = L\kappa\left(\frac{V_{r+1}^{n}-2V_r^{n}+V_{r-1}^{n}}{(\Delta x)^2}\right) + Lm\left(\left(1+V_r^n\right)^3+\left(1+V_r^n\right)-2\left(1+V_r^n\right)^3\right), \quad r=1,\dots,N. \tag{4.17}$$

Before we continue, we derive the effect of the central difference on the perturbation V_r^n:

$$V_{r+1}^n - 2V_r^n + V_{r-1}^n = \exp\left(\iota\pi k\frac{\Delta x}{l}\right) - 2 + \exp\left(-\iota\pi k\frac{\Delta x}{l}\right) \tag{4.18a}$$

$$= -2\left(1 - \cos\left(\pi k\frac{\Delta x}{l}\right)\right) \tag{4.18b}$$

$$= -4\sin^2\left(\pi k\frac{\Delta x}{2l}\right) \tag{4.18c}$$

As in Section 4.3.2, we neglect the higher-order terms in ϵ in equations (4.17). Using equality (4.18c), we obtain

$$\frac{V_r^{n+1} - V_r^n}{\Delta t} = -\frac{4L\kappa}{(\Delta x)^2}\sin^2\left(\pi k\frac{\Delta x}{2l}\right)V_r^n - 2LmV_r^n, \quad r = 1, \ldots, N. \tag{4.19}$$

The amplification factor of the perturbation can be extracted from (4.19) as

$$\lambda(k) = 1 - \frac{4L\kappa\Delta t}{(\Delta x)^2}\sin^2\left(\pi k\frac{\Delta x}{2l}\right) - 2Lm\Delta t \tag{4.20}$$

and automatically fulfils the condition that $\lambda(k) < 1$. The most oscillatory mode, namely for wavenumber $k = N$, is the most sensitive to instabilities. The stability properties of the Forward Euler scheme (4.16) are accordingly determined by the inequality

$$-1 < 1 - \frac{4L\kappa\Delta t}{(\Delta x)^2} - 2Lm\Delta t = \lambda(N), \tag{4.21}$$

which yields the stability condition

$$\Delta t < \frac{1}{\frac{2L\kappa}{(\Delta x)^2} + Lm}\ . \tag{4.22}$$

In three dimensions, this stability condition becomes

$$\Delta t < \frac{1}{\frac{6L\kappa}{(\Delta x)^2} + Lm}\ . \tag{4.23}$$

Similarly, perturbation of the equilibrium value $\eta_i(x) = 0$ results in the condition

$$\Delta t < \frac{2}{\frac{4L\kappa}{(\Delta x)^2} + Lm} \tag{4.24}$$

in the one-dimensional case, while in the three-dimensional case, the following condition is obtained:

$$\Delta t < \frac{2}{\frac{12L\kappa}{(\Delta x)^2} + Lm} \, . \tag{4.25}$$

Since condition (4.23) is stronger than condition (4.25), the former determines the stability properties of the Forward Euler scheme. Both conditions contain a dependency on the square of the grid spacing Δx, which means that for a finer spatial discretisation the restriction on the time step soon becomes very tight, resulting in an infeasibly small Δt.

4.4.2 Numerical results

We test stability condition (4.23) with numerical experiments on a $32 \times 32 \times 32$ grid, for $p = 5$. The phase field variables are initialised with small values generated according to a uniform distribution over the interval $(-0.001, 0.001)$. The values of the model parameters are chosen as $\kappa = 0.5$, $L = 1$, and $m = 1$ (see also Section 4.2), and the discretisation spacing is $\Delta x = 1$. For the chosen set of parameters, according to condition (4.23), the time step size has to fulfil condition

$$\Delta t < \frac{1}{\frac{6L\kappa}{(\Delta x)^2} + Lm} = 0.25 \tag{4.26}$$

in order to perform stable computations with the explicit scheme (4.16).

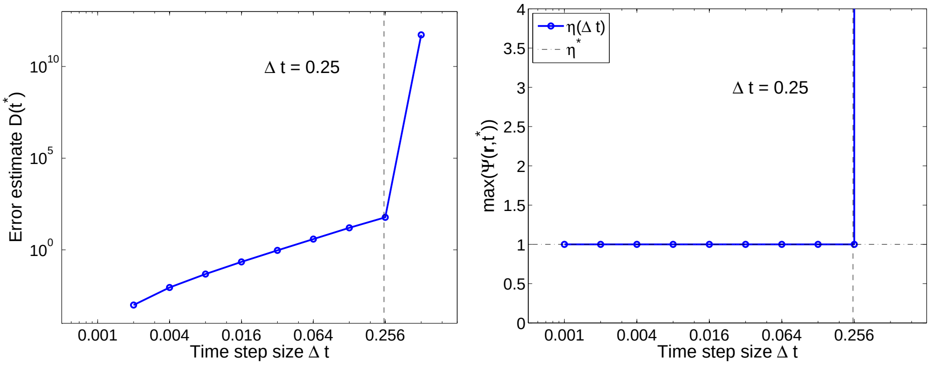
First, we approximate the exact solution $\boldsymbol{\eta}^*$ at time point $t^* = 40$ by evolving the phase field variables with an explicit time stepper based on scheme (4.16) for 40000 time steps with $\Delta t = 0.001$. The latter time step size is chosen well below stability restriction (4.26). Next, we perform simulations for increasing time step sizes Δt with the same explicit time stepper. Each simulation starts from the same initial state and runs until time point t^* is reached. For every tested time step size Δt, the error $D(t^*)$ at time point t^* is measured as:

$$D(t^*) = \sum_{\mathbf{r} \in \Omega} \left(|\Psi^*(\mathbf{r}) - \Psi(\mathbf{r}, t^*)|^2 \right) , \tag{4.27}$$

with Ω the system domain and with

$$\Psi(\mathbf{r}, t) = \sum_{i=1}^{p} \eta_i^2(\mathbf{r}, t). \tag{4.28}$$

The quantity Ψ^* in equation (4.27) is computed by applying formula (4.28) to the exact solution $\boldsymbol{\eta}^*$. When time step size Δt is chosen too large, small errors start



(a) Error measure $D(t^*)$

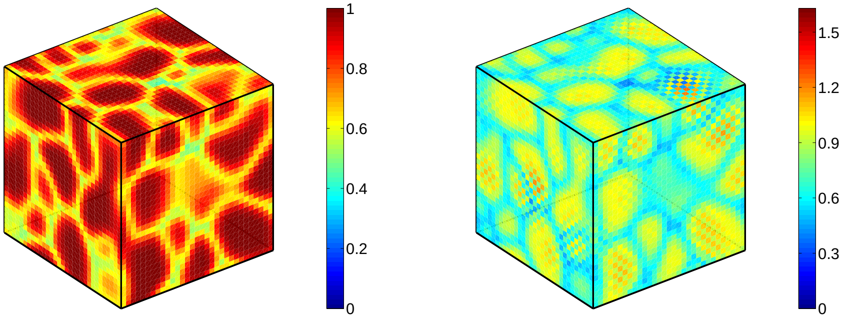
(b) $\max(\Psi(\mathbf{r}, t^*))$

Figure 4.3: Error measure $D(t^*)$ and the maximal value of $\Psi(\mathbf{r}, t^*)$ as a function of the time step size Δt, obtained from numerical experiments with the Forward Euler scheme (4.16) for $t^* = 40$, compared with the theoretically derived stability limit.

to grow and cause oscillations in the solution. These growing errors are detected by the error measure $D(t^*)$.

Figure 4.3(a) shows the error measure $D(t^*)$ as a function of the time step size Δt. The theoretically expected stability limit (4.26) is indicated as well. The graph of the error measure slowly increases with Δt until approximately $\Delta t = 0.25$, after which the slope of the graph significantly increases. This steep rise is due to the onset of instabilities in the computations. This can also be seen on Fig. 4.3(b), where the maximal value of $\Psi(\mathbf{r}, t^*)$, taken over all grid points of the system domain Ω, is depicted as a function of time step size Δt. For the exact solution $\boldsymbol{\eta}^*$, this maximal value is approximately equal to 1. Figure 4.3(b) shows that for a time step size larger than $\Delta t = 0.25$, the value of $\max(\Psi(\mathbf{r}, t^*))$ explodes. Both Figures 4.3(a) and 4.3(b) thus confirm stability condition (4.23).

According to the formula for the amplification factor (4.20), the mode that is the most sensitive to instabilities is the most oscillatory mode, for $k = N$. Figure 4.4 depicts the values of $\Psi(\mathbf{r}, t)$ at time point $t = 30$, computed for time step size $\Delta t = 0.001$, which is well below the stability limit (4.26), and computed for time step size $\Delta t = 0.3$, which is chosen just above the stability limit. For $\Delta t = 0.3$, the onset of instabilities is clearly visible by the appearance of rapid oscillations on the solution, which confirms the analysis.



(a) Solution computed for $\Delta t = 0.001$

(b) Solution computed for $\Delta t = 0.3$

Figure 4.4: Solution at time point $t = 30$ computed with the Forward Euler scheme (a) for $\Delta t = 0.001$, chosen well below stability limit (4.26), and (b) for $\Delta t = 0.3$, chosen just above the stability limit. The most oscillatory mode is seen to be the most sensitive to instabilities.

4.5 Implicit discretisation

Application of the implicit Backward Euler scheme in combination with second-order central finite differences to the Laplacian in the isotropic continuum field model (2.37) results in the following system:

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^{n}}{\Delta t} = L\kappa \frac{\eta_{i_{r+1}}^{n+1} - 2\eta_{i_r}^{n+1} + \eta_{i_{r-1}}^{n+1}}{(\Delta x)^2}$$

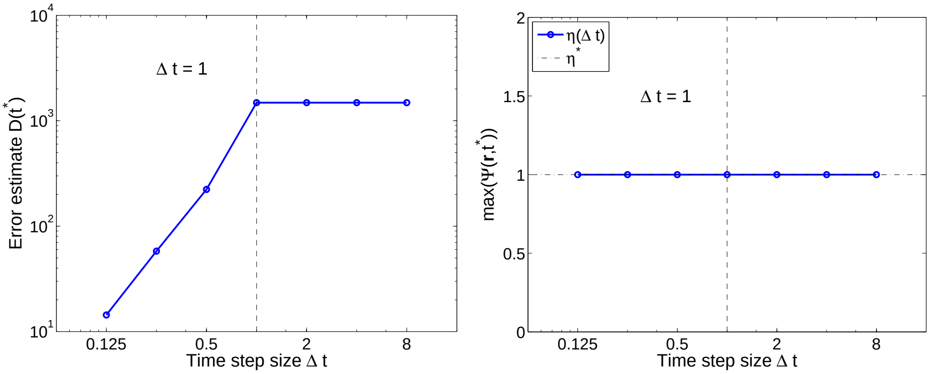
$$+ Lm \left(\left(\eta_{i_r}^{n+1}\right)^3 + \eta_{i_r}^{n+1} - 2\eta_{i_r}^{n+1} \sum_{j=1}^{p} \left(\eta_{j_r}^{n+1}\right)^2 \right),$$

$$r = 1, \dots, N, \qquad i = 1, \dots, p. \quad (4.29)$$

4.5.1 Analysis and numerical results

The addition of perturbation (4.8) to the equilibrium value $\eta_i(x) = \pm 1$ results in an amplification factor given by

$$\lambda(k) = \frac{1}{1 + \frac{4L\kappa\Delta t}{(\Delta x)^2} \sin^2\left(\pi k \frac{\Delta x}{2l}\right) + 2Lm\Delta t} . \quad (4.30)$$



(a) Error measure $D(t^*)$

(b) $\max(\Psi(\mathbf{r}, t^*))$

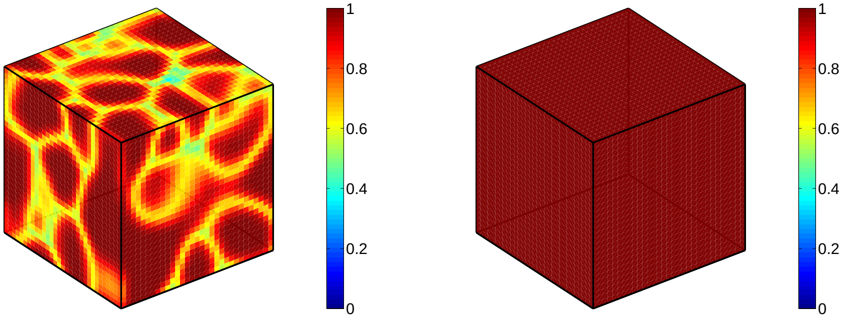
Figure 4.5: Error measure $D(t^*)$ and the maximal value of $\Psi(\mathbf{r}, t^*)$ as a function of the time step size Δt, obtained from numerical experiments with the Backward Euler scheme (4.29) for $t^* = 40$. The implicit scheme is stable for the large time step sizes, but significantly loses in accuracy for $\Delta t \geq 1$.

The value of $\lambda(k)$ always fulfils condition (4.9). The same perturbation, but now on the equilibrium value $\eta_i(x) = 0$, results in the amplification factor

$$\lambda(k) = \frac{1}{1 + \frac{4L\kappa\Delta t}{(\Delta x)^2}\sin^2\left(\pi k \frac{\Delta x}{2l}\right) + Lm\Delta t}. \tag{4.31}$$

Again, the value of $\lambda(k)$ always fulfils condition (4.9). Since both amplification factors derived above will never exceed the value 1 in absolute value, implicit discretisation scheme (4.29) is unconditionally stable.

To check the stability properties of the Backward Euler scheme with numerical experiments, we have implemented an implicit time stepper that combines Gauss-Seidel iterations with Newton steps. The experiments are run with the same parameters as listed in Section 4.4.2, but now for time step sizes ranging from $\Delta t = 0.125$ to $\Delta t = 8$. The graphs on Fig. 4.5(b) confirm that the Backward Euler scheme is stable for large time steps: the maximal value of $\Psi(\mathbf{r}, t^*)$ over the system domain Ω never exceeds the value 1. The logarithmic plot of Fig. 4.5(a), which shows the error measure $D(t^*)$ as a function of the time step size Δt, reflects that the Backward Euler scheme is only of first-order accuracy: the value of $D(t^*)$ increases linearly with Δt and reaches a limiting value for $\Delta t \approx 1$. For this time step size, at $t = 40$, the microstructure only consists of one grain, which is illustrated on Fig. 4.6(b). This indicates that for a large time step, the implicit time stepper does not become unstable, but significantly loses in accuracy.



(a) Solution computed with the Forward Euler scheme for $\Delta t = 0.001$

(b) Solution computed with the Backward Euler scheme for $\Delta t = 8$

Figure 4.6: Solution at time point $t = 40$ computed (a) with the Forward Euler scheme for $\Delta t = 0.001$, chosen well below stability limit (4.26), and (b) with the Backward Euler scheme for $\Delta t = 1$. No instabilities are detected during the computations with the implicit time stepper.

4.6 Semi-implicit discretisation

Application of the semi-implicit SBDF1 scheme in combination with second-order central finite differences to the Laplacian in the isotropic continuum field model (2.37) results in the following system:

$$\frac{\eta_{i_r}^{n+1} - \eta_{i_r}^{n}}{\Delta t} = L\kappa \frac{\eta_{i_{r+1}}^{n+1} - 2\eta_{i_r}^{n+1} + \eta_{i_{r-1}}^{n+1}}{(\Delta x)^2} + Lm \left(\left(\eta_{i_r}^{n}\right)^3 + \eta_{i_r}^{n} - 2\eta_{i_r}^{n} \sum_{j=1}^{p} \left(\eta_{j_r}^{n}\right)^2 \right),$$

$$r = 1, \ldots, N, \qquad i = 1, \ldots, p. \quad (4.32)$$

This differentiation scheme corresponds to system (3.12) in Section 3.2.1.

4.6.1 Analysis

A perturbation of the form (4.8) on the equilibrium value $\eta_i(x) = \pm 1$, substituted into scheme (4.32) yields the equations

$$\frac{V_r^{n+1} - V_r^n}{\Delta t} = L\kappa \left(\frac{V_{r+1}^n - 2V_r^n + V_{r-1}^n}{(\Delta x)^2} \right) + Lm \left((1 + V_r^n)^3 + (1 + V_r^n) - 2(1 + V_r^n)^3 \right), \qquad r = 1, \ldots, N. \tag{4.33}$$

Again, we neglect the higher-order terms in ϵ in (4.33). Using equality (4.18c), we obtain

$$\frac{V_r^{n+1} - V_r^n}{\Delta t} = -\frac{4L\kappa}{(\Delta x)^2} \sin^2 \left(\pi k \frac{\Delta x}{2l} \right) V_r^{n+1} - 2LmV_r^n \tag{4.34}$$

and amplification factor

$$\lambda(k) = \frac{1 - 2Lm\Delta t}{1 + \frac{4L\kappa\Delta t}{(\Delta x)^2} \sin^2 \left(\pi k \frac{\Delta x}{2l} \right)} \ . \tag{4.35}$$

Figure 4.7 plots the value of $\lambda(k)$ as a function of the wavenumber k for different values of the time step size Δt. It can be seen that the condition $\lambda(k) < 1$ is automatically fulfilled by (4.35). As illustrated on Fig. 4.7, the modes that are the most sensitive to instabilities, for which $\lambda(k)$ will fall below the value -1, are the modes with wave numbers close to $k = 0$. For the latter wavenumber, the amplification factor becomes

$$\lambda(0) = 1 - 2Lm\Delta t. \tag{4.36}$$

In order to fulfil $-1 < \lambda(0)$, the time step size Δt thus has to obey the inequality

$$\Delta t < \frac{1}{mL}, \tag{4.37}$$

Similarly, perturbation of the equilibrium value $\eta_i(x) = 0$ yields stability condition

$$\Delta t < \frac{2}{Lm} \ . \tag{4.38}$$

Conditions (4.37) and (4.38) correspond to the stability conditions derived in Section 4.3.2 for the reaction part of the continuum field model and are independent of the dimensions of the system. Since stability condition (4.38) is less strict than condition (4.37), the stability properties of the SBDF1 scheme are determined by the latter condition.

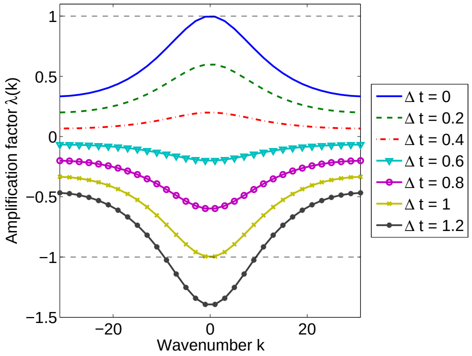


Figure 4.7: Amplification factor $\lambda(k)$ (4.35) obtained for the SBDF1 scheme (4.32) as a function of the wavenumber k, for different values of the time step size Δt.

4.6.2 Numerical results

The same numerical experiments as in Section 4.4.2 are performed to test stability condition (4.37), but now with a time stepper based on scheme (4.32). The expected stability limit for the time step size according to (4.37) is

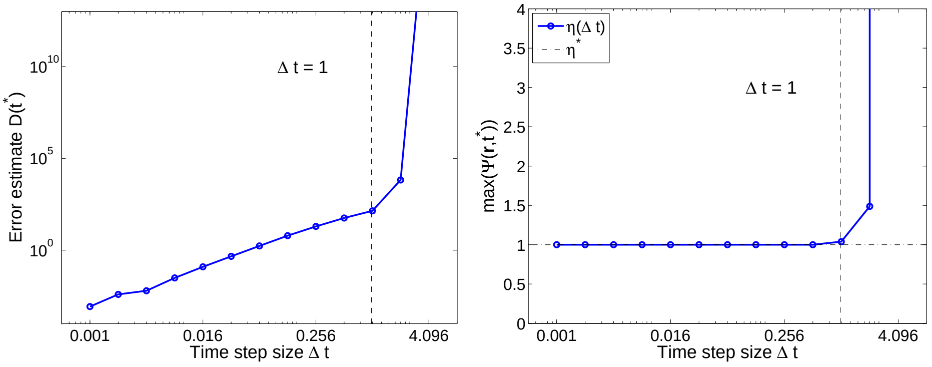
$$\Delta t < \frac{1}{mL} = 1 \ . \tag{4.39}$$

Figure 4.8(a) shows the error measure $D(t^*)$ as a function of the time step size Δt, as well as the theoretically expected stability limit (4.39). The graph of the error measure shows an increase of the slope at approximately $\Delta t = 1$. On Fig. 4.8(b), the maximal value of $\Psi(\mathbf{r}, t^*)$ also starts to rise at approximately $\Delta t = 1$, which corresponds to the onset of instabilities in the computations. The numerical experiments thus confirm the theoretically derived stability condition (4.37).

Figure 4.9 shows the solution at time point $t = 31.5$ as computed with the explicit time stepper for time step size $\Delta t = 0.001$, chosen well below the stability limit (4.26), and for the semi-implicit time stepper with time step size $\Delta t = 1.05$, just above the expected stability limit (4.39) of the SBDF1 scheme. For time step size $\Delta t = 1.05$, instabilities are visible under the form of slow oscillations.

4.7 Conclusion

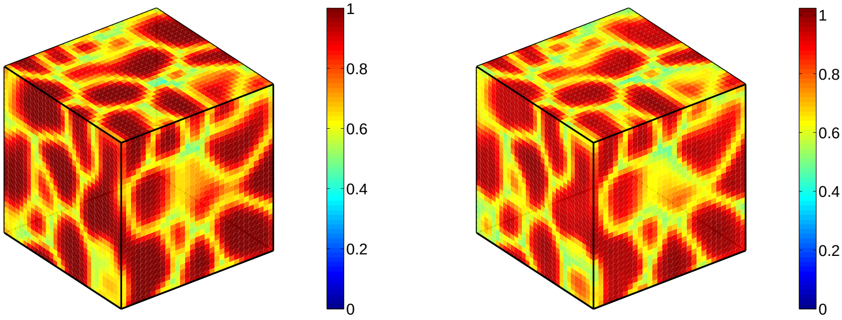
This chapter started with a brief study of the stability properties of the semi-discretised continuum field model. Subsequently, the fully discretised equations of continuum field model (2.37) are examined. The latter model consists of two



(a) Error measure $D(t^*)$

(b) $\max(\Psi(\mathbf{r}, t^*))$

Figure 4.8: Error measure $D(t^*)$ and the maximal value of $\Psi(\mathbf{r}, t^*)$ as a function of the time step size Δt, obtained from numerical experiments with the SBDF1 scheme (4.32) for $t^* = 40$, compared with the theoretically derived stability limit.



(a) Solution computed with the Forward Euler scheme for $\Delta t = 0.001$

(b) Solution computed with the SBDF1 scheme for $\Delta t = 1.05$

Figure 4.9: Solution at time point $t = 31.5$ computed (a) with the Forward Euler scheme for $\Delta t = 0.001$, well below stability limit (4.23), and (b) with the SBDF1 scheme for $\Delta t = 1.05$, just above stability limit (4.37). Slow oscillatory modes are seen to be the most sensitive to instabilities.

parts: a diffusion part and a reaction part. We have analysed the influence of both parts separately on the stability properties of the explicit Forward Euler time discretisation scheme. For this purpose, a theoretical stability analysis is performed by studying the amplification of Fourier modes. The analysis of the diffusion part is done in accordance with the approach described in [93]. Because of the presence of nonlinear terms, a different approach is followed for the analysis of the reaction part: the effect of a perturbation, which consists of scaled Fourier modes, on the equilibrium values of the system is studied as a function of time step size Δt. A comparison of the stability conditions thus obtained shows that the diffusion part is the most restrictive part for the time step size in the case of explicit time discretisation.

Following a similar approach as for the diffusion part and the reaction part separately, stability conditions are derived for the explicit Forward Euler scheme, the implicit Backward Euler scheme, and the semi-implicit SBDF1 scheme, now applied to the full continuum field model. For three-dimensional systems, the stability condition of the Forward Euler scheme is found to be

$$\Delta t < \frac{1}{\frac{6L\kappa}{(\Delta x)^2} + mL} \ . \tag{4.40}$$

The stability condition obtained for the SBDF1 scheme is

$$\Delta t < \frac{1}{mL} \ . \tag{4.41}$$

In contrast to condition (4.40), this inequality does not include an explicit dependency on the grid spacing Δx. However, the parameter m is related to Δx for accuracy reasons. The thickness of a grain boundary that is locally in equilibrium is proportional to $\sqrt{\kappa/m}$ [85]. A larger value of m thus corresponds to a smaller boundary thickness, which in turn asks for a smaller grid spacing Δx. With this relation in mind, it can be seen that the parameters m and Δx supplement each other in stability condition (4.40) for the explicit Forward Euler scheme. The implicit Backward Euler scheme is found to be unconditionally stable. All theoretically derived stability conditions are confirmed by numerical experiments.

While explicit time steppers are generally very efficient in their computations, the stability condition of the Forward Euler scheme contains a dependency on the square of the grid spacing Δx. For finer spatial discretisations, the time step size will therefore soon become infeasibly small. In contrast, the implicit Backward Euler scheme is unconditionally stable. However, simulation with this implicit scheme involves the solution of a single large nonlinear system. Also, since the Backward Euler scheme is only of first-order accuracy, the possibility of a large time step is restricted by accuracy concerns. The semi-implicit SBDF1 scheme only has a weak stability condition, and decouples the system of partial differential equations into several smaller systems that are relatively easy to solve and can be solved simultaneously. When an efficient solver is used to solve these smaller systems, the SBDF1 time discretisation scheme is therefore the best choice for the continuum field model.

Chapter 5

Bounding box algorithm

5.1 Introduction

As discussed in Chapter 3, several algorithms have been designed to overcome the computational limitations of the phase field method. In [129, 130], we presented a bounding box method based on the same observations that led to the sparse algorithms of [45, 133, 65] (see Section 3.3), but now for semi-implicit time integration. It computes the phase field model only within cuboid hulls of so-called grain regions. In contrast with the algorithms described above, it is appropriate for implicit and semi-implicit time stepping schemes. The algorithm was designed in the context of the study of grain growth in the presence of second-phase particles according to the model of [25, 86]. Thanks to its object-oriented design, the algorithm can easily be extended to more complex phase field models, for example for grain growth in materials which show orientation dependence. Moreover, the object-oriented approach has definite advantages in post-processing. In [129], the bounding box algorithm was implemented for the solution of model equations (2.39), but it can naturally be applied to other models as well.

The structure of this chapter is as follows. In Section 5.2, the concepts of the bounding box algorithm are explained, as well as its sparse data structure and the simulation algorithm. The simulation conditions of a test case are specified in Section 5.3, after which the characteristics of the algorithm are discussed. The computational requirements of our implementation of the algorithm are examined in Section 5.4. Section 5.5 presents a parallel implementation of the algorithm, and in Section 5.6, the extension of the bounding box data structure to microstructures with anisotropic boundary properties is explained. Some concluding remarks are formulated in Section 5.7.

5.2 Bounding box algorithm

5.2.1 Concepts

A phase field variable η_i is defined to be *active* at a grid point $\mathbf{r}$ at time point t when its absolute value exceeds a small positive threshold value ϵ:

$$|\eta_i(\mathbf{r}, t)| > \epsilon. \tag{5.1}$$

At a given time point, at every grid point, only a few phase field variables are active and thus contribute to the evolution of the microstructure: inside each grain, away from its boundaries, one phase field variable, η_i, is active and near grain boundaries, only those phase field variables corresponding to the neighbouring grains, are active.

We now define a *grain region* $G_i(t)$ as the set of ordered couples $(\mathbf{r}, \eta_i(\mathbf{r}, t))$ with grid points $\mathbf{r}$ that are connected, and with $|\eta_i(\mathbf{r}, t)| > \epsilon$:

$$G_i(t) = \{(\mathbf{r}, \eta_i(\mathbf{r}, t)) : (|\eta_i(\mathbf{r}, t)| > \epsilon) \ \wedge \ (\mathbf{r} \text{ are connected})\}. \tag{5.2}$$

To take periodic boundary conditions into account, the grain regions are allowed to wrap around the grid boundaries. Also, different grain regions, corresponding to different phase field variables, can overlap, which allows them to interact.

For every grain region $G_i(t)$, a *bounding box* $\mathcal{B}_i(t)$ is established as the smallest cuboid grid part enclosing its grid points $\mathbf{r}$. The bounding box $\mathcal{B}_i(t)$ is completely characterised by the coordinates of two opposite delimiting grid points as

$$\mathcal{B}_i(t) = (\mathbf{r}_{i,\min}(t), \mathbf{r}_{i,\max}(t)), \tag{5.3}$$

with

$$\mathbf{r}_{i,\min}(t) = (\min(x), \min(y), \min(z)) \tag{5.4}$$

and

$$\mathbf{r}_{i,\max}(t) = (\max(x), \max(y), \max(z))\,, \tag{5.5}$$

both taken over all grid points $\mathbf{r}(x, y, z)$ with $(\mathbf{r}, \eta_i(\mathbf{r}, t)) \in G_i(t)$.

Finally, we define the set $B_i(t)$ as

$$B_i(t) = \{(\mathbf{r}, \eta_i(\mathbf{r}, t)) : \mathbf{r} \text{ lies within } \mathcal{B}_i(t)\}. \tag{5.6}$$

As the values of the phase field variables $\eta_i(\mathbf{r}, t)$ evolve with time, the grain regions $G_i(t)$, the bounding boxes $\mathcal{B}_i(t)$, and the sets $B_i(t)$ evolve accordingly.

5.2.2 Data structure

To implement the bounding box data structure, we followed the approach of object-oriented programming. The microstructure is thus treated as a class with a number of attributes: the time point, the grid size and, if present in the model, second-phase particles or other features of the polycrystalline microstructure. A grain region in turn is also treated as a class. The attributes of a grain region are the model parameters associated with each η_i, namely κ and L, the bounding box delimiters, the crystallographic orientation type corresponding to η_i represented by the index i, and the set B_i.

Figure 5.1 shows a Unified Modelling Language (UML) diagram of the object-oriented data structure. UML is widely used to visualise object-oriented systems. In the bounding box data structure, the microstructure is represented by the class Microstructure; a grain region by the class GrainRegion. The line connecting the elements of the data structure in Fig. 5.1 indicates their association relationship. The notation at each end of the line indicates the multiplicity, which is the number of classes that participate in the association.

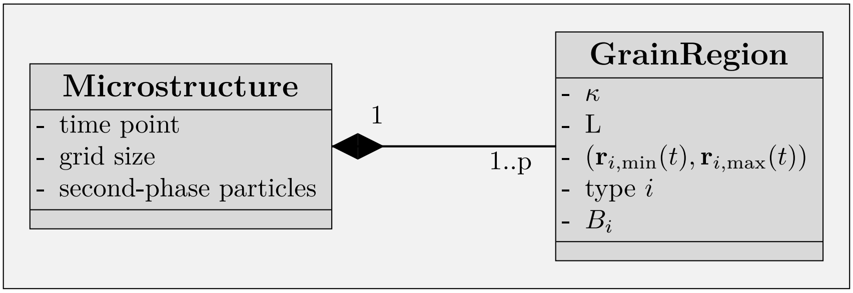


Figure 5.1: UML diagram of the object-oriented data structure composed during initialisation of the bounding box algorithm. The microstructure and grain regions are treated as classes with a number of attributes.

The sparse 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.

5.2.3 Initialisation of the data structure

To initialise the bounding box algorithm, a polycrystalline microstructure is required. This initial microstructure can be obtained for example from microscopic images, from other simulations, or by the construction of for example a Voronoi tessellation. We will describe two initialisation approaches in detail. The first approach obtains the sparse bounding box data structure by processing data available from a previous phase field simulation on a full grid. The second approach constructs the data structure by generating nuclei in accordance with a uniform distribution over the grid.

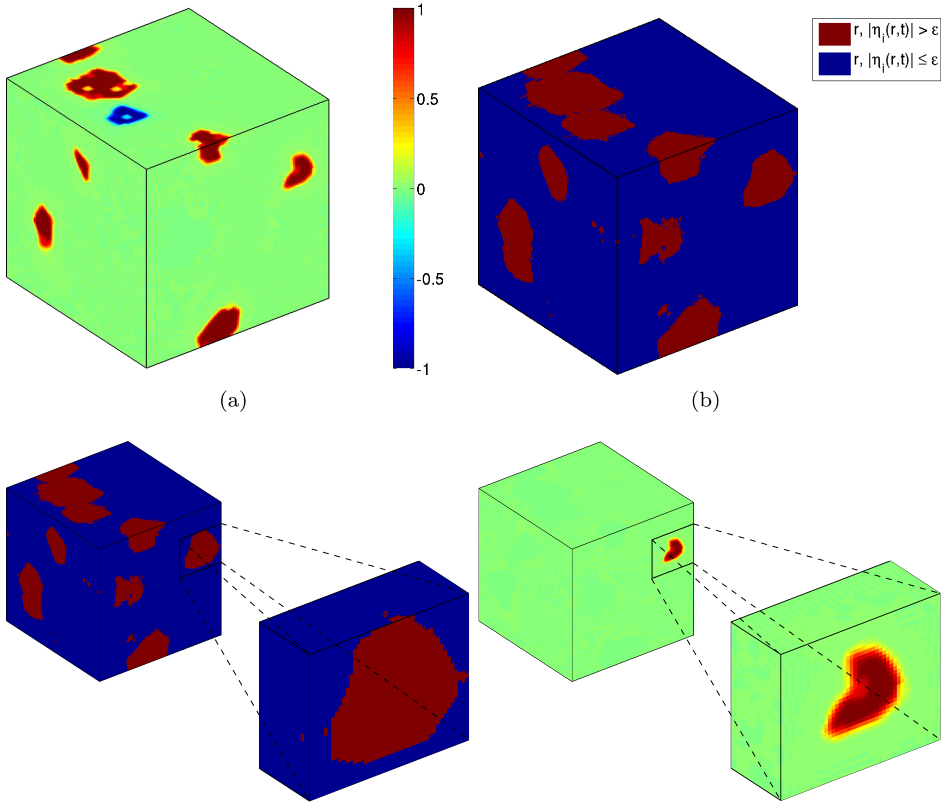
Initialisation from previous phase field simulation

Consider the set of phase field variables $\hat{\eta}_l(\mathbf{r})$ for $l = 1, \ldots, q$, containing data from a phase field simulation of grain growth on a full grid. The values of each phase field variable vary between the two equilibrium values -1 and 1, as illustrated in Fig. 5.2(a). In a full phase field simulation, it is possible that several n_l grains possess the same crystallographic orientation and are thus represented by the same phase field variable $\hat{\eta}_l$. We will identify these grains by a sequence number k that ranges from 1 to n_l, with n_l the number of grain regions associated with one phase field variable $\hat{\eta}_l$.

The initialisation of the bounding box data structure starts with several preprocessing steps. First, the threshold value ϵ is applied to the phase field variables and for every involved phase field variable $\hat{\eta}_l$, all grain regions $\hat{G}_{l,k}$, with $k = 1, \ldots, n_l$ are located (see Fig. 5.2(b)). Second, the corresponding bounding boxes $\hat{\mathcal{B}}_{l,k}$ and the sets $\hat{B}_{l,k}$ are established, as illustrated in Fig. 5.2(c). Third, a grain region renumbering takes place and a new set of phase field variables is introduced, such that there is a one-to-one mapping between grain regions and phase field variables. More precisely, the grain regions are renumbered such that every grain region $\hat{G}_{l,k}$ for $l = 1, \ldots, q$ and $k = 1, \ldots, n_l$ is given a unique index i, ranging from 1 to p, the total number of grain regions, with $p = \sum_{l=1}^{q} n_l$. That is, grain region $\hat{G}_{l,k}$ is rewritten as G_i, a grain region corresponding to one of the new phase field variables, η_i, for $i = 1, \ldots, p$, with bounding box information copied from $\hat{B}_{l,k}$ to B_i. Figure 5.2(d) illustrates the mapping of one isolated grain region to a single phase field variable η_i. The renumbering of grain regions and phase field variables is only allowed if the model parameters associated with $\hat{\eta}_l$ are correctly copied to the new set of phase field variables η_i.

If second-phase particles are present in the input data structure, the additional parameter Φ, which represents the particles (see Section 2.5.2), is processed to locate all particles. An array is then constructed that contains the linearised coordinates of all grid points occupied by a second-phase particle. In this way, the parameter Φ is implicitly represented.

After the preprocessing of the phase field variables, the bounding box data structure is set up. For every grain region G_i, a GrainRegion-object is instantiated. Also, a Microstructure-object is created and the links between the Microstructure-



(a)

(b)

(c)

(d)

Figure 5.2: Locating the grain regions and determining the bounding boxes during initialisation of the bounding box data structure, when the initial microstructure is obtained from a previous phase field simulation. (a) The values of one phase field variable, $\hat{\eta}_l$, vary between the two equilibrium values -1 and 1. (b) The threshold value ϵ is applied and the grain regions $\hat{G}_{l,k}$ are located. (c) The bounding boxes $\hat{\mathcal{B}}_{l,k}$ and the sets $\hat{B}_{l,k}$ are established. (d) A new phase field variable η_i is assigned to every $\hat{G}_{l,k}$ and the values inside the corresponding bounding box $\hat{\mathcal{B}}_{l,k}$ are isolated.

object and the GrainRegion-objects are established. The preprocessing procedure, followed by the data structure set-up is summarised in Algorithm 5.1.

Algorithm 5.1: Preprocessing algorithm that constructs a sparse bounding box data structure from data resulting from a previous phase field simulation on a full grid.

```
Input: Set of phase field variables $\hat{\eta}_l$ with $l = 1, \ldots, q$, defined on a full grid
       and, if second-phase particles are present, parameter $\Phi$
Output: Bounding box data structure
for $l = 1$ to $q$ do
    Find all grain regions $\hat{G}_{l,k}$;
    for $k = 1$ to $n_l$ do
        Determine bounding box delimiters and $\hat{B}_{l,k}$;
    end
end
Compute $p = \sum_{l=1}^{q} n_l$;
foreach grain region $\hat{G}_{l,k}$ do
    Rename $\hat{G}_{l,k}$ to $G_i$, corresponding to a unique phase field variable $\eta_i$;
    Copy the bounding box information from $\hat{B}_{l,k}$ to $B_i$;
end
if second-phase particles are present then
    Process the parameter $\Phi$ to find the coordinates of all particles;
    Construct an array that contains the linearised coordinates of all
    particles;
end
Instantiate Microstructure-object;
Instantiate GrainRegion-object for every grain region $G_i$;
Set up bounding box data structure;
```

The assignment of a unique phase field variable to every grain region ensures coalescence-free simulations with the bounding box algorithm. In the situation where grains can split into subgrains (e.g. recovery after deformation), the renumbering procedure can be reexecuted to guarantee a one-to-one phase field variable to grain region relation.

We have implemented Algorithm 5.1 as a MATLAB-function, which saves the resulting bounding box data structure in a dedicated data format.

Initialisation by random generation of nuclei

A second possible technique to initialise a microstructure is by spreading a set of spherical grain nuclei over the domain, together with a distribution of second-

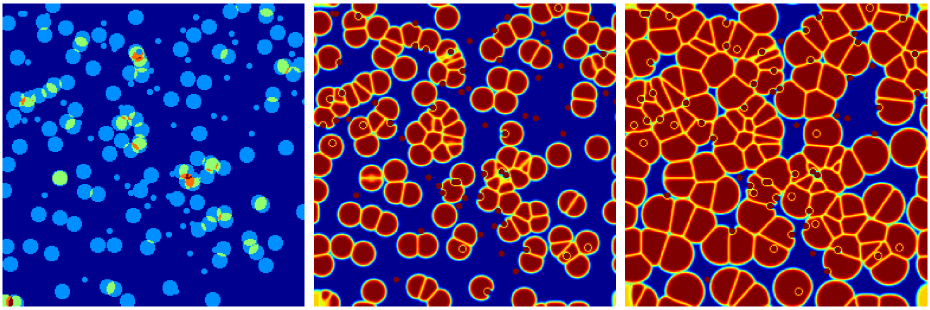


Figure 5.3: Initialisation by generating grain nuclei and circular second-phase particles according to a uniform distribution over the domain. (a) Initial state: some grain nuclei and/or particles may overlap. (b/c) Microstructure evolution: all grain nuclei grow and impinge. Initially overlapping nuclei become neighbouring grains.

phase particles. This initialisation method somewhat resembles the nucleation of new grains during recrystallisation (see Chapter 1).

First, for every phase field variable η_i, with $i = 1, \ldots, p$, a grid point C_i is chosen according to a specified distribution over the domain of the microstructure. Depending on the application, a different distribution function can be used. In this work, we will always assume a uniform distribution of the grain nuclei. Second, a spherical grain region G_i with a small radius is created with C_i as its centre. The phase field variable η_i is now initialised such that η_i equals 1 inside the grain region G_i and 0 outside the grain region. Third, for every generated grain region, the corresponding bounding box and the set B_i are determined. No attention is paid to the fact that some nuclei may overlap. After an initial transitional period, the overlapping nuclei will evolve to neighbouring grains, as shown in Fig. 5.3 for a two-dimensional microstructure.

In this work, the second-phase particles are assumed to have a spheroid shape. A spheroid shape is obtained by rotating an ellipse about one of its axes. It is characterised by the ratio $r_a = c/a$ of its two axis radii a and c, illustrated in Fig. 5.4. The second-phase particles are initialised similarly to the grains. Until a desired volume fraction f_V of particles is obtained, new particles are added to the microstructure. The volume fraction of the second-phase particles f_V is defined as the number of grid points representing the second-phase particles, divided by the total number of grid points in the system. For each particle, first, a grid point D is chosen according to a uniform distribution over the system domain as the centre of the new particle. Second, the particle orientation, which is the orientation of the c-axis, is generated according to a desired distribution function. Third, the coordinates of a spheroid with centre D and axes according to the generated orientation are computed. Next, the linearised coordinates of the spheroid are added to an array that contains the linearised coordinates of all grid points occupied by a second-phase particle. No effort is made to separate the second-phase particles, since particle clusters also occur in real materials. Some particles may therefore overlap. As in [4], we assume that the influence of this small amount of particle clusters on the pinning effect is small. Finally, as in Algorithm 5.1, the bounding box data structure is set up. Algorithm 5.2 summarises the initialisation procedure described above.

Algorithm 5.2: Initialisation by generating nuclei over the microstructure.

```
Input: Number of phase field variables $p$ and volume fraction $f_{\mathrm{V}}$
Output: Bounding box data structure
for $i = 1$ to $p$ do
    Choose grid point $C_i$ according to a uniform distribution over the grid;
    Generate a grain region $G_i$ with a small radius and centre $C_i$;
    Set $\eta_i$ to 1 inside $G_i$ and to 0 outside $G_i$;
    Determine the bounding box delimiters of $G_i$, and the set $B_i$;
end
if $f_V > 0$ then
    while desired volume fraction $f_V$ is not reached do
        Choose grid point $D$ according to a uniform distribution over the
        grid;
        Generate particle orientation;
        Compute the coordinates of a spheroid particle with centre $D$ and
        axes according to the orientation;
        Add the linearised coordinates of grid points occupied by the particle
        to an array representing the particles;
    end
end
Instantiate Microstructure-object;
Instantiate GrainRegion-object for every grain region $G_i$;
Set up bounding box data structure;
```

5.2.4 Simulation

In a grain growth simulation, at any given time point, the phase field variables are only locally active, inside the grain regions. Outside the grain regions, the phase field values are assumed to be zero. The bounding box algorithm exploits this property by solving the model equations only inside the bounding boxes, i.e. for the phase field values included in the sets B_i.

At each time point t, for every set B_i, the model equations are solved. The

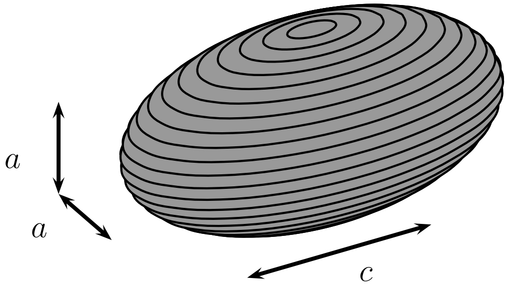


Figure 5.4: In this work, the second-phase particles are modelled as spheroids. They are characterised by two axis radii a and c.

bounding boxes of the different grain regions overlap and the phase field variables interact within the regions of overlap. Depending on the computed updates of η_i during one time step Δt, it is possible that grain region G_i and correspondingly the bounding box $\mathcal{B}_i$ and the set B_i grow or shrink. To anticipate this change, the calculation of each time step is preceded by a preparation of $\mathcal{B}_i$. In each grid dimension, a margin of M grid points is added to $\mathcal{B}_i$. The delimiters of $\mathcal{B}_i$ are thus shifted as

$$\mathbf{r}'_{i,\min} = \mathbf{r}_{i,\min} - M \cdot (1,1,1) \tag{5.7}$$

and

$$\mathbf{r}'_{i,\max} = \mathbf{r}_{i,\max} + M \cdot (1,1,1). \tag{5.8}$$

To take into account the periodic boundary conditions of model (2.39), the enlarged bounding box is allowed to wrap around the grid boundaries. However, this preparatory step has to ensure that the dimensions of a single bounding box can never exceed the dimensions of the full grid. The newly added margin is then filled with zeros, which is in accordance with the assumption that phase field values outside G_i are equal to zero. Note that the margin allows a grain region to grow inside its bounding box during one time step, but not for more than M grid points at once in each dimension.

After the preparatory step, the model equations are solved for each B_i separately by locally applying homogeneous Dirichlet conditions. Possible choices for the solution method are iterative methods such as multigrid methods (see Chapter 6), the successive overrelaxation (SOR) method, or the Gauss-Seidel method. Note however that the bounding boxes have different sizes. Therefore, not every solution method will attain its maximal efficiency, e.g. solvers based on Fourier transforms will lose speed. In [129], the Gauss-Seidel method is used. When the new values of B_i are computed, the algorithm checks whether the grain region G_i has shrunk or grown and adjusts the bounding box delimiters and the set B_i accordingly.

A simulation with the bounding box algorithm thus proceeds as described in Algorithm 5.3.

Algorithm 5.3: Bounding box algorithm

```
Input: Bounding box data structure at $t_1$; $t_{\text{end}}$, $\epsilon$ and $M$
Output: Bounding box data structure at $t_{\text{end}}$
for $t = t_1, t_2, \ldots, t_{end}$ do
    for $i = 1$ to $p$ do
        Add a margin of $M$ grid points in each grid direction to $B_i$;
        Fill the margin with zeros;
        Solve the discretised model equations for $B_i$;
        Check whether $G_i$ has grown or shrunk and update the bounding
        box delimiters and $B_i$ accordingly;
    end
end
```

5.2.5 Remarks

We chose to restrict the computation and the storage of the phase field variables during a grain growth simulation to the grid points contained by the cuboid hulls of the grain regions. The cuboid shape of the bounding boxes is compatible with the structure of the rectangular grids that are used throughout this work. A bounding box can therefore be described by the use of only two delimiting grid points, located in two opposing corners. The overhead to represent the boxes in the data structure is therefore very small. Furthermore, the cuboid shape allows for implicit time stepping: the systems that result from the application of implicit time integration schemes can be used as the input for any iterative solver without further mediation. The choice to limit the computation and the storage to cuboid bounding boxes thus leads to a manageable data structure.

To a certain extent, the bounding box algorithm tracks the movement of the grain boundaries. However, it does not have the same difficulties that front-tracking models can meet (see Section 2.3.3). The choice of the cuboid shape of the bounding boxes does not make any prior assumptions on the shape of the grain regions. Furthermore, the possible changes that occur in the data structure during a grain growth simulation, such as the disappearance of a grain, are much more straightforward to cope with than the critical events a front-tracking model has to be able to handle.

The parameter M, which measures the margin that is added to the bounding boxes at the start of each time step, is related to the time step size Δt. For a larger Δt, the grain regions will grow more during one time step. In this work, however, we have not tested the interplay between these two parameters.

The implementation of the bounding box algorithm used in [129, 130] employs the Gauss-Seidel method to solve the phase field equations, which are discretised with the 1SBDF scheme. Convergence is reached when the norm of the residual divided by the norm of the right-hand side falls below a user imposed tolerance value. In these works, a time step size is used that is rather below the stability condition found in Section 4.6. Because of this choice of time step size, a higher accuracy is obtained and only a few iterations of the Gauss-Seidel method are necessary to solve the equations.

Section 5.2.4 describes how the model equations are solved for each set B_i separately by locally applying homogeneous Dirichlet conditions. However, we have also implemented the bounding box algorithm in combination with solver scheme (3.23), assuming that the sets B_i can be solved with periodic boundary conditions, using the FFTW library [39]. Numerical experiments with this implementation show that the assumption of periodic boundary conditions is correct and yields accurate results. Unfortunately, because all bounding boxes have a different size, the efficiency of the FFTW library cannot be exploited to the fullest.

5.3 Characteristics of the bounding box algorithm

5.3.1 Test case

In this section, we test the bounding box algorithm using model (2.39). We compare the new algorithm with a parallel implementation of an algorithm solving the equations (2.39) globally, based on the semi-implicit finite difference scheme proposed in [28]. The solution scheme of this parallel implementation is described by formula (3.23) in Section 3.4.2. This algorithm and its characteristics will hereafter be referred to as the *conventional algorithm*, with *conventional* characteristics.

To obtain an initial polycrystalline microstructure to set up the bounding box data structure, simulations with the conventional algorithm were run on a equispaced $256 \times 256 \times 256$ grid. Because of memory limitations, only 100 phase field variables were employed, whereas according to [74], more than 200 phase field variables are required to prevent grain coalescence in a three-dimensional simulation. The parameter values were set to $\kappa = 0.5$, $L = 1$ and $m = 1$, and no second-phase particles were included. The discretisation spacings were $\Delta x = 1$ 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 bounding box data structure was initialised with preprocessing Algorithm 5.1. For the subsequent bounding box simulations, the parameter M was chosen equal to 1.

The bounding box algorithm now depends on two parameters: the applied threshold value ϵ and the initial mean grain size. The latter is determined by the time point t_s at which the conventional simulation is stopped. The parameters ϵ and the mean grain size influence the accuracy, computing memory and computing time of the bounding box algorithm.

5.3.2 Bounding box data structure

During the initialisation of the bounding box data structure, the active regions of the phase field variables are identified and isolated. From this point on, all data storage is thus restricted to the phase field values contained by the bounding boxes. 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 active phase field variables per grid point and equal the number of phase field values included in the sets B_i. This number depends on ϵ and the initial mean grain size and, as will be shown below, is typically a small fraction of what is required for the conventional implementation.

The initialisation procedure described by Algorithm 5.1 was performed on the simulation results specified in Section 5.3 at time points $t_s = 200$, 400, 600, and 800. Different values of ϵ were applied: 10^{-3}, 10^{-4}, 10^{-5}, and 10^{-6}. Table 5.1 shows the required number of phase field values per grid point for the bounding box algorithm and the conventional algorithm as a function of ϵ and t_s and thus the initial mean grain size. The fine-grained topology of the microstructure at the initial evolution stages results in smaller initial mean grain sizes and larger numbers of required phase field values. It can be seen that the bounding box algorithm is far more efficient than the conventional algorithm.

5.3.3 Bounding box simulation

The bounding box algorithm only solves equations (2.39) for the grid points within the sets B_i. A large amount of computing time and computing memory is thus saved. During a simulation, the topology of a microstructure becomes more and more coarse-grained. While the memory demand of the conventional algorithm

Table 5.1: Number of phase field values required per grid point for the bounding box algorithm and the conventional algorithm as a function of ϵ and the initial mean grain size (grid points).

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

Table 5.2: Number of phase field values per grid point for the bounding box algorithm and the conventional algorithm as a function of ϵ and the simulation time.

	threshold value ϵ				
simulation time t	10^{-3}	10^{-4}	10^{-5}	10^{-6}	conventional
t_s	5.75	6.48	7.26	8.10	100
$t_s + 200$	6.71	6.74	6.90	6.90	100
$t_s + 400$	6.72	6.73	6.81	6.81	100
$t_s + 600$	6.68	6.69	6.73	6.73	100
$t_s + 800$	6.64	6.65	6.68	6.68	100

is independent of the topology, the bounding box algorithm is more efficient for coarser-grained topologies.

To study the efficiency of the bounding box computations, a conventional simulation was performed on a $256 \times 256 \times 256$ grid until simulation time $t = 4600$, where the grain topology was clearly defined. To make sure that no further coalescence would occur in the conventional simulation, a phase field variable reassignment was performed. The resulting microstructure contained 67 grains, with a mean grain size of 2.3×10^5 grid points and a different crystallographic orientation for every grain. From time point $t_s = 4600$ on, both the conventional algorithm and the bounding box algorithm were applied with the described microstructure as initial state. Table 5.2 shows the time evolution of the mean number of phase field values per grid point for both algorithms for different threshold values ϵ. It can be seen that the memory efficiency of the bounding box algorithm increases with the simulation time for the smallest values of ϵ. Also, in the course of the simulation, the memory efficiencies for the different threshold values converge. This indicates that a lower threshold value ϵ only increases the computational requirements considerably at the beginning.

Next, we compare the simulation accuracy of the bounding box algorithm with that of the conventional algorithm. Since the microstructure, described in the previous paragraph, has a different crystallographic orientation for every grain, it allows for coalescence-free simulations with the conventional algorithm. A common way to study microstructural evolution, is by looking at the function Ψ, which is defined as

$$\Psi(\mathbf{r},t) = \sum_{i=1}^{p} \eta_i^2(\mathbf{r},t). \tag{5.9}$$

Function $\Psi(\mathbf{r},t)$ can be calculated during a bounding box simulation, only taking into account the values inside the bounding boxes. It takes the value 1 inside grains and considerably smaller values at grain boundaries.

The simulation accuracy of the bounding box algorithm is now studied by looking at the differences between the results of the conventional algorithm and those of the bounding box algorithm for different threshold values $\epsilon = 10^{-3}$, 10^{-4}, 10^{-5}, and 10^{-6}, after 1, 1000, and 7000 time steps. At these three time points, the differences between the results for function $\Psi(\mathbf{r},t)$ at every grid point, as well as the mean grain size, are computed and compared. After one time step, the point-wise differences between the results for the threshold values $\epsilon = 10^{-3}$, 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. After 1000 time steps, the point-wise differences are only of order 10^{-3}. The mean grain size obtained by the bounding box algorithm deviates less than 10^{-7}, relatively, from the size obtained by the conventional algorithm. The differences between the results obtained for the different threshold values ϵ 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 for the different threshold values differ by less than 10^{-7}, relatively. The relative differences between the mean grain sizes computed by the bounding box algorithm and the conventional algorithm have increased now, but are still only of order 10^{-5}. This shows that the bounding box algorithm is highly accurate. Furthermore, the accuracy of the bounding box computations is little influenced by the magnitude of the threshold value ϵ.

Finally, we study the effect of the threshold value ϵ and the time point t_s on the initial configuration. When the initial microstructure is obtained by simulation, the value of ϵ can influence the initial configuration. If 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. Consider a small positive threshold value ϵ. In the time steps before two neighbouring grains with the same crystallographic orientation coalesce, the values of the involved phase field variable start to increase along the contact boundary and eventually exceed ϵ in magnitude. At that point, the grains have become one according to the definition in Section 5.2 and will also be considered as one grain by preprocessing

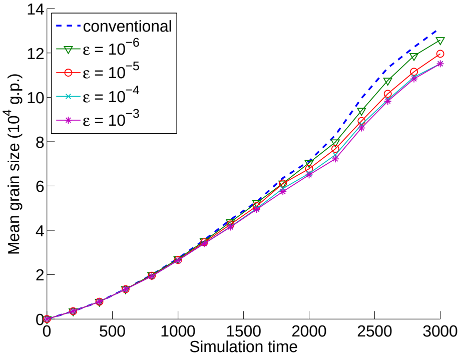


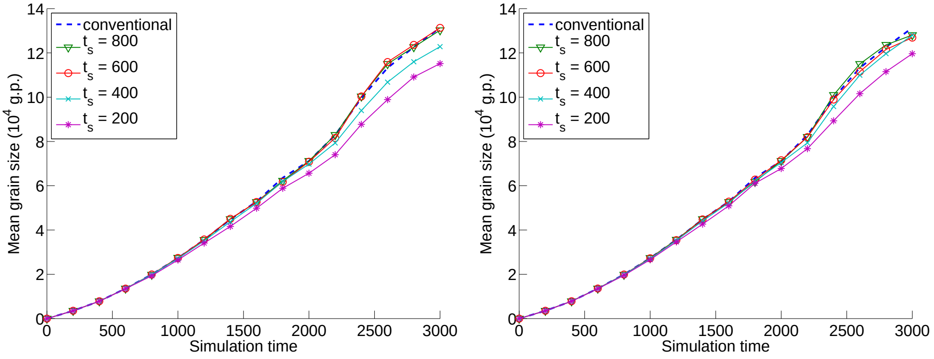
Figure 5.5: Effect of ϵ on the initial configuration: evolution of the mean grain size, measured in grid points (g.p.), from $t_s = 200$ to $t = 3000$, computed by the conventional algorithm and by the bounding box algorithm for different threshold values ϵ.

Algorithm 5.1. As a result, fewer different grains are taken into account and the initial mean grain size is larger. By using high values of ϵ, the neighbouring grains will not be treated as one, but as separate grains. The use of too high threshold values will however disturb the accuracy of the simulation results.

To investigate the influence of ϵ on the initial configuration and further, on the long term kinetics, the bounding box is applied to the simulation results specified at the start of this section. Figure 5.5 shows the evolution of the mean grain size from $t_s = 200$ to $t = 3000$, computed by the conventional algorithm and by the bounding box algorithm for several values of ϵ. 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 initial configuration: when t_s is chosen earlier, some amount of the coalescence of the small grains which nucleated during the initial simulation 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 5.6 illustrates the evolution of the mean grain size as computed by the conventional algorithm and by the bounding box algorithm for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$, starting from the different time points t_s until $t = 3000$. As anticipated, the mean grain size is smaller when t_s is chosen earlier. For smaller values of ϵ, this effect is not clear: this can be explained by the effect discussed in the previous paragraph.

Both the influence of ϵ and t_s on the initial configuration are a consequence of creating an initial microstructure with the conventional algorithm. Nevertheless, we feel that this conventional method, possibly executed on a coarser mesh, is a convenient way to obtain an initial large polycrystalline structure since in the



(a) $\epsilon = 10^{-4}$

(b) $\epsilon = 10^{-5}$

Figure 5.6: Effect of t_s on the initial configuration: evolution of the 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 threshold values (a) $\epsilon = 10^{-4}$ and (b) $\epsilon = 10^{-5}$.

case of ideal grain growth, the typical grain structure with stationary grain size distribution is recovered after a short transition time (see also [140]).

Based on the observations concerning the memory reduction, accuracy and initial configuration for the different threshold values, we advocate to use $\epsilon = 10^{-5}$ or 10^{-6}. The simulations further showed that at every grid point, approximately 7 phase field variables are active. Since the bounding box algorithm only takes into account the active phase field values, this means that the computational requirements of the algorithm depend on the system size and not on the total number of phase field variables.

5.4 Computational requirements

5.4.1 Bounding box data structure

From the start of the bounding box algorithm on, all phase field values not exceeding ϵ in absolute value are assumed to be zero. They are not included in the sets B_i 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 space than the data from a conventional simulation. Table 5.3 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}, together with the conventional storage requirements. For smaller mean

Table 5.3: Bounding box storage requirements (GB) as a function of ϵ and the initial mean grain size (grid points), compared to conventional storage requirements.

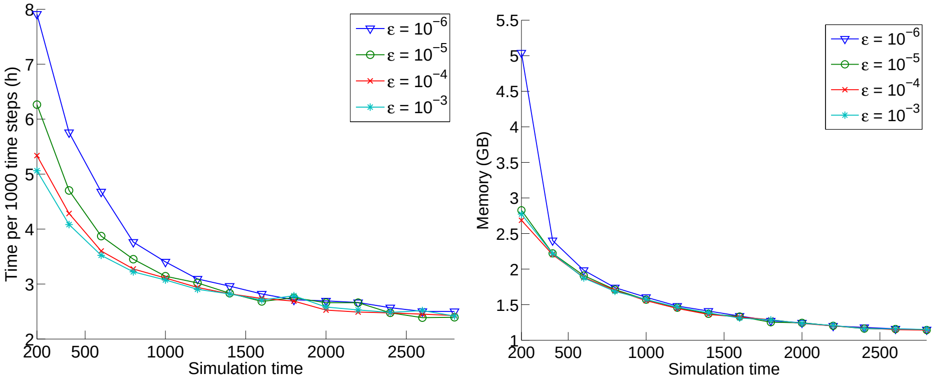
		threshold value ϵ				
starting t_s	mean grain size	10^{-3}	10^{-4}	10^{-5}	10^{-6}	conventional
$t_s = 200$	3.5×10^3	0.84	1.27	1.93	4.18	13.4
$t_s = 400$	7.8×10^3	0.70	1.00	1.37	1.90	13.4
$t_s = 600$	1.4×10^4	0.63	0.86	1.14	1.50	13.4
$t_s = 800$	2.0×10^4	0.59	0.77	0.98	1.24	13.4

grain size, the storage requirements are higher. Furthermore, the threshold value ϵ has a larger influence on the storage amount when the mean grain size is smaller.

5.4.2 Bounding box simulation

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 ϵ. Figure 5.7 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 threshold values ϵ. 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 ϵ, the computational requirements are higher. Later on in the simulations, the requirements are approximately the same for the different threshold values.

The conventional computational requirements are constant in time, whereas the bounding box algorithm requires less resources as simulation time progresses. In Table 5.4, 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, in contrast to the bounding box algorithm which could be executed on a single processor.



(a) Computing time per 1000 time steps (b) Computing memory per 1000 time steps

Figure 5.7: Evolution of (a) computing time (h) and (b) computing memory (GB) per 1000 time steps of the bounding box algorithm for $t_s = 200$ and different threshold values ϵ.

Table 5.4: Computing time (h) and computing memory (GB) as a function of the simulation time and the mean grain size (grid points), required during the first 5000 time steps of the bounding box algorithm with $\epsilon = 10^{-5}$ on a single processor, compared to the conventional algorithm on 20 processors.

		bounding box		conventional	
simulation time	mean grain size	time	memory	time	memory
200 → 400	$3.52 \times 10^3 \to 7.82 \times 10^3$	6.3	2.8	6.8	30.3
400 → 600	$7.82 \times 10^3 \to 1.35 \times 10^4$	4.7	2.2	6.8	30.3
600 → 800	$1.35 \times 10^4 \to 1.95 \times 10^4$	3.8	1.9	6.8	30.3
800 → 1000	$1.95 \times 10^4 \to 2.66 \times 10^4$	3.5	1.7	6.8	30.3
1000 → 1200	$2.66 \times 10^4 \to 3.46 \times 10^4$	3.1	1.6	6.8	30.3

5.5 Parallelisation

5.5.1 Parallelisation scheme

Concepts

We have further accelerated the bounding box algorithm by parallelisation in [131]. The grain regions G_i, with $i = 1, \ldots, p$, are divided into K different subsets of w_k elements. Each of these subsets is assigned to a different processor P_k and

$$\sum_{k=1}^{K} w_k = p. \tag{5.10}$$

The numbers w_k are chosen such that each processor disposes of approximately the same number of grain regions. Another possibility to divide the grain regions, which we have not implemented, is to add a weight to each grain region G_i that is related to the size of its bounding box $\mathcal{B}_i$. The numbers w_k can then be decided such that the sum of the weights is approximately equal for each processor.

This approach to parallelisation divides the computing work as well as the computing memory over the processors, thus accelerating the simulation process and alleviating the memory requirements for the individual processors.

Initialisation

When the system is initialised with Algorithm 5.2, each processor P_k is responsible for the creation of w_k grain regions. On the other hand, when the system is initialised from data resulting from previous bounding box simulation or as generated with Algorithm 5.1, a dedicated procedure divides the grain regions into K approximately equally sized subsets as described above. If particles are present in the microstructure, the data representing the second-phase particles is initialised on every processor. Currently, the parallel implementation of the bounding box algorithm does not include further dynamic load balancing: the partitioning of the work load only takes place at the beginning of a simulation.

Simulation

During a bounding box simulation, within each time step, processor P_k computes the solution of the model equations for the assigned grain regions. When a simulation is performed with continuum field model (2.39), at each time step, summation $\sum_{j=1}^{p} \eta_j^2$ has to be computed. Since the different phase field variables are spread over different processors, the computation of this sum requires communication between the processors. First, every processor P_k computes the partial sum Σ_k as

$$\Sigma_k = \sum_{\substack{j=1 \\ j \in S_k}}^{p} \eta_j^2 \ . \tag{5.11}$$

The set S_k is the set of all indices j for which grain region G_j is assigned to processor P_k. Next, the values of Σ_k are summed over all the processors, by performing the MPI_Allreduce-operation.

5.5.2 Evaluation

We evaluate the parallelisation scheme described above by running simulation experiments with continuum field model (2.39) on a $256 \times 256 \times 256$ grid with parameter values $f_{\mathrm{V}} = 0\%$, $L = 1$, $\kappa = 0.5$, $m = 1$, and discretisation spacings $\Delta t = 0.2$ and $\Delta x = 1$, until time point $t = 200$ is reached. The experiments are performed on an increasing number of processors, each time starting from the same initial microstructure with respectively $p = 25000$, $p = 50000$, and $p = 75000$ grain regions. The type of scaling thus studied is called strong scaling, and measures the speed-up of solving the same problem using more processors [68].

Figure 5.8 shows the computing time taken by the simulations as a function of the number of processors K. The total computing time quickly decreases for increasing values of K. However, for larger values of K, the speed-up of the parallelisation scheme seems to slow down, as the addition of more processors does not decrease the computing time significantly.

The speed-up of a parallelisation scheme is computed as the computing time required by one processor divided by the computing time required by K processors. Due to the communication between the processors and other overhead related to parallel computing, the speed-up is generally smaller than its ideal value K. To measure how close the speed-up is to its ideal value, the efficiency of a parallelisation scheme is defined as the speed-up divided by K. Figure 5.9 depicts the speed-up and the efficiency as a function of the number of involved processors K, for the different tested initial values of p. For a larger initial value of p, the work load per processor is higher and the parallelisation scheme proves to be more efficient. For each value of p, the number w_k of grain regions per processor decreases as the number of involved processes increases. Adding more processes thus makes the parallel scheme less and less efficient, since the extra communication overhead outweighs the possibility of dividing the work load. For $p = 75000$, the efficiency graph in Fig. 5.9(b) jumps above the value 1 for $K = 2$. This is related to the fact that the problem size is almost too large to run the simulation on a single processor.

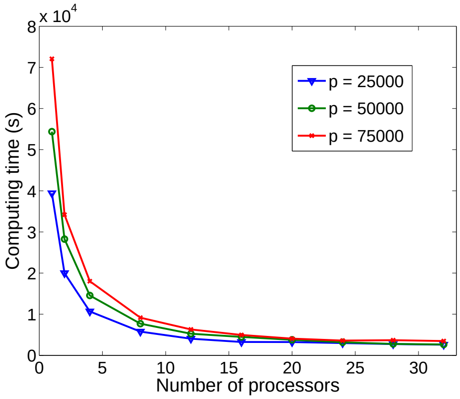
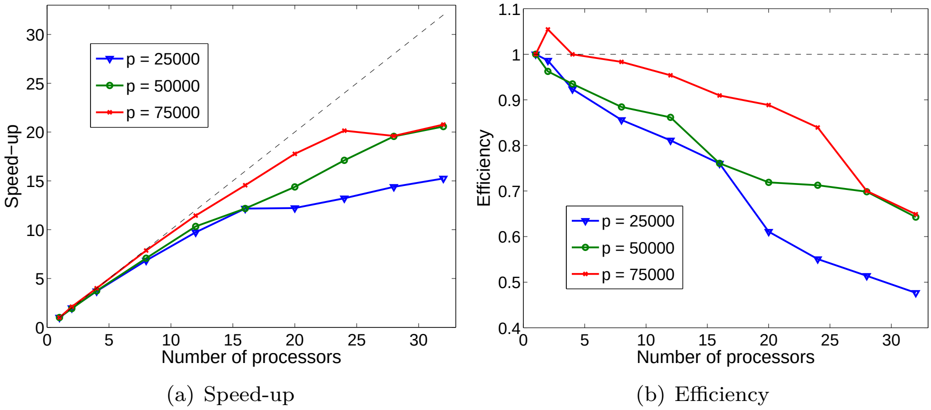


Figure 5.8: Computing time (s) as a function of the number of processors for simulations with continuum field model (2.39) on a $256 \times 256 \times 256$ grid, with $f_{\mathrm{V}} = 0\%$, $L = 1$, $\kappa = 0.5$, $m = 1$, $\Delta t = 0.2$ and $\Delta x = 1$, run until $t = 200$. Experiments are performed for $p = 25000$, $p = 50000$, and $p = 75000$ grain regions respectively.



(a) Speed-up

(b) Efficiency

Figure 5.9: (a) Speed-up and (b) efficiency as a function of the number of processors for simulations with continuum field model (2.39) on a $256 \times 256 \times 256$ grid, with $f_{\mathrm{V}} = 0\%$, $L = 1$, $\kappa = 0.5$, $m = 1$, $\Delta t = 0.2$ and $\Delta x = 1$, run until $t = 200$. Experiments are performed for $p = 25000$, $p = 50000$, and $p = 75000$ grain regions respectively.

5.6 Extending the bounding box algorithm

The object-oriented design of the bounding box data structure and the bounding algorithm allows for easy extension to more complex phase field models. Continuum field model (2.44) attributes different characteristics to each grain boundary through the parameters $\kappa(\Theta)$ and $\gamma(\Theta)$, with Θ the ensemble of parameters that defines the misorientation between neighbouring grains. This section discusses the extension of both the bounding box data structure and the bounding box algorithm to take these anisotropic characteristics into account.

5.6.1 Extending the bounding box data structure

Misorientation dependence of the grain boundary energy is brought into the existing data structure through two changes. First, a network structure is introduced in the structure in Fig. 5.1 by including links between neighbouring grains. Two grains are defined to be neighbours when their bounding boxes overlap, i.e. the intersection of their bounding boxes is not empty. This connection between neighbouring grain regions is indicated on the new, extended UML diagram shown in Fig. 5.10. Note that a grain region cannot become its own neighbour.

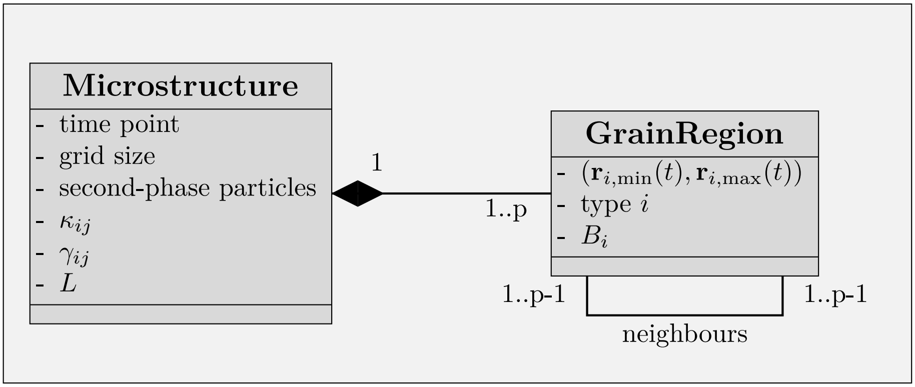


Figure 5.10: UML diagram of the bounding box data structure extended to model (2.44) with anisotropic properties.

The second difference lies in the assignment of the attributes of the Microstructure-class and the GrainRegion-class. The parameters κ_{ij} and γ_{ij} are both depending on the misorientation between the orientations of two neighbouring grain regions. Fig. 5.10 shows that every grain region can become a neighbour of every other grain region. Therefore, both parameters are now considered as attributes of the microstructure, as well as the mobility parameter L. Note that depending on the dimensions of the system, the crystallographic orientation can be described by a different type of parameter, e.g. a rotation matrix in the three-dimensional case. The attribute "type i" in Fig. 5.10 is thus a generic parameter representing the crystallographic orientation.

During the initialisation of the bounding box data structure, the network structure now has to be set up as well. For each newly generated grain, the initialisation algorithm therefore checks for every other grain whether their bounding boxes overlap and correspondingly creates new links in the network data structure.

5.6.2 Extending the bounding box algorithm

Two major changes are required to extend the bounding box algorithm. The first difference is due to the misorientation dependency of the parameters κ_{ij} and γ_{ij} in model (2.44). To calculate the misorientation dependent terms in the model equations, several steps are added to the algorithm just before the equations are solved for a particular grain region G_i (see Algorithm 5.4). For every neighbouring grain region G_j, first the intersection of the bounding boxes of G_i and G_j is determined. Next, the corresponding parameter values κ_{ij} and γ_{ij} are retrieved, and finally, the contributions of the i/j connection are calculated and added to the respective coupling terms.

Algorithm 5.4: Extended bounding box algorithm

```
Input: Bounding box data structure at $t_1$; $t_{\text{end}}$, $\epsilon$ and $M$
Output: Bounding box data structure at $t_{\text{end}}$
for $t = t_1, t_2, \ldots, t_{end}$ do
    for $i = 1$ to $p$ do
        Add a margin of $M$ grid points in each grid direction to $B_i$;
        Fill the margin with zeros;
        foreach neighbour $G_j$ of grain region $G_i$ do
            Determine the intersection of the bounding boxes;
            Retrieve the parameter values $\kappa_{ij}$ and $\gamma_{ij}$;
            Compute and add the contribution of the $i/j$ connection to the
            misorientation dependent terms;
        end
        Solve the discretised model equations for $B_i$;
        Check whether $G_i$ has grown or shrunk and update the bounding
        box delimiters and $B_i$ accordingly;
    end
    Update the grain network;
end
```

The second change is that at the end of each time step, the grain network is brought up-to-date. When a grain region has grown and meets new neighbours, the algorithm adds the corresponding new connections to the network structure. Since only the second-order neighbours of a grain region can become its new neighbours, the algorithm restricts to these grain regions to check for new neighbours. Similarly, when a grain region has shrunk and looses contact with previously neighbouring grain regions, the algorithm removes the corresponding connections from the network. When a grain region has disappeared, the connections between this grain region and its former neighbouring grain regions, as well as the grain region itself, are removed from the network.

Depending on the generation method of the grain regions, only shortly after initialisation, it is possible that the microstructure is not completely covered by the bounding boxes and that a grain region meets new neighbours that were previously not connected as second-order neighbours. Therefore, until the grain regions have impinged and the microstructure is completely covered, or more specifically, until the constraint

$$\forall \mathbf{r} \in \Omega : \max_{i=1,\ldots,p} |\eta_i(\mathbf{r})| > \epsilon \tag{5.12}$$

is met, with Ω the system domain, all grain regions need to be checked for new overlap.

5.6.3 Parallelisation

The extended version of the bounding box algorithm was also parallelised, but because of the complex coupling between the boxes, another approach was followed than in Section 5.5: the algorithm was accelerated through task parallelism. This means that every involved processor has access to the same data structure, but only processes a part of the bounding boxes.

Similar to the parallelisation scheme described in Section 5.5, within each time step, processor P_k solves continuum field model (2.44) for an assigned subset of w_k grain regions, with $\sum_{k=1}^{K} w_k = p$, where K is the total number of processors and p the total number of grain regions. Again, the sizes w_k of the subsets are chosen such that each processor P_k is responsible for approximately the same amount of grain regions. During a simulation, at the end of each time step, each processor has computed an update for a subset of w_k grain regions. Before the next time step can be computed, all local updates are communicated among the processes, such that at the beginning of the next time step, all processes again have local access to the same information.

5.7 Conclusion

In this chapter, a sparse bounding box algorithm is presented to perform efficient phase field simulations of microstructural evolution in polycrystalline materials. The algorithm only solves the phase field equations locally, inside bounding boxes that delimit regions of active phase field variables. In contrast to other sparse algorithms, the bounding box data structure naturally allows for semi-implicit time integration.

The computational requirements of the bounding box algorithm depend on the system size and not on the total number of involved phase field variables. In combination with the one-to-one mapping between grain regions and phase field variables, this allows to perform coalescence-free simulations of grain growth without the excessive memory usage or computing time associated with existing methods.

The bounding box algorithm is further accelerated through parallelisation, which is described and evaluated in Section 5.5. Because of its object-oriented design, the bounding box algorithm has advantages in post-processing and is extendible to more complex models. The latter is illustrated in Section 5.6, where extensions of the bounding box implementation are discussed with regard to grain growth in a microstructure with anisotropic boundary properties.

The applicability of the bounding box algorithm will be illustrated in Chapter 7 by phase field simulations of grain growth in the presence of second-phase particles with continuum field model (2.39), and in Chapter 8 by phase field simulations of grain growth in a microstructure with anisotropic boundary properties with model (2.44).

Chapter 6

Multigrid method

6.1 Introduction

For realistic structures, the required execution time and memory of phase field models such as (2.27) and (2.37) impose limitations on any practical implementation. As mentioned in Chapter 3, several algorithms and methods have been designed and tested to address these computational challenges. In particular, the use of an efficient solver is very important. In this chapter, we present a nonlinear multigrid solver based on the Full Approximation Scheme (FAS), developed for model (2.27).

In [120, 42], multi-phase field model (2.27) is presented and simulation results are discussed in [97]. The solvers employed in these works are constructed using explicit discretisation schemes. While explicit time steppers can be computationally very efficient, they generally suffer a limited time step size due to a strict stability condition. In contrast, implicit discretisation methods allow the choice of a larger time step size, but result in systems that need to be solved at each time step. To make sure that this extra execution time is compensated by the possibility of a large time step size, it is necessary to look for dedicated solvers. For phase field model (2.27) in particular, the nonlinear systems resulting from implicit discretisation are very large.

Multigrid algorithms accelerate the convergence of standard iterative solvers. When the components of these algorithms are carefully selected, they can scale linearly with the number of spatial unknowns. A number of studies has been performed on the development of multigrid methods for phase field simulations. In [111], a nonlinear multigrid solver is implemented to solve a phase field model for binary alloy solidification. The solver is based on FAS and follows an adaptive grid approach. In the work of [73], monotone multigrid solvers are constructed to solve vector-valued Allen-Cahn equations, which are extensively used for phase field modelling. In [62], an extension of the method of [73] is presented to solve the Cahn-Hilliard equation, employed in conservative phase field models. Furthermore, Uzawa-type multigrid algorithms [10], linear multigrid methods [21] and FAS methods [64, 139] have been developed to solve the latter equation type. However, in all of these studies, only a few phase field variables are considered.

In this chapter, a nonlinear multigrid solver based on FAS is proposed for two-dimensional phase field simulations with the multi-phase field model according to [120, 42], which is described in detail in Section 2.4. This is the first time that a FAS solver is implemented for this type of phase field models. In Section 6.2, we restrict the model to a simple phase field model consisting of one phase. Three classical discretisation schemes are applied to this simple model. The resulting systems are solved with two different types of multigrid solvers, allowing us to introduce and explain the basic concepts of multigrid methods. Section 6.3 is dedicated to the full multi-phase field model. After discretisation of the model, the nonlinear multigrid solver is presented. The solver shows the desired grid size independent properties. Simulation results are presented in Section 6.4. The chapter ends with a conclusion in Section 6.5.

6.2 A simple phase field model

In this section, we introduce the concepts of multigrid methods. We start by reducing the multi-phase field model to a single phase system, which is easier to handle. Next, three different classical discretisation schemes are applied to this model. Each of these discretisation schemes has different stability properties and requires a different type of solution method. The first scheme does not require for a solver, since it is explicit, whereas for each of the two other implicit schemes, a multigrid solver is implemented. One of these solvers is a nonlinear FAS solver, which will be applied to the full multi-phase field model in Section 6.3.

6.2.1 Model derivation

We limit multi-phase field model (2.27) to a system containing two phases ϕ_1 and ϕ_2. The two phases can further be reduced to a single phase by renaming ϕ_1 to ϕ, and using equality $\phi_2 = 1 - \phi_1 = 1 - \phi$, which originates from the sum constraint in (2.18). The gradient free energy density $a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi})$ defined in (2.20) now reads

$$a(\boldsymbol{\phi}, \nabla\boldsymbol{\phi}) = a(\nabla\phi) = s\sigma\|\nabla\phi\|^2, \tag{6.1}$$

where s is a normalisation constant and σ the surface energy density. For the bulk potential, we choose the single phase version of the multi-well potential (2.22), also called the double-well potential:

$$w(\phi) = 9\sigma\phi^2(1-\phi)^2. \tag{6.2}$$

The governing equation for the phase field variable ϕ can now be derived by taking the variational derivative of the free energy (2.19) with respect to ϕ:

$$\omega\epsilon\frac{\partial\phi}{\partial t} = -\frac{\delta\mathcal{F}(\phi)}{\delta\phi} = \nabla\cdot\frac{\partial\mathcal{F}(\phi)}{\partial(\nabla\phi)} - \frac{\partial\mathcal{F}(\phi)}{\partial\phi}. \tag{6.3}$$

Applying the derivative in equation (6.3) to the gradient free energy density (6.1) and the double-well potential (6.2) yields a simple phase field model:

$$\omega\epsilon\frac{\partial\phi}{\partial t} = \epsilon(2s\sigma)\nabla^2\phi - \frac{1}{\epsilon}18\sigma\phi(2\phi^2-3\phi+1). \tag{6.4}$$

Periodic boundary conditions are assumed.

6.2.2 Discretisation

We apply three different classical finite difference schemes to model (6.4) and briefly comment on their stability properties.

Explicit discretisation

First-order explicit discretisation of model (6.4) on a $[0,1]\times[0,1]$ domain yields system

$$\begin{aligned}\phi_{r,s}^{n+1} &= \phi_{r,s}^{n} + \frac{\Delta t}{\omega}2s\sigma\left(\frac{\phi_{r+1,s}^{n}-2\phi_{r,s}^{n}+\phi_{r-1,s}^{n}}{(\Delta x)^2}+\frac{\phi_{r,s+1}^{n}-2\phi_{r,s}^{n}+\phi_{r,s-1}^{n}}{(\Delta y)^2}\right)\\ &\quad - \frac{\Delta t}{\omega\epsilon^2}18\sigma\phi_{r,s}^{n}(1-\phi_{r,s}^{n})(1-2\phi_{r,s}^{n}),\\ &\qquad r=1,\ldots,N_x, \qquad s=1,\ldots,N_y,\end{aligned} \tag{6.5}$$

with $N_x = 1/\Delta x$ and $N_y = 1/\Delta y$. Second-order central differences are used to discretise the Laplace operator. In this scheme, the values $\phi_{r,s}^{n+1}$ at each time step are computed as an explicit function of the values $\phi_{r,s}^{n}$ at the previous time step. Since no matrix system has to be solved, it is a computationally attractive choice. However, because explicit discretisation schemes suffer from a strong stability restriction on the time step size, simulations with this type of discretisation schemes can easily take too much time.

Semi-implicit discretisation

The stability restriction of the explicit scheme on the choice of the time step size Δt is mostly due to the diffusion part of the model equation, which corresponds to the stability discussion on continuum field model (2.37) in Chapter 4. This problem can be alleviated by using an implicit, rather than an explicit discretisation scheme. We now apply a semi-implicit scheme that treats the diffusion part of model (6.4) implicitly and the reaction part explicitly. This way, a larger time step size can be used than for an explicit method, while the system to solve is linear and thus not very complicated:

$$\begin{aligned}\phi_{r,s}^{n+1} - \frac{\Delta t}{\omega} 2s\sigma \left(\frac{\phi_{r+1,s}^{n+1} - 2\phi_{r,s}^{n+1} + \phi_{r-1,s}^{n+1}}{(\Delta x)^2} + \frac{\phi_{r,s+1}^{n+1} - 2\phi_{r,s}^{n+1} + \phi_{r,s-1}^{n+1}}{(\Delta y)^2} \right) \\ = \phi_{r,s}^{n} - \frac{\Delta t}{\omega \epsilon^2} 18\sigma \phi_{r,s}^{n}(1-\phi_{r,s}^{n})(1-2\phi_{r,s}^{n}), \\ r = 1, \dots, N_x, \qquad s = 1, \dots, N_y. \quad (6.6)\end{aligned}$$

Implicit discretisation

Applying a first-order, fully implicit discretisation scheme to (6.4) yields system

$$\begin{aligned}\phi_{r,s}^{n+1} - \frac{\Delta t}{\omega} 2s\sigma \left(\frac{\phi_{r+1,s}^{n+1} - 2\phi_{r,s}^{n+1} + \phi_{r-1,s}^{n+1}}{(\Delta x)^2} + \frac{\phi_{r,s+1}^{n+1} - 2\phi_{r,s}^{n+1} + \phi_{r,s-1}^{n+1}}{(\Delta y)^2} \right) \\ + \frac{\Delta t}{\omega \epsilon^2} 18\sigma \phi_{r,s}^{n+1}(1-\phi_{r,s}^{n+1})(1-2\phi_{r,s}^{n+1}) = \phi_{r,s}^{n}, \\ r = 1, \dots, N_x, \qquad s = 1, \dots, N_y. \quad (6.7)\end{aligned}$$

Fully implicit schemes are unconditionally stable and allow a large time step size Δt. However, the discretised model that has to be solved at each time step is now nonlinear.

Stability properties

Table 6.1 lists the stability conditions for the explicit, the semi-implicit and the implicit discretisation schemes. The restriction is worked out for the set of parameter values $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\omega = 1.0$, $\sigma = 1.0$, and $\Delta x = \Delta y = 1/64$.

Table 6.1: Stability conditions for the explicit, the semi-implicit and the implicit discretisation scheme. The conditions are calculated for an example set of parameter values $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\omega = 1.0$, $\sigma = 1.0$, and $\Delta x = \Delta y = 1/64$.

Scheme	Stability condition	Example
Explicit:	$\Delta t < \left(\frac{8s\sigma}{\omega \Delta x^2} + \frac{9\sigma}{\omega \epsilon^2}\right)^{-1}$	$\Delta t < 0.00123$
Semi-implicit:	$\Delta t < \left(\frac{9\sigma}{\omega \epsilon^2}\right)^{-1}$	$\Delta t < 0.1111$
Implicit:	–	–

The stability condition for the explicit scheme is the strongest: there is a dependency on the square of Δx in the inequality. In contrast, the implicit scheme is unconditionally stable.

6.2.3 Multigrid solver

Linear multigrid solver

To solve explicit discretisation scheme (6.5), at each time step, the values $\phi_{r,s}^{n+1}$ are computed as an explicit function of the values $\phi_{r,s}^{n}$ at the previous time step. In contrast, discretisation schemes (6.6) and (6.7) both define a system of equations that have to be solved simultaneously. To make sure that the execution time required to solve these systems is compensated by the possibility of a large time step size, it is appropriate to look for efficient solvers.

Discretisation scheme (6.6) describes a linear system of equations $A\phi = b$. This system can be solved by a standard iterative solver, such as the Jacobi or the Gauss-Seidel method. Unfortunately, the convergence rate of standard iterative solvers typically stalls after a small number of iterations. A closer look into the behaviour of the error during these computations reveals that the high-frequency modes of the error are smoothed out after a few iterations already, while the low-frequency modes remain much longer present, causing a decrease of the convergence rate after a few iteration steps. However, the resulting smooth error can be well approximated on a coarse grid. Multigrid methods are based on the idea that the low-frequency error that is present on a fine grid can be removed efficiently by certain computations on a coarser grid. Because calculations on a coarse grid are significantly less expensive than calculations on a fine grid, the overall efficiency of an iterative solver can be improved.

Multigrid algorithms consist of different components:

- a *smoothing operator*: a solver with appropriate smoothing properties to reduce the high-frequency modes of the error on a fine grid (e.g. Jacobi, Gauss-Seidel),
- intergrid operators for the transfer between the coarse and the fine grids:
  - a *restriction operator*: for the downsampling from the fine to the coarse grid (e.g. injection, full weighting),
  - a *prolongation operator*: to interpolate from the coarse onto the fine grid (e.g. bilinear interpolation),
- a *coarse-grid solver*: the solver on the coarsest grid. Since the coarsest grid is often small, the coarse-grid solver can be a direct solver.

An important characteristic of multigrid methods is that the convergence properties of a multigrid solver with carefully selected components are independent of the grid size. Moreover, the computational work of a multigrid solver often scales with the number of unknowns. These features make multigrid solvers very appropriate for large-scale simulations.

Figure 6.1 shows a schematic representation of a two-grid method following the multigrid principle. One cycle of the two-grid method consists of three parts. First, ν_1 iterations of the smoothing operator are applied to system $A\phi = b$ on the fine grid. This way, the high frequencies of the error $e = \phi^* - \phi$ are removed, with ϕ^* the exact solution of $A\phi = b$. After this step, also called the *presmoothing* step, the error e is sufficiently smooth so that it can be well approximated on the coarse grid. Second, a *coarse-grid correction step* is performed. The residual $r = b - A\phi$ is computed and restricted to the coarse grid with the restriction operator. On this coarse grid, the error e is computed by solving the system $Ae = r$, also called the defect equation. The computed correction e is then prolongated back to the fine grid and used to construct the new approximation $\phi_{\text{new}} = \phi + e$. Because the error is smooth, interpolation should work very well and the correction of the fine-grid solution should be effective. Finally, ν_2 *postsmoothing* iterations are performed to remove any high-frequency error components that were reintroduced by the coarse-grid correction through the prolongation operator.

To solve system $Ae = r$ on the coarse grid, the two-grid method can be applied recursively. This way, a multigrid cycle is obtained that uses an entire grid hierarchy and proceeds as in Algorithm 6.1. The smallest system, on the coarsest grid of the hierarchy, can be solved by a direct solver. The parameter γ specifies the number of multigrid cycles that is to be carried out on the current grid. Figure 6.2 shows a schematic representation of a multigrid cycle for two different values of γ, in the case of a three-grid hierarchy and in the case of a four-grid hierarchy. For $\gamma = 1$, the schematic representation of a multigrid cycle has the shape of the letter V. For this reason, this type of cycle is also called a V-cycle. Similarly, a multigrid

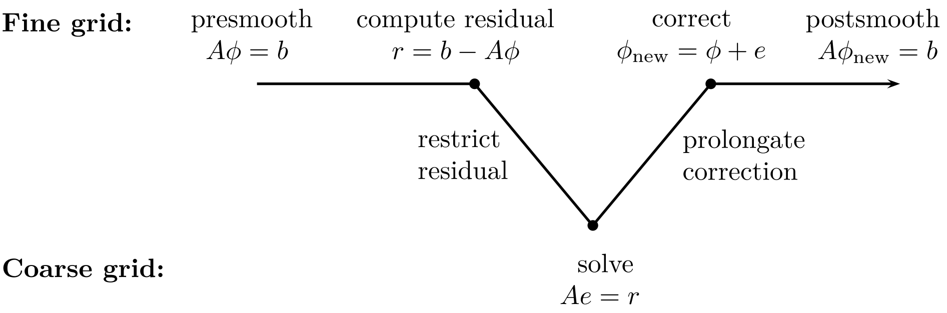


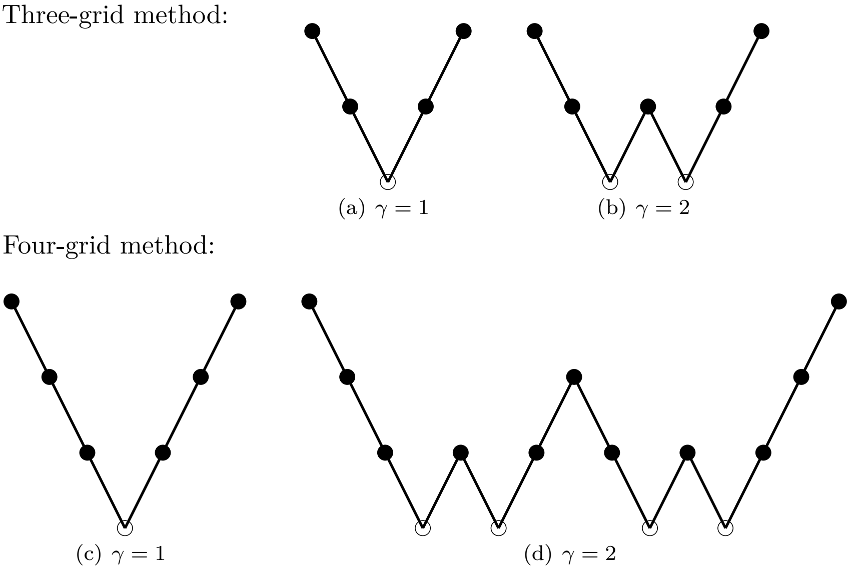
Figure 6.1: Schematic representation of a two-grid method.

cycle with $\gamma = 2$ is called a W-cycle. Unless explicitly mentioned, we will use the parameter value $\gamma = 1$ in this text. Also, we will apply standard coarsening to construct the grid hierarchy. Coarser grids are thus obtained by doubling the grid spacing in both dimensions.

Algorithm 6.1: Standard linear multigrid cycle

```
$\phi_{\text{f}} = \texttt{multigridCycle}(A_{\text{f}}, b_{\text{f}}, \phi_{\text{f}}, \nu_1, \nu_2, \gamma)$;
if $A_f$ sufficiently small then
    $\phi_{\text{f}} \longleftarrow A_{\text{f}}^{-1} b_{\text{f}}$;
else
    for cycle $= 1$ to $\gamma$ do
        $\phi_{\text{f}} \longleftarrow \texttt{preSmooth}(A_{\text{f}}, b_{\text{f}}, \phi_{\text{f}}, \nu_1)$;
        $r_{\text{f}} \longleftarrow b_{\text{f}} - A_{\text{f}}\phi_{\text{f}}$;
        $b_{\text{c}} \longleftarrow \texttt{restrict}(r_{\text{f}})$;
        $v_{\text{c}} \longleftarrow 0$;
        $v_{\text{c}} \longleftarrow \texttt{multigridCycle}(A_{\text{c}}, b_{\text{c}}, v_{\text{c}}, \nu_1, \nu_2, \gamma)$;
        $\phi_{\text{f}} \longleftarrow \phi_{\text{f}} + \texttt{prolongate}(v_{\text{c}})$;
        $\phi_{\text{f}} \longleftarrow \texttt{postSmooth}(A_{\text{f}}, b_{\text{f}}, \phi_{\text{f}}, \nu_2)$;
    end
end
```

To solve system (6.6), we implemented a linear multigrid solver. For both smoothing operators, the Gauss-Seidel method was chosen, and respectively full weighting and bilinear interpolation were chosen for the restriction and the prolongation operator. The grid hierarchy was implemented such that the number of grid points in the x- and in the y-dimension is equal and a power of two. The coarsest grid consists of one grid point. The system on the coarsest grid can thus be solved by simply inverting the matrix A, which is then defined by one scalar. At each time step, system $A\phi = b$ on the finest grid is solved by performing



(a) $\gamma = 1$

(b) $\gamma = 2$

(c) $\gamma = 1$

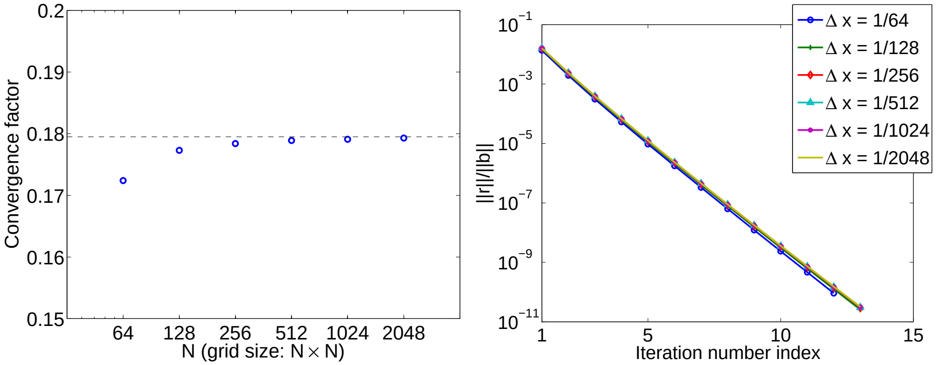
(d) $\gamma = 2$

Figure 6.2: Schematic representation of one multigrid cycle for two different numbers of grids, and for two different values of γ. The solid dots ($\bullet$) represent smoothing steps, while the open dots ($\circ$) represent solution on the coarsest grid. The connecting lines (\ and /) indicate the intergrid transfers.

multigrid cycles as described in Algorithm 6.1, until the convergence criterion or a maximal number of iterations is met. The convergence criterion uses the norm of the residual scaled by the norm of the right-hand side: convergence is reached when $\|r\|/\|b\| \leq \tau$ is obtained, with τ a tolerance value provided by the user.

To verify the properties of the implemented solver, we perform simulations on a square domain $[0, 1] \times [0, 1]$ for the set of parameter values $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, $\Delta t = 0.1$, and $\tau = 10^{-11}$, for several values of $\Delta x = \Delta y$. The smallest tested fine grid contained 64×64 grid points, and for each subsequent larger grid, the grid spacing Δx was divided by 2, each time obtaining more spatial accuracy. The phase field variable ϕ is initialised with the value 1 inside a square delimited by the points $(N/5, N/5)$ and $(4N/5, 4N/5)$, with N the number of grid points in each dimension. Outside the square, ϕ is set to zero.

Figure 6.3 depicts the convergence factor and the convergence behaviour of the linear multigrid solver during the tenth time step of a simulation, for different values of Δx. It can be seen that the convergence factor is asymptotically independent of the grid size. Another property of an optimal multigrid solver is that the amount of computing work scales linearly with the number of unknowns, i.e. with the grid size. Figure 6.4 shows a logarithmic plot of the execution time



(a) Convergence factor

(b) Evolution of $||r||/||b||$

Figure 6.3: Convergence factor and convergence behaviour of the linear multigrid solver during the tenth time step for different values of Δx on a $[0,1] \times [0,1]$ domain. The set of parameter values $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, $\Delta t = 0.1$, and $\tau = 10^{-11}$ is applied.

Grid size	Execution time
64 × 64	3.9000e-01 s
128 × 128	1.4300e+00 s
256 × 256	7.6900e+00 s
512 × 521	3.5420e+01 s
1024 × 1024	1.4910e+02 s
2048 × 2048	6.1038e+02 s

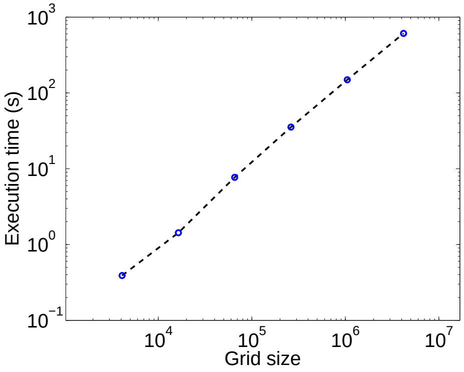


Figure 6.4: Execution time required by the linear multigrid solver to compute 100 time steps with the parameter set $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, $\Delta t = 0.1$, and $\tau = 10^{-11}$, as a function of the grid size.

needed to compute the first 100 time steps of a simulation as a function of the grid size. It can be seen that the required execution time scales linearly with the number of unknowns. The table in Fig. 6.4 lists the execution times for the different grid sizes.

Nonlinear multigrid solver

Implicit discretisation scheme (6.7) defines a nonlinear system $A(\phi) = b$, which cannot be solved by the linear multigrid solver. To solve this system with a multigrid method, there exist two types of approaches. In the first approach, a global linearisation method, e.g. Newton's method, is applied to the nonlinear problem. The resulting linearised system can be solved by a standard multigrid solver. In the second type of approach, a multigrid method is directly applied to the nonlinear problem. The two main multigrid components, namely the error smoothing and the coarse-grid correction are now used to solve the nonlinear problem itself. This leads to the so-called Full Approximation Scheme (FAS), which is the approach we will follow. An advantage of FAS compared to the first approach are the memory requirements: it is not necessary to compute and store the fine-grid Jacobian, as is necessary in Newton-based solution methods.

Algorithm 6.2: Full Approximation Scheme multigrid cycle

```
$\phi_\mathrm{f}$ = FASMultigridCycle($A_\mathrm{f}, b_\mathrm{f}, \phi_\mathrm{f}, \nu_1, \nu_2, \gamma$);
if $A_f$ sufficiently small then
    $\phi_\mathrm{f} \longleftarrow A_\mathrm{f}^{-1} b_\mathrm{f}$;
else
    for cycle = 1 to $\gamma$ do
        $\phi_\mathrm{f} \longleftarrow$ preSmooth($A_\mathrm{f}, b_\mathrm{f}, \phi_\mathrm{f}, \nu_1$);
        $r_\mathrm{f} \longleftarrow b_\mathrm{f} - A_\mathrm{f}(\phi_\mathrm{f})$;
        $r_\mathrm{c} \longleftarrow$ restrictOne($r_\mathrm{f}$);
        $\phi_\mathrm{c} \longleftarrow$ restrictTwo($\phi_\mathrm{f}$);
        $b_\mathrm{c} \longleftarrow A_\mathrm{c}(\phi_\mathrm{c}) + r_\mathrm{c}$;
        $u_\mathrm{c} \longleftarrow$ FASMultigridCycle($A_\mathrm{c}, b_\mathrm{c}, \phi_\mathrm{c}, \nu_1, \nu_2, \gamma$);
        $v_\mathrm{c} \longleftarrow u_\mathrm{c} - \phi_\mathrm{c}$;
        $\phi_\mathrm{f} \longleftarrow \phi_\mathrm{f} +$ prolongate($v_\mathrm{c}$);
        $\phi_\mathrm{f} \longleftarrow$ postSmooth($A_\mathrm{f}, b_\mathrm{f}, \phi_\mathrm{f}, \nu_2$);
    end
end
```

Similar to the linear case, the nonlinear FAS multigrid method can be recursively defined on the basis of a two-grid method. The fundamental idea of nonlinear multigrid is the same as in the linear case. However, on the coarse grid, we do not work with the errors, but with full approximations to the discrete solution. Algorithm 6.2 shows the algorithmic description of one FAS cycle. No global linearisation is needed in the FAS multigrid process, except on the coarsest grid. On the other grids, first, the error of the fine-grid approximation ϕ_f is smoothed by ν_1 iterations of the nonlinear presmoother. Second, in contrast to the linear case, not only the residual r_f is transferred to the coarse grid, but also the relaxed approximation ϕ_f itself. Two restriction operators are applied. The first restriction operator acts on the residual, while the second transfers the full approximation to the coarse grid. Third, the coarse-grid error is computed. The nonlinear equivalent of the defect equation, which is given by

$$A_{\mathrm{c}}(\phi_{\mathrm{c}} + v_{\mathrm{c}}) - A_{\mathrm{c}}\phi_{\mathrm{c}} = r_{\mathrm{c}}, \tag{6.8}$$

or

$$A_{\mathrm{c}}(u_{\mathrm{c}}) = A_{\mathrm{c}}(\phi_{\mathrm{c}} + v_{\mathrm{c}}) = r_{\mathrm{c}} + A_{\mathrm{c}}\phi_{\mathrm{c}}, \tag{6.9}$$

is now solved for the unknown u_{c}. Since only the error is smoothed by the relaxation process, it is only the error that can be well approximated on the coarser grids. Therefore, only the correction v_{c} is transferred back to the fine grid, as in the linear case, and not the full approximation u_{c}. Finally, ν_2 iteration steps of the nonlinear postsmoother are applied.

To solve equations (6.7), we have implemented a nonlinear FAS multigrid method. In this implementation, the restriction operator of the residual is full weighting, while the restriction operator for the transfer of the current approximation ϕ_{f} to the coarse grid was selected to be injection, which is a common choice. For the prolongator, bilinear interpolation is chosen. The same grid hierarchy as for the linear multigrid solver is used. Both the presmoothing and the postsmoothing procedure are now nonlinear relaxation procedures, with suitable error smoothing properties. In this work, the Gauss-Seidel method using local linearisation is employed. We can rewrite nonlinear system (6.7) as

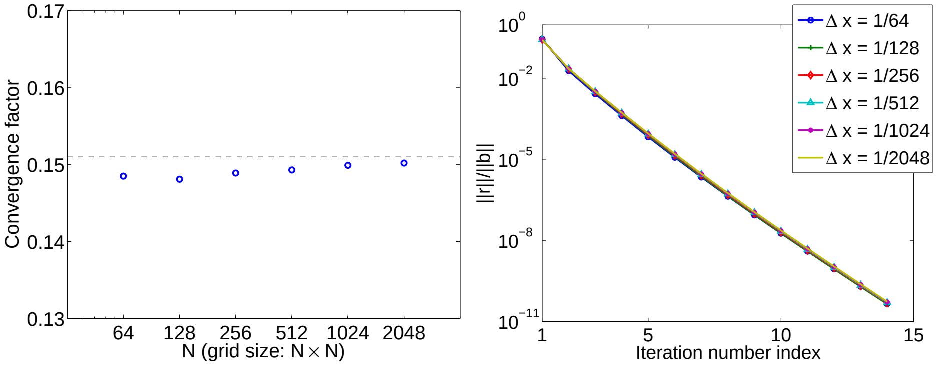
$$A_{r,s}(\phi_{1,1},\ \ldots,\ \phi_{r,s},\ \ldots,\ \phi_{N_x,N_y}) = b_{r,s}, \qquad r = 1,\ldots,N_x, \qquad s = 1,\ldots,N_y. \tag{6.10}$$

The nonlinear Gauss-Seidel method iterates over system (6.10) and solves every $((s-1)N_x + r)$th equation in turn for the unknown $\phi_{r,s}$. The $(m+1)$th iteration then reads

$$A_{r,s}(\phi_{1,1}^{m+1},\ \ldots,\ \phi_{r-1,s}^{m+1},\ \phi_{r,s}^{m+1},\ \phi_{r+1,s}^{m},\ \ldots,\ \phi_{N_x,N_y}^{m}) = b_{r,s}, \qquad r = 1,\ldots,N_x, \qquad s = 1,\ldots,N_y, \tag{6.11}$$

and is solved by Newton's method.

The properties of the implemented FAS solver are now verified for the same set of parameter values as the linear multigrid solver, but with a larger time step size, namely $\Delta t = 0.1$. The same hierarchy of grids is tested, with the same initial states. Figure 6.5 depicts the convergence factor and the convergence behaviour of the FAS solver during the tenth time step, for different spatial accuracies. Again,



(a) Convergence factor

(b) Evolution of $||r||/||b||$

Figure 6.5: Convergence factor and convergence behaviour of the FAS solver during the tenth time step for different values of $\Delta x = \Delta y$ on a $[0,1] \times [0,1]$ domain. The set of parameter values $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, $\Delta t = 0.1$, and $\tau = 10^{-11}$ is applied.

Grid size	Execution time
64 × 64	1.2000e-01 s
128 × 128	4.3000e-01 s
256 × 256	1.8900e+00 s
512 × 521	8.0200e+00 s
1024 × 1024	3.2500e+01 s
2048 × 2048	1.2969e+02 s

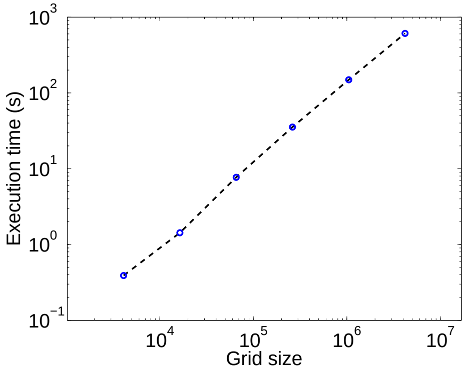


Figure 6.6: Execution time required by the FAS solver to compute 100 time steps with the parameter set $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, $\Delta t = 0.1$, and $\tau = 10^{-11}$, as a function of the grid size.

the convergence factor proves to be independent of the grid size. Figure 6.4 shows a logarithmic plot of the execution time needed to compute 100 time steps. The execution times for the different grid sizes are listed. As for the linear multigrid solver, it can be seen that the execution time scales linearly with the grid size.

6.2.4 Comparison

Next, we compare the implementation of both multigrid solvers with the explicit solver for the parameter set $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, and $\tau = 10^{-11}$, on a 512×512 grid with $\Delta x = \Delta y = 1/512$. For the explicit solver, a time step size of $\Delta t = 0.00001$ is applied, while for the linear and the nonlinear solver, respectively $\Delta t = 0.1$ and $\Delta t = 1.0$ are applied. This is in agreement with the stability conditions of Table 6.1. Table 6.2 shows the simulation time required by the three solvers on a 2.4 GHz Intel Core 2 Duo Processor (with 1 GB RAM) to run until $t = 10$. Even though one time step with the explicit solver is much faster than one time step with the multigrid solvers, the possibility of a larger time step size due to the different stability characteristics largely compensates this. The nonlinear FAS solver proves to be the fastest.

Table 6.2: Comparison of the different solvers for the explicit, the semi-implicit, and the implicit discretisation scheme. All three solvers were run for the parameter set $s = 1/(6.4)^2$, $\epsilon = 1.0$, $\sigma = 1.0$, and $\tau = 10^{-11}$, on a 512×512 grid with $\Delta x = \Delta y = 1/512$, each for a different time step size Δt.

Solver	Δt	# time steps	Execution time	Time/time step
Explicit	0.00001	1000000	28512.00 s	0.03 s
Semi-implicit	0.1	100	35.31 s	0.35 s
Implicit	1.0	10	8.06 s	0.81 s

All three discretisation schemes are of first-order accuracy. The explicit scheme is straightforward to implement, but because of the strong stability condition and thus severe restriction on the size of the time step, simulations with this discretisation scheme take too much time. Simulations with the nonlinear solver of the implicit scheme are computationally intensive. Although this scheme is unconditionally stable, its accuracy is still only first-order. Therefore, depending on the desired accuracy, the linear or the nonlinear multigrid solver is a more appropriate choice for model (6.4).

6.3 Multi-phase field model

In this section, we describe a nonlinear multigrid solver based on the FAS scheme for two-dimensional phase field simulations with multi-phase field model (2.27) [132]. The model is discretised with a semi-implicit discretisation scheme in Section 6.3.1. In Section 6.3.2, the nonlinear multigrid solver is constructed. The multigrid components are described and the properties of the solver are tested and discussed. The solver has grid size independent properties and scales linearly with the number of spatial unknowns. No comparison is made with other classical iterative solvers.

In Section 6.4, simulation results for two types of applications illustrate the operation of the solver.

6.3.1 Discretisation

To obtain statistically relevant results from a grain growth simulation, the use of a large domain with a large number of grains is required. Since this requirement makes model (2.27) memory intensive, it is appropriate to employ a two-step scheme for the discretisation in time, rather than a multi-step scheme, where the values of all phase field variables have to be retained for several time steps or stages simultaneously. Central differences are used for the discretisation of the first and second-order spatial derivatives. No artificial oscillations were observed that would have required the use of one-sided differences for the first-order derivative term. To avoid spurious oscillations near the sharp grain boundaries, we have decided not to use the Crank-Nicolson method, which is not stiffly stable.

Model equations (2.27) are discretised with a first-order semi-implicit discretisation scheme that treats the gradient energy part implicitly and the potential part explicitly:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{1}{\omega}\left(\nabla \cdot \frac{\partial a}{\partial(\nabla\phi_i)} - \frac{\partial a}{\partial \phi_i}\right)^{n+1} - \frac{1}{\omega\epsilon^2}\left(\frac{\partial w}{\partial \phi_i}\right)^{n} - \lambda^{n+1}, \quad i = 1, \dots, p, \tag{6.12a}$$

$$\sum_{i=1}^{p} \phi_i^{n+1} = 1. \tag{6.12b}$$

This way, all terms involving spatial derivatives are treated implicitly and the stability condition on the time step size proved to be independent of the spatial grid size. We also experimented with fully implicit schemes. However, this led to a convergence deterioration of the Newton solver. After some rearrangements, the following numerical scheme is obtained:

$$\phi_i^{n+1} - \frac{\Delta t}{\omega}\left(\nabla \cdot \frac{\partial a}{\partial(\nabla\phi_i)} - \frac{\partial a}{\partial \phi_i}\right)^{n+1} + \lambda^{n+1} = \phi_i^n - \frac{\Delta t}{\omega\epsilon^2}\left(\frac{\partial w}{\partial \phi_i}\right)^{n}, \quad i = 1, \dots, p, \tag{6.13a}$$

$$\sum_{i=1}^{p} \phi_i^{n+1} = 1. \tag{6.13b}$$

The spatial discretisation of the gradient free energy component (2.30) is given by

$$\left(\nabla\cdot\frac{\partial a}{\partial(\nabla\phi_i)}-\frac{\partial a}{\partial\phi_i}\right)_{rs}\simeq -4s\sum_{j\neq i}\sigma_{ij}\left\{\phi_{i_{rs}}\left(\frac{(\Delta_{0x}\phi_{j_{rs}})^2}{(\Delta x)^2}+\frac{(\Delta_{0y}\phi_{j_{rs}})^2}{(\Delta y)^2}\right)\right.$$
$$\left.-\phi_{j_{rs}}\left(\frac{\Delta_{0x}\phi_{i_{rs}}\Delta_{0x}\phi_{j_{rs}}}{(\Delta x)^2}+\frac{\Delta_{0y}\phi_{i_{rs}}\Delta_{0y}\phi_{j_{rs}}}{(\Delta y)^2}\right)\right\}$$
$$-2s\sum_{j\neq i}\sigma_{ij}\left\{\phi_{i_{rs}}\phi_{j_{rs}}\left(\frac{\phi_{j_{r+1s}}-2\phi_{j_{rs}}+\phi_{j_{r-1s}}}{(\Delta x)^2}+\frac{\phi_{j_{rs+1}}-2\phi_{j_{rs}}+\phi_{j_{rs-1}}}{(\Delta y)^2}\right)\right.$$
$$\left.-\phi_{j_{rs}}^2\left(\frac{\phi_{i_{r+1s}}-2\phi_{i_{rs}}+\phi_{i_{r-1s}}}{(\Delta x)^2}+\frac{\phi_{i_{rs+1}}-2\phi_{i_{rs}}+\phi_{i_{rs-1}}}{(\Delta y)^2}\right)\right\},$$
$$i=1,\dots,p,\qquad r=1,\dots,N_x,\qquad s=1,\dots,N_y,\quad (6.14)$$

with $\Delta_{0x}\phi_{i_{rs}}=(\phi_{i_{r+1s}}-\phi_{i_{r-1s}})/2$ and $\Delta_{0y}\phi_{i_{rs}}=(\phi_{i_{rs+1}}-\phi_{i_{rs-1}})/2$. As for the simple phase field model, periodic boundary conditions are assumed.

6.3.2 Nonlinear multigrid solver

Spatial discretisation of scheme (6.13) yields a nonlinear system of equations that has to be solved at each time step. To solve this nonlinear system, a multigrid solver is implemented based on the Full Approximation Scheme (FAS) [16], which was described in some detail in Section 6.2.3. For the intergrid operators, full weighting is chosen as the restriction operator for the residual, injection for the restriction of the solution, and bilinear interpolation as the prolongation operator for both the solution and the residual. The convergence criterion of the multigrid cycle uses the norm of the residual scaled by the norm of the right-hand side. The smoothers are implemented using the collective Gauss-Seidel method. This method sweeps over all grid points in a lexicographic order, solving the spatially discretised equations (6.13) for all phase field variables and the function λ simultaneously. Therefore, at each grid point, a small nonlinear system of $p+1$ equations in $p+1$ unknowns has to be solved. For the solution of this nonlinear system, an inner Newton iteration is employed. The equations at a single grid point can be written in the form

$$N(\boldsymbol{\phi}_\lambda^{n+1})=B, \quad (6.15)$$

where $\boldsymbol{\phi}_\lambda$ is the vector-valued order parameter $\boldsymbol{\phi}$ extended with the unknown λ. The Newton iteration can now be formulated as

$$\boldsymbol{\phi}_\lambda^{k+1}=\boldsymbol{\phi}_\lambda^{k}+J_N^{-1}\left(B-N(\boldsymbol{\phi}_\lambda^{k})\right),\qquad k=0,1,2,\dots \quad (6.16)$$

with J_N the Jacobian of the nonlinear operator N. Remark that B remains constant throughout the Newton iterations: it is the explicit part of the discretisation scheme. Iterations (6.16) are performed until the norm of the residual of (6.15), scaled by the norm of the right-hand side B, falls under the desired accuracy. The dense linear system involved in equation (6.16) is solved with a direct algorithm, namely Gaussian elimination with partial pivoting. This type of solver scales with the cube of the number of unknowns, which is here $O((p+1)^3) \simeq O(p^3)$. For large, realistic grain growth simulations, ideally, thousands of phase field variables are included and thus $p \gg 1000$. However, at every grid point, only a few phase field variables are active [45, 65, 129, 133] and have to be taken into account. Therefore, with an adaptation of the multigrid solver to the tracking of the active phase field variables, the use of a direct solver should not be problematic. For the coarse-grid discretisation, the direct coarse-grid analogue of the fine-grid operator is implemented. The associated system is solved with the same direct solver as in the smoothers.

Once the smoother is defined, the optimal cycle type and number of presmoothing and postsmoothing steps can be determined. Table 6.3 shows the convergence rate, the number of cycles and the execution time, averaged over 10 time steps, for different multigrid cycle types on a $[0,1] \times [0,1]$ domain with $p = 3$, $s = 1/(6.4)^2$, $\Delta x = \Delta y = 1/256$, and $\Delta t = 0.1$. The other parameter values are chosen $\omega = 1$, $\epsilon = 1$, and $\sigma_{ij} = 1.0$, for all $i, j = 1, \ldots, p$ with $i \neq j$. Unless mentioned otherwise, the latter parameter values are assumed for all following simulation tests. Also, each time, a tolerance of $\tau = 10^{-8}$ is enforced. The structure is initialised with a sharp interface Voronoi tessellation: the domain is decomposed into p different cells and each cell is assigned to a different phase field variable ϕ_i. The cycle type is written as $V(\nu_1, \nu_2)$ or $W(\nu_1, \nu_2)$, with ν_1 and ν_2 respectively the number of presmoothing and postsmoothing iterations. Overall, the $V(1,1)$-cycle turned out to be the most efficient.

Table 6.3: Convergence rate, number of cycles and execution time for different multigrid cycles averaged over 10 time steps for $p = 3$, $s = 1/(6.4)^2$, $\Delta t = 0.1$, and $\Delta x = \Delta y = 1/256$.

Cycle type	Convergence rate	Number of cycles	Time
V(1,1)	0.2067	11.4	3.2490 s
V(2,2)	0.0869	7.4	4.0150 s
W(1,1)	0.0389	6.0	4.5190 s
W(2,2)	0.0031	3.1	4.4810 s

6.3.3 Convergence properties

We will now verify the convergence properties of the multigrid solver as a function of the grid spacing $\Delta x = \Delta y$, and as a function of the number of phase field variables p. Table 6.4 shows the execution time and the convergence factor averaged over ten time steps of simulations on a $[0,1] \times [0,1]$ domain with $\Delta x = 1/64$, $1/128$, $1/256$ and $1/512$, for $s = 1/(6.4)^2$, $\Delta t = 0.1$ and $p = 10$. It can be seen that the convergence properties of the multigrid solver are independent of Δx. The execution time scales linearly with the number of spatial unknowns. This is confirmed by both Fig. 6.7 and Fig. 6.8. In Fig. 6.7, the relative execution time per time step is plotted during the first ten time steps of the simulations. The execution times of the simulation for $\Delta x = 1/64$ are chosen as a reference. The sharp interfaces of the initial Voronoi tessellation cause a large initial residual, which increases for larger systems. Therefore, the initial time step takes relatively more execution time for larger systems. Figure 6.8(a) shows the behaviour of the residual norms scaled by the norm of the right-hand side during the first time step, while Fig. 6.8(b) shows the behaviour during the tenth time step. It can be seen that the multigrid convergence is grid size independent. The larger initial residual for smaller grid spacings is reflected in the graph of Fig. 6.8(a).

Table 6.4: Average execution time and convergence factor for different numbers of spatial unknowns during the first ten time steps of simulations on a $[0,1] \times [0,1]$ domain with $\Delta x = \Delta y = 1/64$, $1/128$, $1/256$ and $1/512$, for $s = 1/(6.4)^2$, $\Delta t = 0.1$ and $p = 10$. $V(1,1)$-cycles are applied.

Δx	System size	Execution time	Convergence factor
1/64	$10 \times 64 \times 64$	2.2250 s	0.3188
1/128	$10 \times 128 \times 128$	9.5110 s	0.3318
1/256	$10 \times 256 \times 256$	38.5610 s	0.3366
1/512	$10 \times 512 \times 512$	153.3250 s	0.3382

Next, we study the convergence properties and the execution time as a function of p. In Table 6.5, the execution time is listed for different values of p, averaged over the first ten time steps of simulations on a $[0,1] \times [0,1]$ domain, with $s = 1/(6.4)^2$, $\Delta x = \Delta y = 1/256$ and $\Delta t = 0.1$. It can be seen that the computation work increases slightly faster than $O(p^2)$. Note that the smoothers include a direct solver which scales as $O((p+1)^3) \simeq O(p^3)$; the execution times listed in Table 6.5 are thus better than expected, probably due to the relatively small number of phase field variables of p used in our experiments. Figure 6.9 shows the relative execution time per time step for the first ten time steps of these simulations. Here, the execution time for $p = 5$ is taken as a point of reference. Again, the first time step takes more processing time as the initial sharp interfaces cause a larger initial residual. The increase in execution time at the tenth time step for $p = 20$ is related to the disappearance of a smaller grain. Figure 6.10 illustrates the evolution of

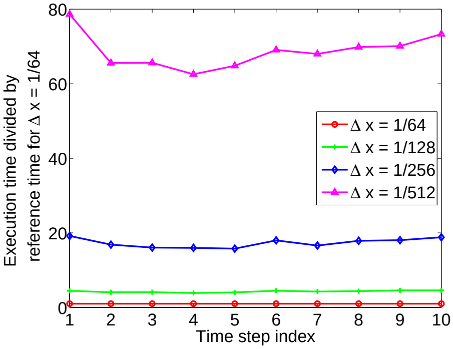
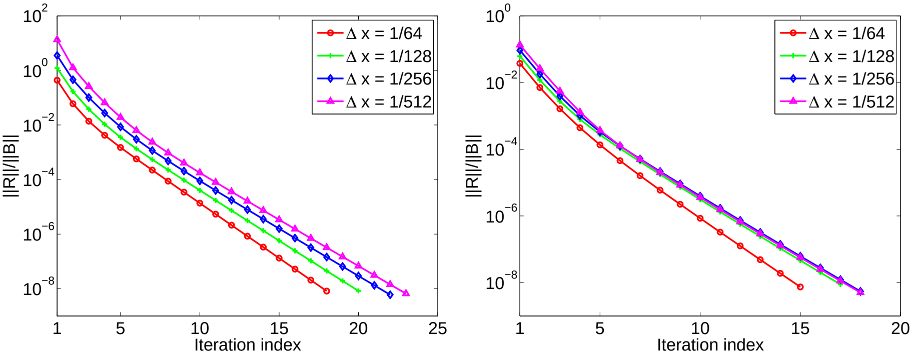


Figure 6.7: Execution time per time step during the first ten time steps of simulations on a $[0, 1] \times [0, 1]$ domain with $\Delta x = \Delta y = 1/64$, $1/128$, $1/256$, and $1/512$, for $s = 1/(6.4)^2$, $\Delta t = 0.1$ and $p = 10$. $V(1, 1)$-cycles are applied.



(a) Convergence during first time step

(b) Convergence during tenth time step

Figure 6.8: Convergence behaviour during (a) the first and (b) the tenth time step of simulations on a $[0, 1] \times [0, 1]$ domain with $\Delta x = \Delta y = 1/128$, $1/256$ and $1/512$, for $s = 1/(6.4)^2$, $\Delta t = 0.1$ and $p = 10$. For larger systems, more multigrid cycles are required during the first time step to reduce the larger initial scaled residual. $V(1, 1)$-cycles are applied.

the scaled residual norms during the first and the tenth time step. It can be seen that convergence deteriorates as p increases. The similar convergence behaviour for $p = 10$ and $p = 15$ in Fig. 6.10(b) can be attributed to coincidence.

It should be noted that the initial extra computing work is less significant for smaller values of Δt. Therefore, a solution to the problem of the initial work is to employ a smaller time step size at the start of a simulation. Afterwards, a larger time step size can be used. Otherwise, it would be impossible to start simulations

Table 6.5: Average execution time and convergence factor for different numbers of phase field variables p during the first ten time steps of simulations on a $[0, 1] \times [0, 1]$ domain, with $s = 1/(6.4)^2$, $\Delta x = \Delta y = 1/256$ and $\Delta t = 0.1$. $V(1, 1)$-cycles are applied.

p	System size	Execution time	Convergence factor
5	$5 \times 256 \times 256$	8.0620 s	0.2497
10	$10 \times 256 \times 256$	38.5610 s	0.3366
15	$15 \times 256 \times 256$	89.0100 s	0.3473
20	$20 \times 256 \times 256$	193.3280 s	0.4342

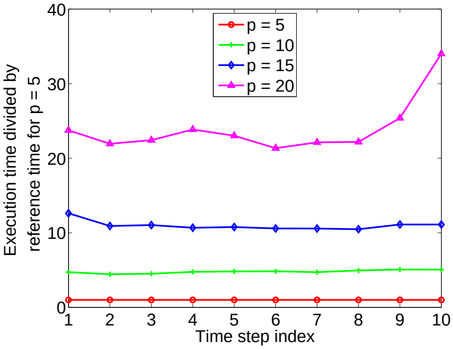


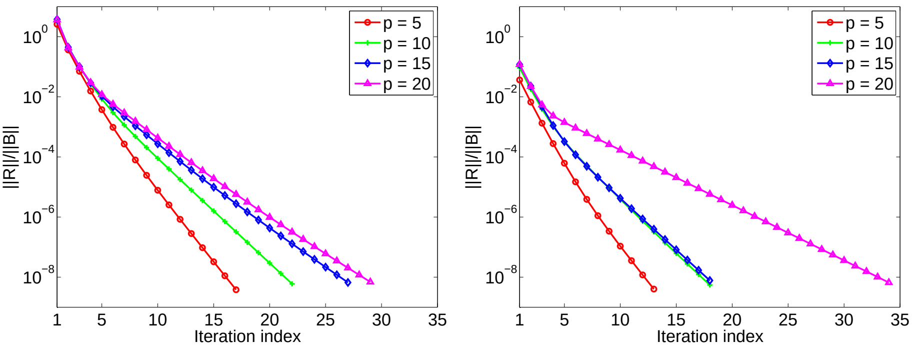
Figure 6.9: Execution time per time step during the first ten time steps of simulations on a $[0, 1] \times [0, 1]$ domain for $p = 5$, 10, 15, and 20, with $s = 1/(6.4)^2$, $\Delta x = \Delta y = 1/256$ and $\Delta t = 0.1$. $V(1, 1)$-cycles are applied.

for very large systems because the initial iterations would take too much time. A possible line of thought is to initialise the structure with smooth interfaces as opposed to a decomposition with sharp interfaces.

6.3.4 Comparison with explicit time stepper

In this section, we will briefly compare the nonlinear multigrid solver with an explicit time stepper. By using an explicit scheme instead of a semi-implicit or a fully implicit scheme, severe restrictions are made with regard to the time step size. However, the systems that need to be solved when using an implicit scheme are larger and more complex, which costs more execution time.

The employed explicit discretisation scheme is the forward Euler method. After applying this scheme to model equations (2.27), the discretised system can be



(a) Convergence during first time step

(b) Convergence during tenth time step

Figure 6.10: Convergence behaviour during (a) the first and (b) the tenth time step of simulations on a $[0,1] \times [0,1]$ domain for $p = 5$, 10, 15, and 20, with $s = 1/(6.4)^2$, $\Delta x = \Delta y = 1/256$ and $\Delta t = 0.1$. $V(1,1)$-cycles are applied.

written as:

$$\phi_i^{n+1} = \phi_i^n + \frac{\Delta t}{\omega}\left(\nabla \cdot \frac{\partial a}{\partial(\nabla\phi_i)} - \frac{\partial a}{\partial \phi_i}\right)^n - \frac{\Delta t}{\omega\epsilon^2}\left(\frac{\partial w}{\partial \phi_i}\right)^n - \lambda^{n+1}, \tag{6.17a}$$

$$i = 1, \ldots, p,$$

$$\sum_{i=1}^{p} \phi_i^{n+1} = 1, \tag{6.17b}$$

with the spatial discretisation of the gradient free energy component (2.30) performed as in (6.14). At every time step, system (6.17) has to be solved for the unknowns $\phi_1^{n+1}, \ldots, \phi_p^{n+1}$ and λ^{n+1}. The system matrix is however constant throughout a simulation and its analytically computed inverse is included as such in the solver. The solution of the system is thus reduced to a matrix-vector product.

Table 6.6 contains the execution time required for respectively the explicit and the semi-implicit time stepper to reach the simulation time point $t = 1.0$ as a function of the chosen time step size. For the chosen parameters $\Delta x = \Delta y = 1/256$ and $s = 1/(25.6)^2$, time step size $\Delta t = 0.001$ is close to the numerical stability limit for the explicit scheme, while for the semi-implicit scheme, $\Delta t = 1.0$ is still a stable choice. Simulations show that the semi-implicit solver is more than five times as fast for this particular parameter set. Furthermore, it can be remarked that the stability constraint for the explicit scheme is strongly dependent on the grid size

Δx, while the grid size dependent part is treated implicitly by the semi-implicit scheme included in the multigrid solver.

Table 6.6: Execution time for the explicit and the semi-implicit solver to reach $t = 1.0$, on a $[0, 1] \times [0, 1]$ domain, with $\Delta x = \Delta y = 1/256$ and $s = 1/(25.6)^2$, for different time step sizes.

Time step Δt	Explicit solver	Semi-implicit solver
0.001	474.35 s	6466.10 s
0.01	–	1512.90 s
0.1	–	279.78 s
1.0	–	92.16 s

At the beginning of a grain growth simulation, when the microstructure consists of a myriad of small grains, it is important to capture the correct growth dynamics. Therefore, the choice of a small time step size Δt, in combination with the explicit solver, is appropriate. Later in the simulation, when the grain growth dynamics are coarser, a large time step size is more advantageous, which is enabled by the use of the multigrid solver for the semi-implicit discretisation scheme.

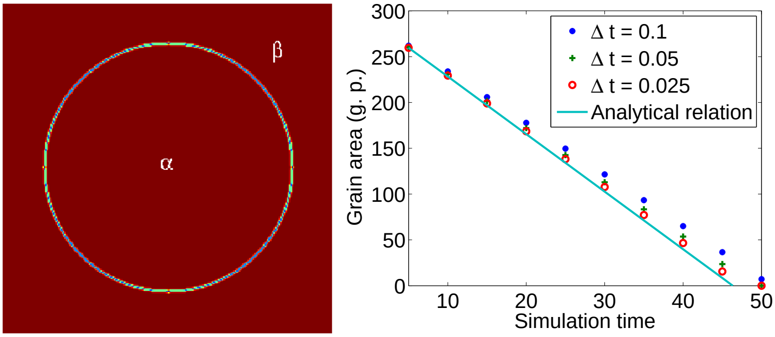
6.4 Applications

6.4.1 Evolution of a circular grain in a matrix

For curvature driven growth [54], the evolution of the area of a circular grain of phase α in a matrix of phase β, illustrated in Fig. 6.11(a), is known to satisfy

$$A_\alpha(t) = A_{\alpha,0} - 2\pi\mu\sigma_{\alpha\beta}t, \tag{6.18}$$

where $A_{\alpha,0}$ is the initial grain area, for $t = 0$. This relation corresponds to growth law (2.2). We test the nonlinear multigrid solver by performing simulations of the evolution of a single grain on a $[0, 1] \times [0, 1]$ domain with $p = 2$, $\omega = 1/\mu = 1$, $\epsilon = 1$, $\sigma_{\alpha\beta} = 1$, $\Delta x = \Delta y = 1/256$, $s = 1/(25.6)^2$ and different time step sizes $\Delta t = 0.1$, 0.05 and 0.025. To study the evolution of the grain, the grain area is computed as the sum of $\phi_\alpha(\Delta x)^2/s$ over all grid points. Figure 6.11(b) shows the shrinkage of the grain area during the simulations for the different values of Δt. It can be seen that the results of the simulations correspond well to analytical relation (6.18). For smaller Δt, the simulations results match the analytical relation better.



(a) Initial structure

(b) Evolution

Figure 6.11: (a) Circular grain of phase α within a matrix of phase β. (b) Computed evolution of the area of a circular shaped grain in a matrix in a simulation on a $[0,1] \times [0,1]$ domain with $p = 2$, $\omega = 1/\mu = 1$, $\epsilon = 1$, $\Delta x = \Delta y = 1/256$, and $s = 1/(25.6)^2$, with different time step sizes $\Delta t = 0.1$, 0.05 and 0.025, compared to analytical relation (6.18).

6.4.2 Grain growth

Another application of the multigrid solver is the evolution of a polycrystalline microstructure. A grain growth simulation is performed on a $[0,1] \times [0,1]$ domain, with $p = 15$, $\Delta x = \Delta y = 1/256$, $s = 1/(25.6)^2$, $\Delta t = 0.1$, and isotropic surface energies $\sigma_{ij} = 1.0$. To initialise the structure, every grid point is assigned to a unique phase field variable ϕ_i, according to a uniform distribution. The domain is thus decomposed into a number of grains that equals the grid size. To visualise the grain growth, we employ a sharp interface representation, where every grid point is assigned to the phase field variable with the highest value at that point. Figures 6.12(a), 6.12(b) and 6.12(c) show the microstructure at time points $t = 2$, $t = 20$ and $t = 160$ respectively.

A second grain growth simulation is performed on a $[0,1] \times [0,1]$ domain, with $\Delta x = \Delta y = 1/256$, $s = 1/(25.6)^2$ and $p = 36$. Simulation experiments described in [35] indicate that it is possible to use a finite, but large number of phase field variables to study grain growth kinetics. In particular, for a short amount of grain growth simulation, it is found that there is no distinguishable difference between grain size distributions obtained from simulations with $p \geq 36$. A small time step size $\Delta t = 0.004$ is chosen to capture all the dynamics of the start of the grain growth. The simulation shows a convergence factor of about 0.09 during these first time steps. Figure 6.13(a) shows the microstructure at time point $t = 0.8$: the initial grains have already grown into small grains. During grain growth, the smaller grains disappear, while the larger grains grow. This is already clear at

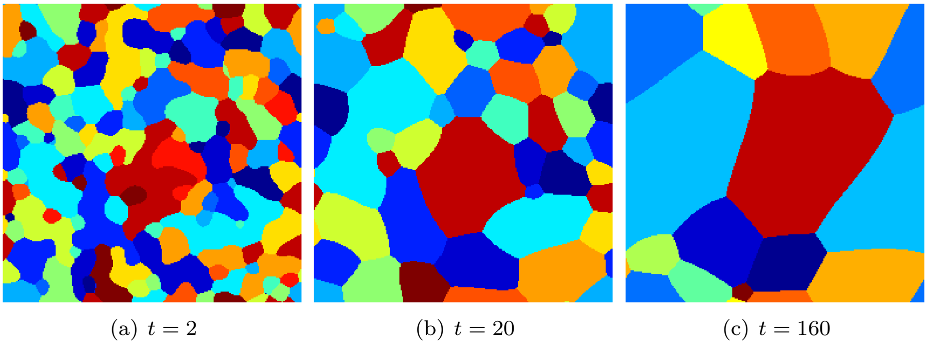


Figure 6.12: Results of a simulation on a $[0,1] \times [0,1]$ domain, with $p = 15$, $\Delta x = \Delta y = 1/256$, $s = 1/(25.6)^2$, and isotropic $\sigma_{ij} = 1.0$: (a) Microstructure at $t = 2$: several grains are coalescing. (b) Microstructure at $t = 20$: some larger grains result from the grain coalescence. (c) Microstructure at $t = 160$: only a few grains remain.

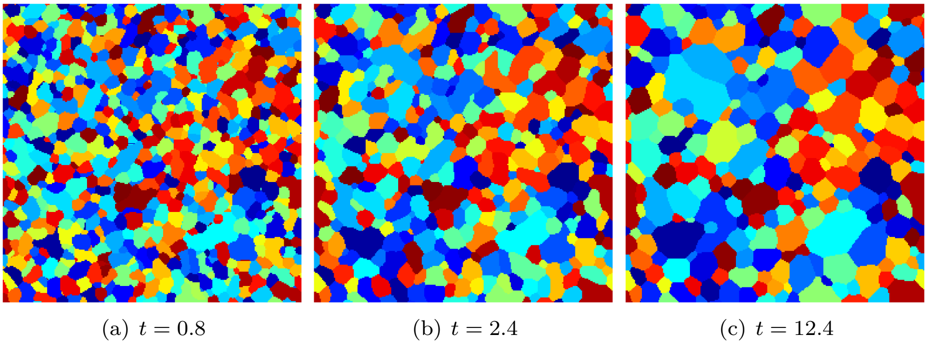


Figure 6.13: Results of a simulation on a $[0,1] \times [0,1]$ domain, with $p = 36$, $\Delta x = \Delta y = 1/256$, $s = 1/(25.6)^2$ and isotropic $\sigma_{ij} = 1.0$ at (a) $t = 0.8$, (b) $t = 2.4$, and (c) $t = 12.4$.

$t = 2.4$, depicted in Fig. 6.13(b), where many grains have disappeared in favour of larger grains. From $t = 2.4$, a larger time step size is chosen: $\Delta t = 0.1$. For this time step size, the convergence factor increases to about 0.53.

6.5 Conclusion

In this chapter, the concepts of multigrid methods were introduced. We described a standard linear multigrid solver, as well as a nonlinear FAS (Full Approximation Scheme) multigrid solver. Three different time discretisation schemes were applied to a simple phase field model, which was obtained by reducing multi-phase field model (2.27) for two phase field variables. It was illustrated how the multigrid solver possesses convergence properties that are independent of the spatial grid resolution and how the execution time of such a solver can scale linearly with the number of unknowns.

In Section 6.3, a nonlinear multigrid solver was presented to solve the full multi-phase field model. The goal of the solver is to perform grain growth simulations in two dimensions in an efficient way. Until now, only explicit discretisation schemes have been used to solve this particular highly complex and nonlinear phase field model (see e.g. [99]). In Section 6.3.1, a semi-implicit discretisation scheme is applied that treats the gradient energy part of the model implicitly and the potential part explicitly. It is chosen to be a two-step scheme because of the memory complexity of the model.

To solve the equations describing the discretised system, a nonlinear multigrid solver based on the FAS scheme is constructed. Experiments with this solver show that the convergence rates are independent of the grid size. However, the convergence properties of the multigrid solver depend on the number of involved phase field variables: for larger numbers, the convergence deteriorates. In spite of its optimal convergence properties, it remains an open question whether the presented multigrid solver is faster than classical iterative solvers. The implementation of a solver combining a Newton iteration with an iterative solver should resolve this issue.

To validate the implementation of the FAS solver, the results of specific test cases are studied. In the first application, the simulation results for the shrinkage rate of a circular grain within a matrix phase shows very good agreement with the existing analytical relation. In the second application, grain growth simulations are performed.

Chapter 7

Pinning effect of second-phase particles on grain growth

7.1 Introduction

The addition of alloying elements, which leads to the formation of finely dispersed second-phase particles, is a common technique to control the grain size of a microstructure. 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 [86, 87], continuum field model (2.39) is presented for simulating grain growth in materials containing small incoherent second-phase particles that 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. Simulations for three-dimensional systems [83, 122, 88] show that the pinning effect of second-phase particles is significantly weaker than in two-dimensional systems. Therefore, it is important that predictive computer models reflect the three-dimensional nature of Zener pinning (see also the discussion in Section 2.2.2). Unfortunately, the computational requirements of three-dimensional phase field simulations are memory consuming and computationally intensive. In this chapter, we illustrate the applicability of the bounding box algorithm presented in Chapter 5. With the bounding box algorithm, three-dimensional simulation results can be obtained for relevant comparison with experimental data without excessive computational requirements.

To gain more insight in the dependence of the pinning force of a particle distribution on the shape of the particles, we perform three-dimensional phase field simulations of grain growth in systems with spheroid particles, for different aspect ratios and volume fractions of the particles. The parallelised implementation of the bounding box algorithm of Section 5.5 is applied, which makes it feasible to run three-dimensional simulations with a large amount of grains.

In Section 7.2, the set-up of the simulation experiments is described. Section 7.3 discusses the simulation results. Depending on the applied volume fraction and the shape of the second-phase particles, their pinning effect on grain growth has a different strength. Nevertheless, the final grain size has a volume fraction dependence of the form $1/f_{\mathrm{V}}^{0.93}$, following theoretical predictions assuming random intersections between grain boundaries and particles. In Section 7.4, a generalised Zener relation with a prefactor depending on the aspect ratio is proposed. Our results are compared with the results of other studies in Section 7.5. The conclusions of this chapter are formulated in Section 7.6.

7.2 Simulation parameters

7.2.1 Phase field model, discretisation and set-up

To simulate grain growth in the presence of second-phase particles, continuum field model (2.39) is used. This model was presented and tested in [86, 87] for grain growth simulation in materials containing small incoherent second-phase particles that are constant in time. The model is discretised with second-order central differences for the spatial derivative and the SBDF1 scheme is applied to the time derivative, as in system (3.12).

All simulations are performed on an equispaced grid with dimensions 256 × 256 × 256, and discretisation spacings $\Delta t = 0.2$ and $\Delta x = 1$. Since we are mostly interested in the final grain size where grain growth is arrested, which is independent of the grain boundary energy and mobility as can be inferred from Zener relation (2.4), the exact values of the model parameters L and κ are not important. Therefore, we take similar values as in previous studies [88, 129], namely $\kappa = 0.5$, $L = 1$, and $m = 1$. The choice of these parameters values is justified in [87] and is based on considerations of both accuracy and efficiency. Periodic boundary conditions are applied, as in [85].

The simulations start from microstructures initialised with Algorithm 5.2. The microstructures are constructed for 25000 phase field variables. After 1000 time steps, about 1000 of the 25000 possible orientations remain. The parallelised implementation of the bounding box algorithm of Section 5.5 is applied and run on 3 nodes of a computer cluster, which are interconnected with an Infiniband network and each contain 4 processors, making a total of 12 processors.

7.2.2 Shape, size and volume fraction of the particles

The purpose of the simulations in this chapter is to study the effect of particle shape on Zener pinning. Simulations are run for spheroid particles (see Section 5.2.3) with different aspect ratios r_a and for different volume fractions $f_\mathrm{V} = 5\%$, 6%, 8%, 10%, 11%, and 12%. More specifically, three different aspect ratios are applied: $r_a = 1$, $r_a = 2$, and $r_a = 3$. Since in these three cases the inequality $c \geq a$ holds, we rename the axis radius c to l, the long axis radius, and the axis radius a to s, the short axis radius. The volume of the three particle shapes is intended to be approximately the same. The formula for the volume V_S of a spheroid is

$$V_\mathrm{S} = \frac{4}{3}\pi l s^2. \tag{7.1}$$

In combination with the constraint that $l \geq s \geq 3$ grid points (g.p.) because of the finite width of the boundaries in the simulations [88], this volume formula yields spheroids with the following dimensions:

- for $r_a = 1$: $l = s = 4.3$ g.p.,
- for $r_a = 2$: $l = 6.8$ g.p. and $s = 3.4$ g.p.,
- for $r_a = 3$: $l = 9$ g.p. and $s = 3$ g.p.

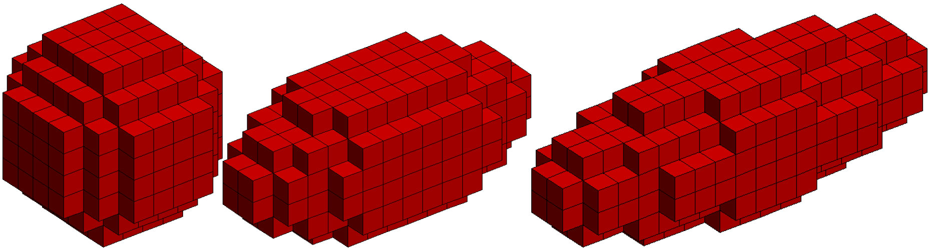
It is however impossible to work with fractions of grid points to represent the particles. After discretisation, the radii of the spheroid particles become:

- for $r_a = 1$: $l = s = 4$ g.p.,
- for $r_a = 2$: $l = 7$ g.p. and $s = 3$ g.p.,
- for $r_a = 3$: $l = 9$ g.p. and $s = 3$ g.p.

The volume of the spheroid particles with $r_a = 3$ thus ends up to be slightly larger than the volume of the other two particle types. Figure 7.1 illustrates the three particle shapes after discretisation.

In this work, a particle is allowed to be oriented with its long axis either along the x-, the y-, or the z-axis of the system. The orientations of the particles within the same microstructure are approximately equally distributed over the three possible axes. If the particles were allowed to be oriented along every possible direction, they would differ in volume due to additional discretisation effects. We assume that the orientation distribution along the three axes is a reasonable approximation to a uniform orientation distribution.

For every parameter combination of aspect ratio and volume fraction, three simulation runs are executed, over which all studied characteristics are averaged.



(a) $r_a = 1$ (b) $r_a = 2$ (c) $r_a = 3$

Figure 7.1: Three different spheroid shapes of the second-phase particles after discretisation, with approximately equal volume. They are characterised by the ratio r_a between the long axis radius l and the short axis radius s.

The initial grain structure and particle distribution is entirely different for every simulation run. There is however a correspondence between the simulation runs for the same aspect ratio or for equal volume fraction: the particles are located at similar locations, which allows one to compare the resulting microstructures visually.

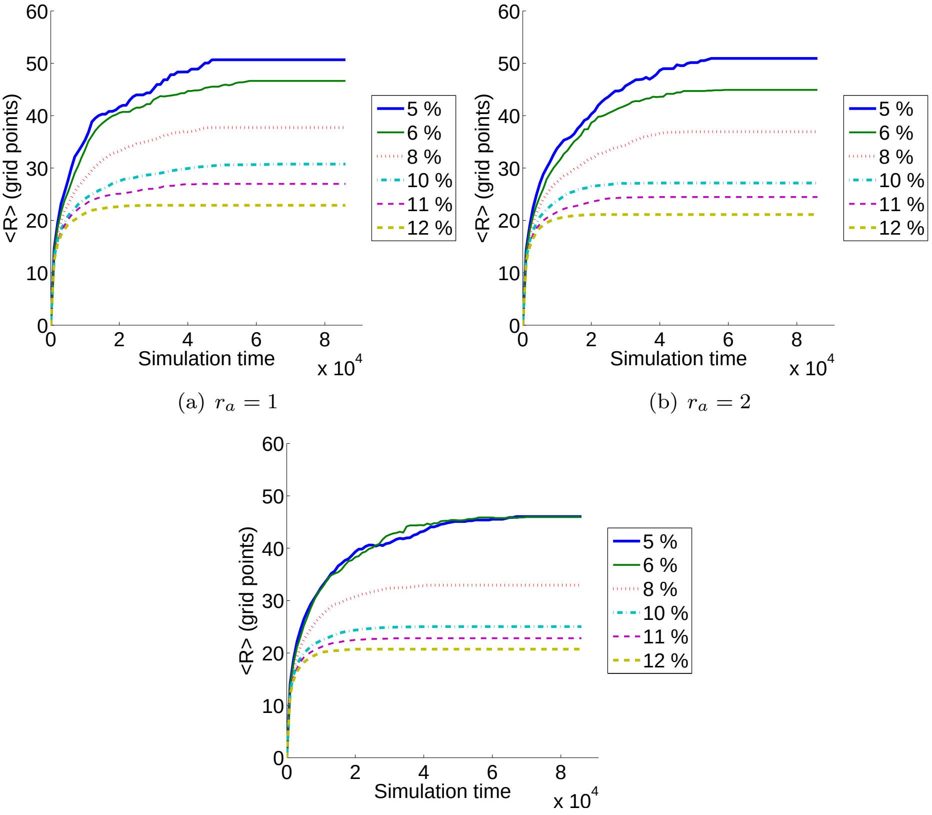
Since particle clusters also occur in real materials, no effort was made to separate the second-phase particles. Some particles may therefore overlap. As in [4], we assume that the influence of this small amount of particle clusters on the pinning effect is small.

7.3 Effect of spheroid particles

7.3.1 Growth kinetics

Figure 7.2 depicts the time evolution of the mean grain radius for different volume fractions f_{V}, for $r_a = 1$, $r_a = 2$, and $r_a = 3$ separately. The mean grain radius $\langle R \rangle$ is computed as the radius of a sphere with a volume equivalent to the mean grain size of the microstructure. For each of the three particle types, the pinning effect proves to be stronger for larger volume fractions: grain growth stops earlier and the limiting grain size is smaller. This is in agreement with results from earlier simulation studies for spherical particles [122, 83] and experimental data.

In Fig. 7.3, the time evolution of the mean grain radius is shown for the different aspect ratios r_a, for each studied volume fraction f_{V} separately. Except for the parameter combination of $f_{\mathrm{V}} = 6\%$ and $r_a = 3$, for every volume fraction grain growth stops earlier and at a smaller mean grain size for increasing aspect ratio.



(a) $r_a = 1$

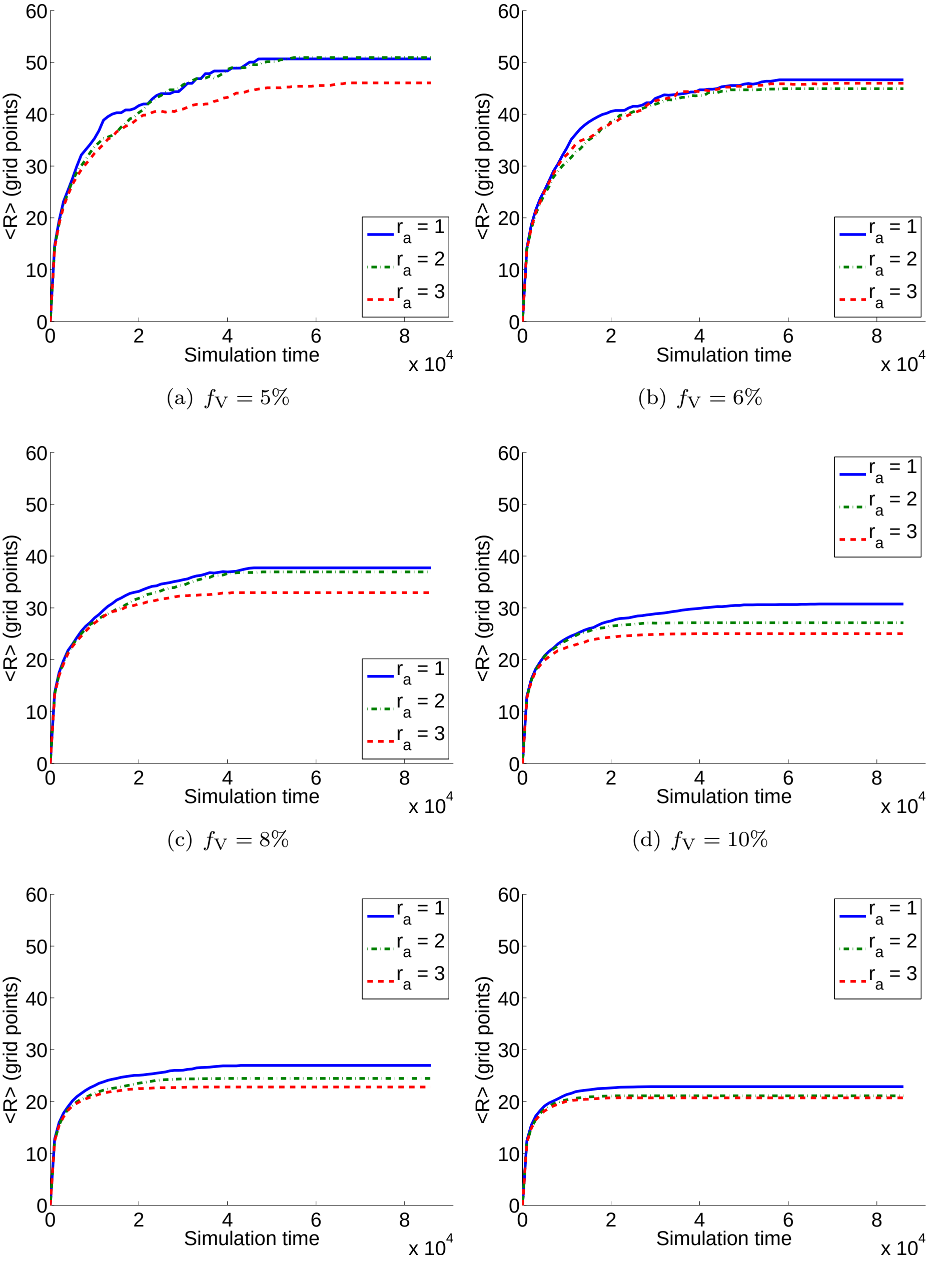
(b) $r_a = 2$

(c) $r_a = 3$

Figure 7.2: Time evolution of the mean grain radius $\langle R \rangle$ for different volume fractions f_{V}, for each studied aspect ratio r_a separately.

This result is qualitatively in agreement with the two-dimensional phase field simulations of [22].

In one simulation run, for the parameter combination of $f_{\mathrm{V}} = 5\%$ and $r_a = 1$, grain growth was not arrested by the particles and the final microstructure only contained one grain, while in another run, only nine grains were present in the microstructure at grain growth stagnation. To preserve a smooth graph on Fig. 7.2(a) and Fig. 7.3(a), we removed the corresponding data points. Otherwise, the graphs in question would go up, rather than stabilise at a limiting value. According to [18], abnormal grain growth is possible in the presence of a stable particle distribution for small grain sizes. Another reason, more plausible for the present case, might be the statistical nature of the results in combination with the finite dimensions and periodic boundary conditions assumed in the simulations.



(a) $f_{\rm V} = 5\%$

(b) $f_{\rm V} = 6\%$

(c) $f_{\rm V} = 8\%$

(d) $f_{\rm V} = 10\%$

(e) $f_{\rm V} = 11\%$

(f) $f_{\rm V} = 12\%$

Figure 7.3: Time evolution of the mean grain radius $\langle R \rangle$ for different aspect ratios r_a, for each studied volume fraction $f_{\rm V}$ separately.

Table 7.1: Number of grains present in the microstructures when grain growth is stopped, for every tested parameter combination. Note that the limiting number of grains for the parameter combination $f_V = 5\%, r_a = 1$ originates from only one simulation run, instead of three for the other parameter combinations.

	$f_V = 5\%$	$f_V = 6\%$	$f_V = 8\%$	$f_V = 10\%$	$f_V = 11\%$	$f_V = 12\%$
$r_a = 1$	27	30	62	110	170	250
$r_a = 2$	30	41	65	159	212	313
$r_a = 3$	36	38	95	198	255	330

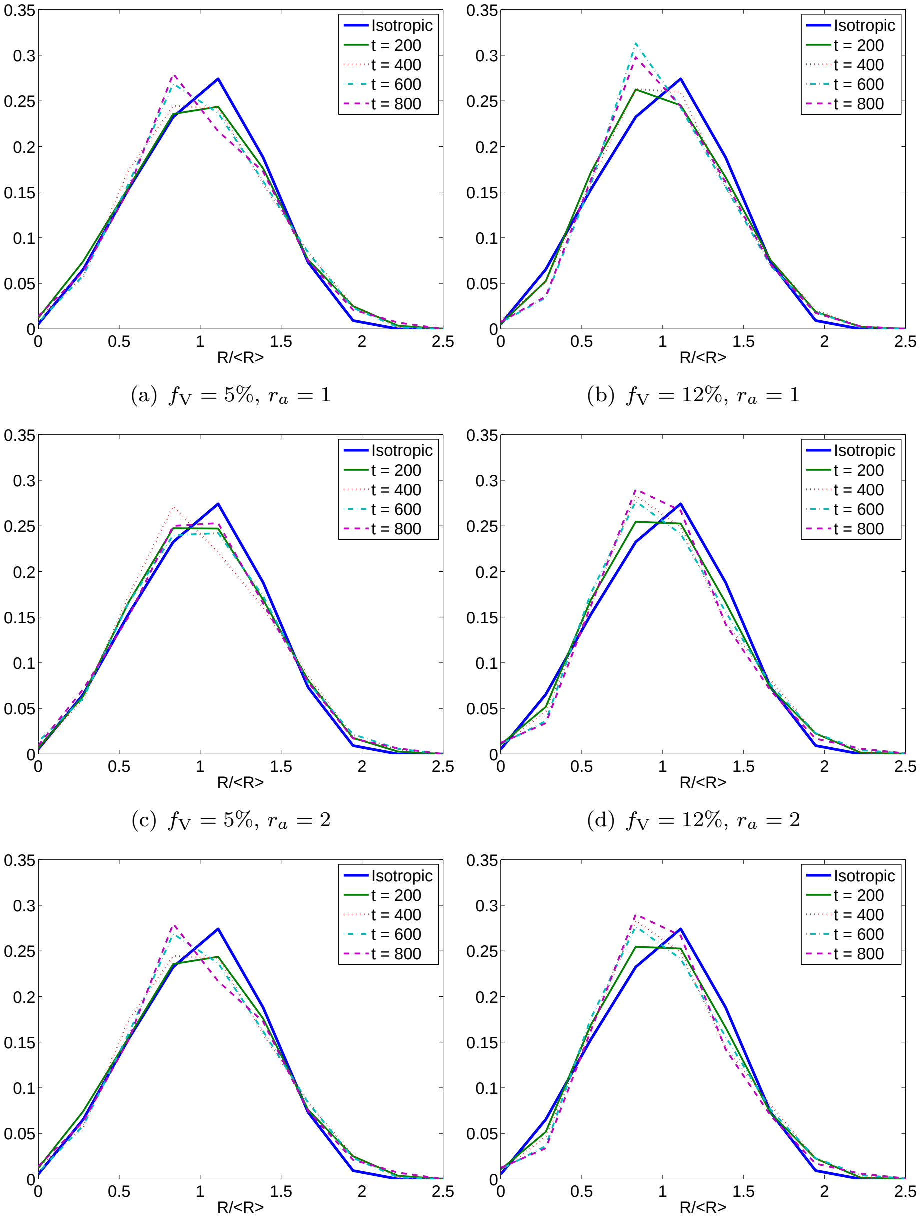
The latter reason also explains why in another simulation run, for the parameter combination of $f_V = 6\%$ and $r_a = 3$, grain growth stopped later than expected. This is reflected in Fig. 7.2(c), where the graphs corresponding to $f_V = 5\%$ and $f_V = 6\%$ almost coincide. Table 7.1 lists the number of grains present in the microstructures when grain growth is stopped, for every parameter combination. The number of grains remaining in a microstructure is in keeping with the limiting grain size: for a smaller limiting size, the limiting number of grains is larger.

7.3.2 Grain size distribution

Figure 7.4 illustrates the evolution of the grain size distribution for both the smallest and the largest tested volume fraction, namely $f_V = 5\%$ and $f_V = 12\%$, for the three studied aspect ratios separately. At the latest depicted time point $t = 800$, there are still at least 200 grains present in every microstructure. For all parameter combinations, the grain size distributions of the microstructures with second-phase particles are compared with the grain size distribution of a single-phase microstructure, which is time invariant [58]. As predicted by mean-field theories [54, 59, 1], the simulation results indicate that, when second-phase particles are present, the peak of the distribution shifts towards smaller grain sizes. Figure 7.4 shows that this is true regardless of the aspect ratio. Furthermore, there does not appear to be a significant influence of the aspect ratio on the distribution shape.

7.3.3 Microstructure evolution

The evolution of a microstructure is captured at time points $t = 200, 2000, 5000$, and 50000 on Fig. 7.5, for a volume fraction of $f_V = 8\%$ of spherical particles ($r_a = 1$). After $t = 50000$, no further grain boundary movement is observed. The microstructures on Fig. 7.5 are visualised with a diffuse interface representation: the values of the sum Ψ are displayed as computed with formula (4.28). Naturally,



(a) $f_{\mathrm{V}} = 5\%$, $r_a = 1$

(b) $f_{\mathrm{V}} = 12\%$, $r_a = 1$

(c) $f_{\mathrm{V}} = 5\%$, $r_a = 2$

(d) $f_{\mathrm{V}} = 12\%$, $r_a = 2$

(e) $f_{\mathrm{V}} = 5\%$, $r_a = 3$

(f) $f_{\mathrm{V}} = 12\%$, $r_a = 3$

Figure 7.4: Evolution of the grain size distribution obtained from simulations for $f_{\mathrm{V}} = 5\%$ and 12%, and for $r_a = 1$, 2, and 3, compared to the time invariant grain size distribution of a single-phase microstructure.

the computation of Ψ is now limited to the phase field values contained by the bounding boxes.

Figure 7.5(a) shows that at the beginning of the simulation, most particles are located at grain boundaries. Later in the simulation, many particles are located within the grains, as can be seen on Figs. 7.5(c) and 7.5(d). The latter figures illustrate the fact that the pinning effect of second-phase particles is much weaker in three-dimensional than in two-dimensional systems, where most of the particles are located at the grain boundaries when grain growth is stopped [88].

7.3.4 Cross-section

To gain more insight in the location of the particles in the microstructure, cross-sections through the middle of pinned structures are shown in Fig. 7.6 for simulations with $f_V = 5\%$ and 12%, and aspect ratios $r_a = 2$ and $r_a = 3$. The figures illustrate that the limiting grain size is smaller for higher volume fractions and for higher aspect ratio. Particles with $r_a = 3$ seem to make up a substantial part of the grain boundaries. At certain points in the microstructure, they have a significant influence on the grain boundary orientation: the grain boundaries follow their orientation in such a way that the particles constitute a part of the boundaries. In this way, the particles are more efficient in pinning grain boundaries compared to the particles with aspect ratio $r_a = 2$. The extra amount of grain boundary that has to be created for this reorientation appears to outweigh the amount of grain boundary that is removed by the particles with r_a smaller than 3.

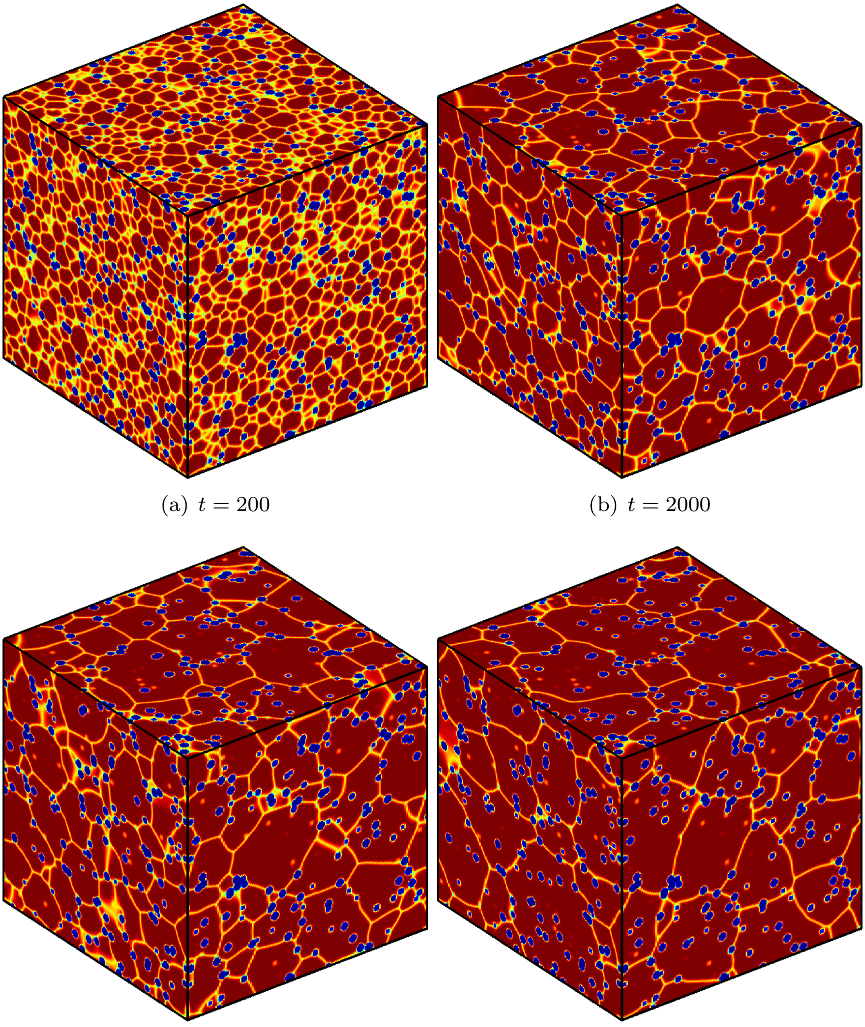
7.3.5 Particle location

At grain growth stagnation, many of the second-phase particles are located at grain boundaries. According to their location, the particles that intersect with grain boundaries can be subdivided into four types:

1. the particles present at interfaces between two grains (ϕ_2),
2. at junctions where three grains meet (ϕ_3),
3. at junctions where four grains meet (ϕ_4),
4. and at junctions where more than four grains meet (ϕ_n).

The total fraction of particles present at boundaries is ϕ_{tot}, with

$$\phi_{\text{tot}} = \phi_2 + \phi_3 + \phi_4 + \phi_n. \tag{7.2}$$



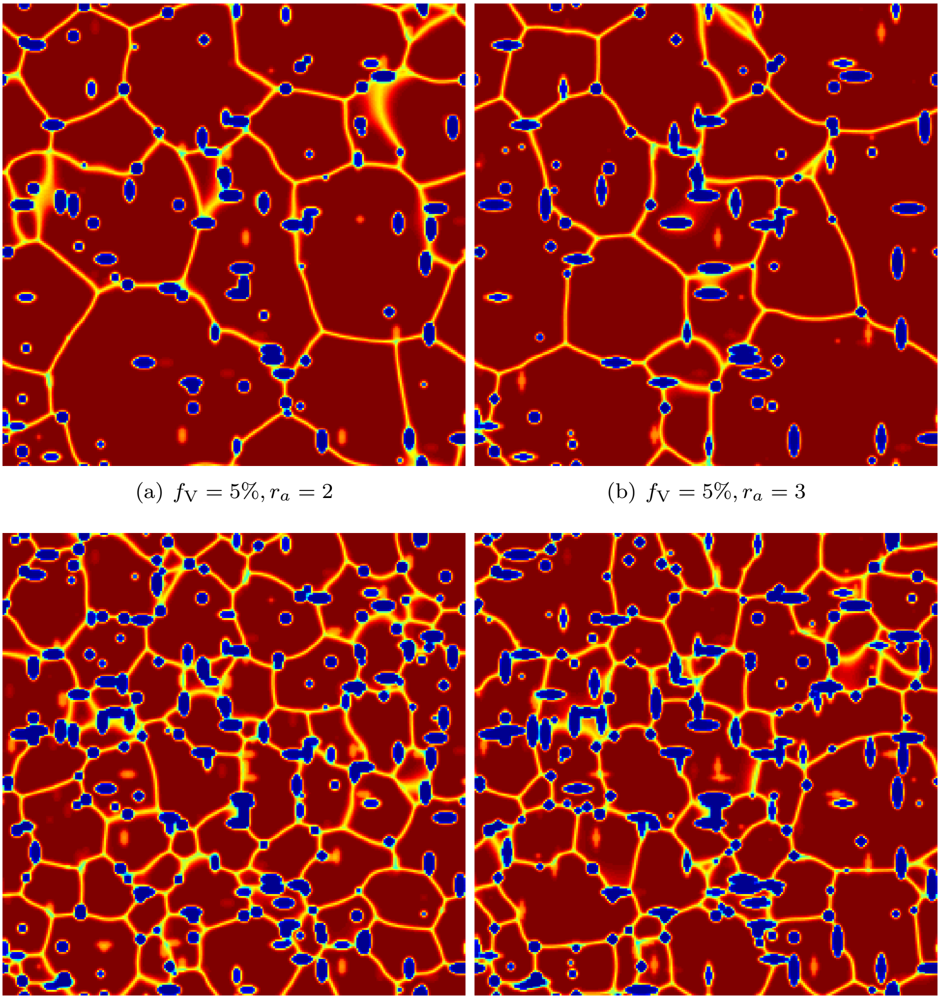
(a) $t = 200$

(b) $t = 2000$

(c) $t = 5000$

(d) $t = 50000$

Figure 7.5: Evolution of a grain structure containing spherical second-phase particles ($f_V = 8\%$, $r_a = 1$), obtained from a phase field simulation on a $256 \times 256 \times 256$ grid. Images are shown at (a) $t = 200$, (b) $t = 2000$, (c) $t = 5000$, and (d) $t = 50000$. No further grain boundary movement is observed after $t = 50000$.



(a) $f_{\mathrm{V}} = 5\%, r_a = 2$

(b) $f_{\mathrm{V}} = 5\%, r_a = 3$

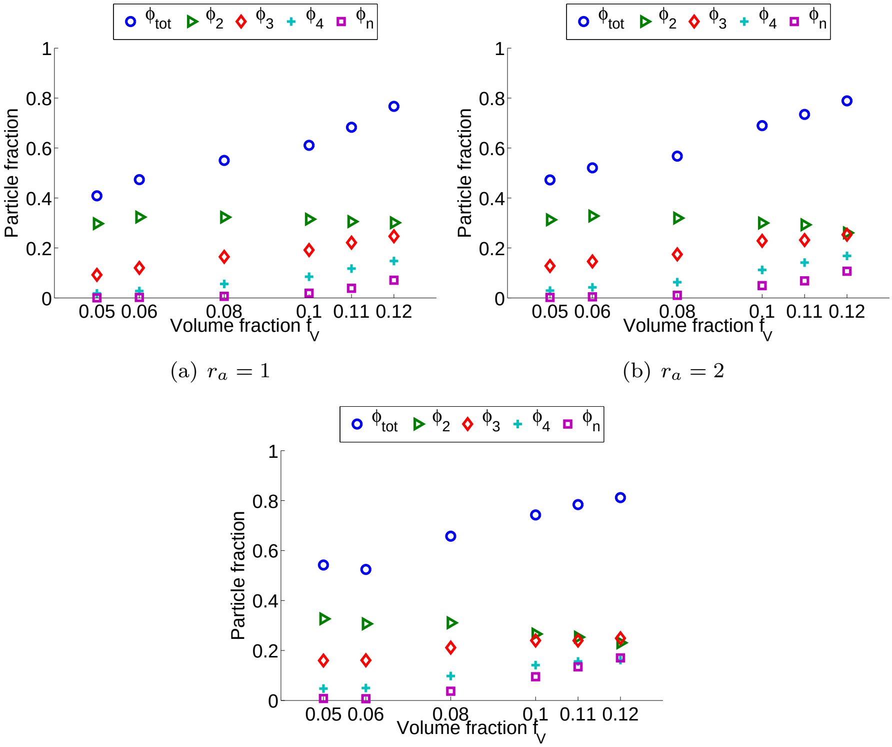
(c) $f_{\mathrm{V}} = 12\%, r_a = 2$

(d) $f_{\mathrm{V}} = 12\%, r_a = 3$

Figure 7.6: Cross-sections of three-dimensional pinned structures obtained from simulations for $f_{\mathrm{V}} = 5\%$ and 12%, and $r_a = 2$ and 3. No further evolution was observed after about (a) 70000, (b) 94000, (c) 34000 and (d) 34000 time steps.

Figure 7.7 shows the fraction ϕ_{tot} as a function of the volume fraction f_{V} for the three tested aspect ratios r_a separately, subdivided according to the different locations of the particles. In agreement with previous studies for spherical particles, the graphs show that the total fraction of particles present at boundaries increases with f_{V}. The present study furthermore shows that this increase is mainly due to an increase of the number of particles located at triple and quadruple junctions, while the fraction ϕ_2 of particles located at grain interfaces is more or less independent of the volume fraction of the spherical particles and slightly decreases for higher aspect ratios. The fraction ϕ_n of particles located at junctions of more than four grains slightly increases with f_{V} for aspect ratio $r_a = 1$ but remains rather small, while for aspect ratio $r_a = 3$, ϕ_n is considerably larger at higher volume fractions. The tendency of the particles to lie at multiple junctions at higher volume fractions is thus stronger for higher aspect ratio.

In Fig. 7.8, the fractions ϕ_i are shown as a function of r_a, for different volume fractions separately. This representation clearly shows that, except for statistical variations, the pinning behaviour changes with aspect ratio: for higher aspect ratio, particles lie on multiple junctions rather than at grain interfaces. This tendency becomes much more pronounced for larger volume fractions, as the amount of particles at multiple junctions relative to ϕ_{tot} increases and the grain size compared to the particle size decreases with volume fraction. Particles with $r_a = 3$ thus have a significant influence on the microstructure topology, especially at high volume fractions. We expect that the effect of particle shape increases further with the aspect ratio of the particles for r_a greater than 3.

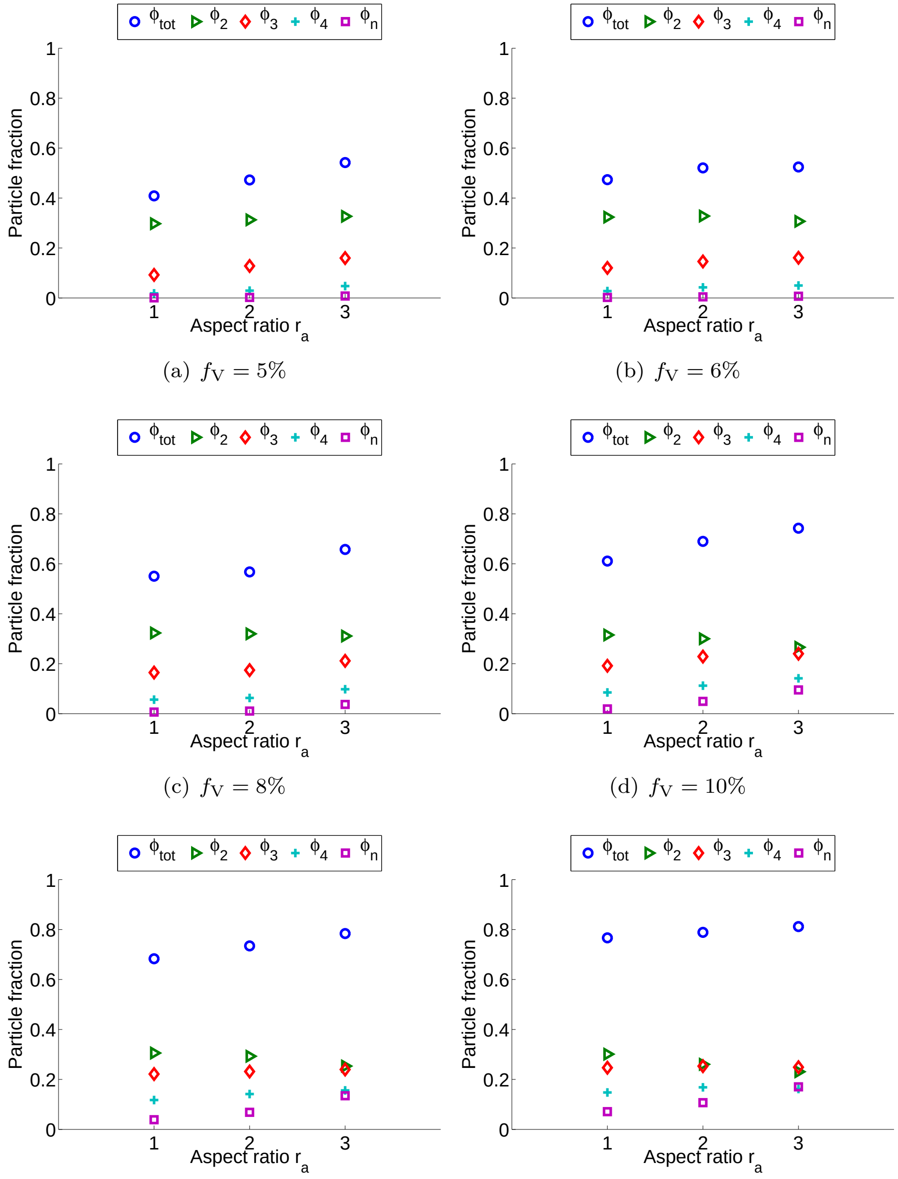


(a) $r_a = 1$

(b) $r_a = 2$

(c) $r_a = 3$

Figure 7.7: Fraction ϕ_{tot} of particles located at grain boundaries, divided into four types: located at grain interfaces (ϕ_2), at junctions where three grains meet (ϕ_3), where four grains meet (ϕ_4), and where more than four grains meet (ϕ_n), for different aspect ratios r_a.



(a) $f_V = 5\%$

(b) $f_V = 6\%$

(c) $f_V = 8\%$

(d) $f_V = 10\%$

(e) $f_V = 11\%$

(f) $f_V = 12\%$

Figure 7.8: Fraction ϕ_{tot} of particles located at grain boundaries, divided into four types: located at grain interfaces (ϕ_2), at junctions where three grains meet (ϕ_3), where four grains meet (ϕ_4), and where more than four grains meet (ϕ_n), for different volume fractions f_V.

7.4 Generalised Zener relation

7.4.1 Analysis of the simulation results

The presence of second-phase particles inhibits grain growth in such a way that grain growth stops when a limiting mean grain radius $\langle R\rangle_{\mathrm{lim}}$ is obtained. Most theories predict a relation between $\langle R\rangle_{\mathrm{lim}}$ and the size and volume fraction of the particles of the form

$$\frac{\langle R\rangle_{\mathrm{lim}}}{r} = K\frac{1}{f_{\mathrm{V}}^{b}}, \tag{7.3}$$

which is a generalisation of the Zener relation [81, 116]. In the classical Zener-Smith analysis, it is assumed that particles are spherical, incoherent and that particles and boundaries intersect on a random basis. As long as the number of particles at grain boundaries is much larger than the number of particles at multiple junctions, this approximation is likely to be valid [18]. According to [51, 81, 18] most particles may be situated in grain corners for high volume fractions. In this case, the Zener-Smith assumptions are not applicable anymore. Nevertheless, it is seen that a Zener type relation is still appropriate at higher volume fractions, although sometimes a weaker volume fraction dependence ($b \approx 0.33$ instead of $b \approx 1$ in formula (7.3)) of the limiting grain radius is found [51, 81]. According to [51], this change in volume fraction dependence is at $f_{\mathrm{V}} = 10\%$; according to [81], it is at $f_{\mathrm{V}} = 5\%$.

In [75], the pinning pressure is derived for a distribution of mono-oriented spheroid particles, following the classical Zener-Smith approach. Because of the assumptions made in this approach, the large differences in pinning forces for different shapes and orientations of single particles are not reflected in the computed effect of the corresponding particle distributions. It is not yet understood whether their approximations or the Zener-Smith approximations in general are applicable for spheroid particles with aspect ratio different from 1. The calculation of the total pinning effect of multiple spheroid particles thus proves to be a complex problem. Analysis of our simulation results can shed new light on the relation between $\langle R\rangle_{\mathrm{lim}}$ on the one hand, and the volume fraction and the aspect ratio of the particles on the other hand.

Zener relation (7.3) is formulated assuming spherical particles that are quantified by means of the particle radius r. To extend the Zener relation to spheroid particles, we propose three different measures m to replace the parameter r, namely the long axis radius $m = l$, the short axis radius $m = s$ and the geometric mean $m = (ls^2)^{\frac{1}{3}}$ of the three axis radii, yielding a relation of the form

$$\frac{\langle R\rangle_{\mathrm{lim}}}{m} = K\frac{1}{f_{\mathrm{V}}^{b}}. \tag{7.4}$$

The third measure is often used in experimental studies to characterise the size of the particles.

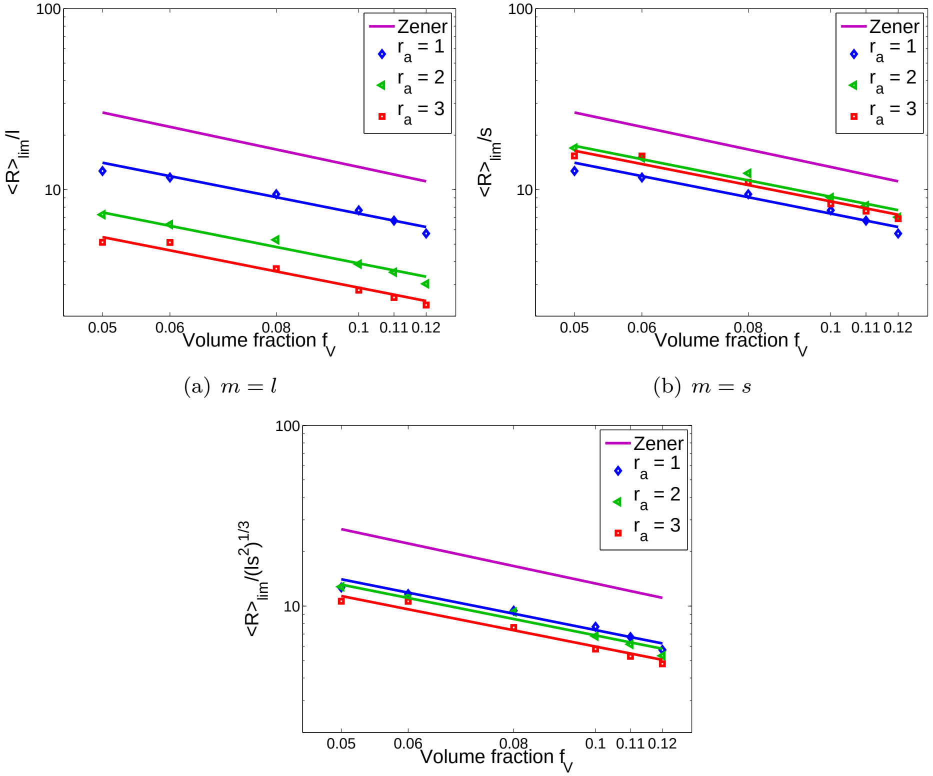
For each of the three proposed measures m, formula (7.4) is fitted to the simulation results. The resulting graphs are shown in Fig. 7.9. Relation (7.4) only has predictive value if the graphs for the three aspect ratios coincide. The three graphs in Fig. 7.9(a) are distinctly separated, with the graph of the particles with $r_a = 1$ on top and the graph of the particles with $r_a = 3$ at the bottom. The measure $m = l$ thus seems to be less appropriate. In Fig. 7.9(b), the graphs lie surprisingly close to each other, even though the measure $m = s$ does not include information on the aspect ratio. In Fig. 7.9(c), the graphs for $r_a = 1$ and $r_a = 2$ almost coincide, while the graph for $r_a = 3$ lies a little lower. As mentioned in the previous subsection, distributions of particles with higher aspect ratio therefore seem to have a stronger pinning effect for the same volume fraction and particle volume. Note that the simulation results of the parameter combination $f_{\mathrm{V}} = 5\%$ and $r_a = 1$ were excluded from the fitting calculations, since only one simulation run with this parameter combination contained a sufficient number of grains at grain growth stagnation.

Table 7.2 shows the estimated values for the parameters K and b obtained by fitting relation (7.4) to the simulation data for the three different aspect ratios r_a and the three different measures m. There is a very good correspondence between the data for $r_a = 1$ and the simulation results of [83], as is also illustrated in Fig. 7.11.

Table 7.2: Estimated values for the parameter set (K, b) in Zener relation (7.4) for three different aspect ratios r_a and three different measures m of the second-phase particles.

m	l	s	$(ls^2)^{\frac{1}{3}}$
$r_a = 1$	$(0.8593, 0.9333)$	$(0.8593, 0.9333)$	$(0.8593, 0.9333)$
$r_a = 2$	$(0.4557, 0.9339)$	$(1.0633, 0.9339)$	$(0.8017, 0.9339)$
$r_a = 3$	$(0.3377, 0.9295)$	$(1.0131, 0.9295)$	$(0.7024, 0.9295)$

As described in Section 7.3.5, the fraction of particles located at multiple junctions increases with increasing volume fraction and aspect ratio (see Figs. 7.7 and 7.8). Still, the values obtained for the parameter b agree very well with those obtained in [96] and [55], where, respectively, $b = 0.92$ and $b = 0.93$ were theoretically predicted, assuming that the main contribution to the pinning effect comes from particles interacting with grain boundaries. Irrespective of the exact explanation for the deviation from 1, the value for b seems to be independent of the aspect ratio of the particles. Refitting of the data for b fixed, namely $b = 0.93$, gives values for K as listed in Table 7.3. The values of K obtained for $m = l$ differ more strongly than those obtained for $m = s$ and $m = (ls^2)^{\frac{1}{3}}$. However, for all three measures



(a) $m = l$

(b) $m = s$

(c) $m = (ls^2)^{\frac{1}{3}}$

Figure 7.9: Fits of relation (7.4) to the limiting mean grain radius divided by (a) $m = l$, (b) $m = s$ and (c) $m = (ls^2)^{\frac{1}{3}}$, for all tested volume fractions f_V and for the three aspect ratios r_a separately. The Zener relation (7.4) only has predictive value if the graphs coincide for the different aspect ratios. The measure $m = l$ is not appropriate, since the graphs are clearly separated. For $m = s$ and $m = (ls^2)^{\frac{1}{3}}$, the three graphs are close to each other.

m, it seems to be more appropriate to have a prefactor K that is a function of the particle aspect ratio. Such a relation is derived in the next subsection.

7.4.2 Analytical derivation

The simulations show that most particles interacting with a grain boundary are oriented with their major axis parallel with the grain boundary. Furthermore, the finding that $b = 0.93$ indicates that the assumption of random intersections

Table 7.3: Estimated values for the parameter K in Zener relation (7.4) for a fixed parameter $b = 0.93$, for three different aspect ratios r_a and three different measures m of the second-phase particles.

m	l	s	$(ls^2)^{\frac{1}{3}}$
$r_a = 1$	0.8664	0.8664	0.8664
$r_a = 2$	0.4606	1.0746	0.8102
$r_a = 3$	0.3372	1.0116	0.7015

between grain boundaries and particles is reasonable. Therefore, as a first approximation, we repeat the Zener-Smith analysis [96, 81], but now adapted for spheroid particles with their major axis parallel with the grain boundary.

The pinning force of a spheroid particle interacting with a grain boundary and oriented with its major axis parallel with the boundary axis is described by equation (2.5) [114]. The maximal pinning force of a spherical particle with equal volume is given by

$$F_Z^S = \pi r^* \sigma_{\mathrm{gb}} = \pi (ls^2)^{\frac{1}{3}} \sigma_{\mathrm{gb}}, \tag{7.5}$$

where the radius r^* is computed as the geometrical mean of the radii of the spheroid particle. The maximal pinning force of the spheroid particle is therefore

$$F_Z^{\max} = (ls^2)^{\frac{1}{3}} \sigma_{\mathrm{gb}} \frac{1 + 2.14\ r_a}{r_a^{1/3}} = s\sigma_{\mathrm{gb}}(1 + 2.14\ r_a). \tag{7.6}$$

Following the original derivation of Zener-Smith [116], we assume that all grain boundaries are flat and that all particles are uniformly distributed in space. Furthermore, all particles interacting with boundaries are assumed to be oriented with their major axis parallel with the boundary and exert their maximal force $F_Z^{\max}$. The number of particles interacting with a grain boundary per unit of boundary area is accordingly

$$n_Z = \frac{f_{\mathrm{V}}}{\frac{4}{3}\pi l s^2} \times 2s = \frac{6 f_{\mathrm{V}}}{4\pi l s}, \tag{7.7}$$

assuming that all particles located within a distance of s on either side of the boundary interact with the boundary. The total pinning force exerted per unit of grain boundary area by the distribution of particles thus equals

$$F_Z^{\mathrm{tot}} = F_Z^{\max} \times n_Z = (1 + 2.14\ r_a) \frac{6 f_{\mathrm{V}} \sigma_{\mathrm{gb}}}{4\pi l}. \tag{7.8}$$

The driving force F_D for grain boundary movement is given by

$$F_D = \frac{2\alpha\sigma_{\mathrm{gb}}}{R}, \tag{7.9}$$

with α a geometrical constant [58]. Grain growth will stop when the driving force for grain boundary movement is equal to the total pinning force F_Z^{tot} or when

$$\frac{2\alpha\sigma_{\mathrm{gb}}}{\langle R\rangle_{\mathrm{lim}}} = (1+2.14\ r_a)\frac{6f_{\mathrm{V}}\sigma_{\mathrm{gb}}}{4\pi l}. \tag{7.10}$$

This equation yields a Zener type relation of the form

$$\frac{\langle R\rangle_{\mathrm{lim}}}{l} = K\frac{1}{1+2.14\ r_a}\frac{1}{f_{\mathrm{V}}}, \tag{7.11}$$

with K a constant coefficient. Generalising this relation gives

$$\frac{\langle R\rangle_{\mathrm{lim}}}{l} = K\frac{1}{1+ar_a}\frac{1}{f_{\mathrm{V}}^b}, \tag{7.12}$$

with K, a and b parameters that can be obtained by fitting the relation to simulation data. If the short axis s is used as a measure for particle size, the generalised Zener relation for spheroid particles becomes

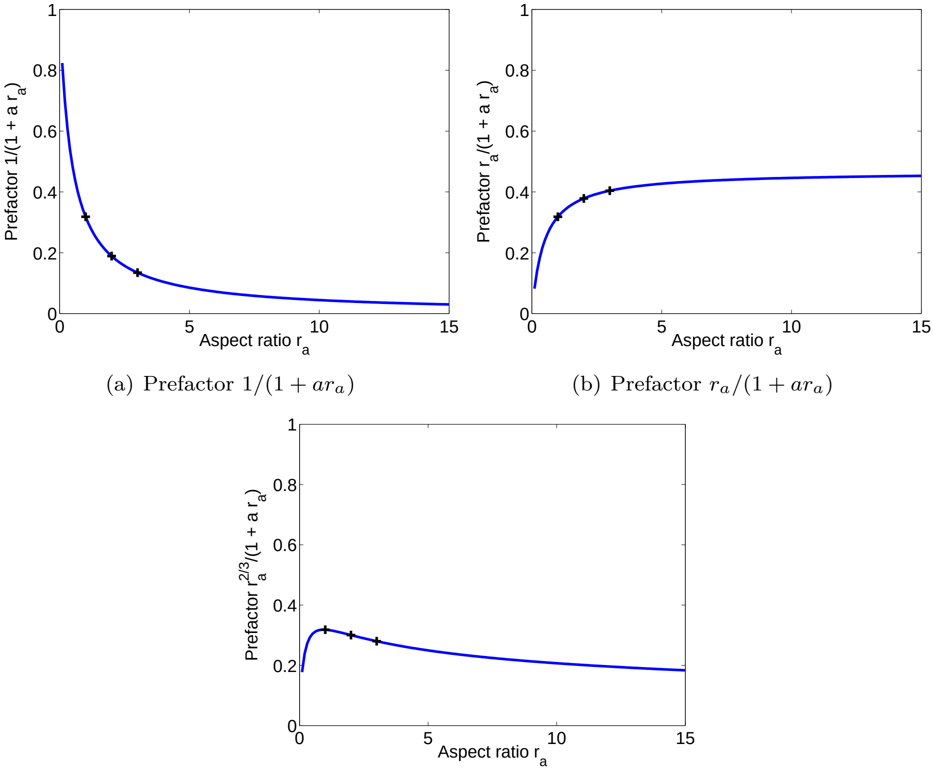
$$\frac{\langle R\rangle_{\mathrm{lim}}}{s} = K\frac{r_a}{1+ar_a}\frac{1}{f_{\mathrm{V}}^b}, \tag{7.13}$$

since $r_a = l/s$. Similarly, if the geometric mean $(ls^2)^{\frac{1}{3}}$ of the three axis radii is used as a measure for the particle size, the generalised Zener relation becomes

$$\frac{\langle R\rangle_{\mathrm{lim}}}{(ls^2)^{1/3}} = K\frac{(r_a)^{2/3}}{1+ar_a}\frac{1}{f_{\mathrm{V}}^b}. \tag{7.14}$$

Figure 7.10 shows a plots of prefactor $1/(1+ar_a)$ in relation (7.12), prefactor $r_a/(1+ar_a)$ in relation (7.13) and of prefactor $(r_a)^{2/3}/(1+ar_a)$ in relation (7.14) respectively, for $a = 2.14$. Since the first and the third prefactor tend to zero for high aspect ratios, whereas the prefactor of relation (7.13) tends to a constant value $1/a$, it is most appropriate to use s as a measure for the particle size in the generalised Zener relation. Figure 7.10 also shows that variations in the prefactors decrease for increasing aspect ratio, but are relatively large for $r_a \leq 2$. This explains why the graphs on Fig. 7.9 do not coincide. From relation (7.13), one may however expect that all curves for $\langle R\rangle_{\mathrm{lim}}/s$ in Fig. 7.9(b) will coincide for higher aspect ratios.

Fitting relation (7.13) to all the simulation data at once, including the data obtained from the simulation run for parameter combination $f_{\mathrm{V}} = 5\%$ and $r_a = 1$, yields the parameter values $(K, b, a) = (3.2774 \pm 2.1646, 0.9040 \pm 0.1392, 2.5492 \pm 1.6789)$. The 95% confidence interval is much smaller for the parameter b than for the other two parameters. Also, the value found for the parameter a is of the same order as the theoretical value 2.14. We stress however that more data points, considering higher aspect ratios and different orientation distributions of the particles, are required for a more accurate fit.



(a) Prefactor $1/(1 + ar_a)$

(b) Prefactor $r_a/(1 + ar_a)$

(c) Prefactor $r_a^{2/3}/(1 + ar_a)$

Figure 7.10: Plots of the prefactors of (a) relation (7.12), (b) relation (7.13) and (c) relation (7.14) as a function of the aspect ratio r_a, for $a = 2.14$. All plots show that variations with r_a decrease with increasing aspect ratio and are relatively small for $r_a > 2$. The prefactor values for $r_a = 1$, 2, and 3 are indicated.

7.5 Comparison with other studies

Figure 7.11 shows an overview of data obtained in different studies. The results of the simulations with spheroid particles are represented by the ratio $\langle R \rangle_{\mathrm{lim}}/m$, with $m = (ls^2)^{\frac{1}{3}}$, for aspect ratios $r_a = 1$ and $r_a = 3$. They are compared with the original Zener relation [116]

$$\frac{\langle R \rangle_{\mathrm{lim}}}{r} = \frac{4}{3}\frac{1}{f_{\mathrm{V}}}, \tag{7.15}$$

with the relation derived by Hillert for low ($f_{\mathrm{V}} < 10\%$) and high ($f_{\mathrm{V}} > 10\%$) volume fractions [55], the relation of Manohar that is based on a large compilation

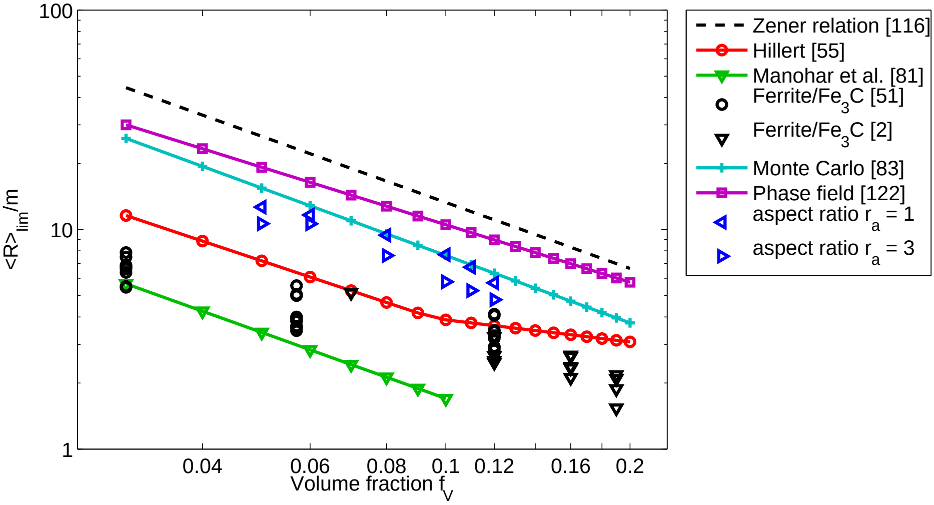


Figure 7.11: Comparison of the simulation results for spheroid particles with aspect ratios $r_a = 1$ and $r_a = 3$, represented by $\langle R \rangle_{\text{lim}}/m$ with $m = (ls^2)^{\frac{1}{3}}$, with the original Zener relation [116], the relation derived by Hillert for low ($f_{\text{V}} < 10\%$) and high ($f_{\text{V}} > 10\%$) volume fractions [55], the relation of Manohar based on a large compilation of experimental data [81], experimental results for cementite particles in a ferrite matrix [51, 2], and results obtained from simulations with a Monte Carlo Potts model [83] and with the continuum field model [122].

of experimental data [81], experimental results for cementite particles in a ferrite matrix [51, 2], and results obtained from simulations with a Monte Carlo Potts model [83] and with the continuum field model [122].

The cementite particles of [51] are characterised as ‘spherical’, but the microscopic pictures in the same article show that they have a rather irregular shape. These experimental data show that there is generally a large scatter on experimentally measured limiting grain sizes, even when they are obtained for the same material. Nevertheless, all experimental data points are considerably below the curves obtained for spherical particles.

Our data points for the distribution of spherical particles ($r_a = 1$) almost coincide with the relation obtained from Monte Carlo Potts simulation and are located slightly below the relation obtained from previous phase field simulations. The simulation data points for $r_a = 3$ lie lower and thus closer to the experimental data. The spheroid shape of the particles with aspect ratio $r_a = 3$ thus gives a better approximation for the pinning effect of real material particles. If particles have a slightly irregular shape, grain boundaries have probably the tendency to align themselves with the largest cross-section of the particles.

The simulation results show that the dispersions of spheroid particles have a stronger pinning effect than the dispersions of spherical particles. The effect is however too small to fully explain the large difference between the final grain sizes obtained in simulations and those obtained experimentally.

7.6 Conclusion

A common technique to control the grain size of a material is by the addition of impurities, which leads to the formation of second-phase particles. These particles inhibit grain boundary movement during grain growth. Despite long-standing research, there is still a large discrepancy between the final grain sizes predicted by theory and simulation and those measured for real materials. To gain more insight in this pinning phenomenon, the effect of particles with a spheroid shape on grain growth is studied in this chapter, using three-dimensional phase field simulations.

In order to simulate grain growth in the presence of second-phase particles with constant properties, the continuum field model (2.39) is employed. The studied particles are spheroid and characterised by their aspect ratio r_a, which is the ratio between the long axis radius and the short axis radius of a particle. As phase field simulations for grain growth are computationally very intensive, a parallel implementation of the bounding box algorithm, which was presented in Section 5.5, is used to execute the simulations in an efficient and fast way. Simulations are performed for different volume fractions, ranging from $f_{\mathrm{V}} = 5\%$ to $f_{\mathrm{V}} = 12\%$, and three different aspect ratios: $r_a = 1$, 2 and 3. For every parameter combination, the results are averaged over three simulation runs.

The pinning effect of a particle distribution is found to be stronger for increasing volume fraction, and for increasing aspect ratio. Similarly, the total fraction of particles present at boundaries increases with volume fraction and with aspect ratio, which is mostly due to an increase of the number of particles located at triple, quadruple or higher-order junctions. The grain boundaries have the tendency to align themselves with the longest axis of the particles with $r_a = 3$. For aspect ratio $r_a = 2$, the extra amount of grain boundary (or curvature) that has to be created in the neighbourhood of a particle when the boundary reorients with the long axis of the particle, seems to be too large compared to the amount of grain boundary removed by the particle after reorientation. Particles with $r_a = 3$ thus have a significant influence on the microstructural topology, especially at higher volume fractions. Nevertheless, the effect is still too small to fully explain the large difference between the final grain sizes obtained in simulations and those obtained experimentally.

In spite of the increasing number of particles located at boundary junctions, for all simulated aspect ratios, a volume fraction dependence of the limiting mean grain size of $1/f_{\mathrm{V}}^{0.93}$ is obtained, which follows the theoretical predictions of [96, 55] where a uniform distribution of spherical particles is assumed. Moreover, based on the simulation results, a generalised Zener type relation of the form

$$\frac{\langle R\rangle_{\mathrm{lim}}}{s} = K\frac{r_a}{1+ar_a}\frac{1}{f_{\mathrm{V}}^{b}} \tag{7.16}$$

is proposed.

The simulation results presented and discussed in this chapter correspond well to other simulation results, but deviate from experimental data. To perform simulations of realistic materials, the continuum field model still has to be extended. We believe that the bounding box algorithm will enable such simulations and provide better insight in microstructural evolution.

Chapter 8

Effect of anisotropic grain boundary properties on grain growth: initial results

8.1 Introduction

As mentioned in Chapter 2, a predictive model for grain growth in polycrystalline materials should include orientation dependent microstructural properties and interactions in order to be able to study the interplay of texture and grain growth. This chapter presents initial results on the effect of anisotropic boundary energy formulation on a polycrystalline microstructure during grain growth, obtained from phase field simulations with continuum field model (2.44).

In Section 8.2, two different discretisation schemes of the anisotropic continuum field model (2.44) are presented: a fully explicit scheme and a fully implicit scheme. For both discretisation schemes, a solver is implemented which integrates with the sparse data structure of the extended bounding box algorithm developed in Chapter 5. Furthermore, the simulation parameters are discussed and determined in accordance with [89, 90] for two different grain boundary energy formulations. The simulation results are analysed and compared in Section 8.3. Section 8.4 ends this chapter with some conclusions.

8.2 Simulation parameters

8.2.1 Phase field model and discretisation

Continuum field model (2.44) was constructed in Section 2.5.3 in order to simulate grain growth in a microstructure with anisotropic boundary properties. The model is formulated as

$$\frac{\partial \eta_i}{\partial t} = L \left(\kappa(\Theta) \nabla^2 \eta_i - m \left(\eta_i^3 - \eta_i + 2\eta_i \sum_{j \neq i}^{p} \gamma_{ij} \eta_j^2 \right) \right), \qquad i = 1, \ldots, p, \quad (8.1)$$

where Θ corresponds to the ensemble of parameters that defines the grain boundary misorientation, which is the difference in crystallographic orientation of two neighbouring grains.

The purpose of the simulations in this chapter is to study the evolution of a polycrystalline microstructure with a fibre texture during grain growth. In such a microstructure, the crystallographic orientations of the grains are nearly identical in one direction, a chosen axis, and random in the plane perpendicular to this axis. The misorientation, which is the difference in orientation between two neighbouring grains, can therefore be described by a single misorientation angle θ. This is illustrated in Fig. 2.5, which shows a schematic representation of grain boundary misorientation in a fibre-textured microstructure.

Application of the Forward Euler method to model (2.44), in combination with second-order central finite differences to the Laplacian, now yields the following discretisation scheme:

$$\frac{\eta_{i,\mathbf{r}}^{n+1} - \eta_{i,\mathbf{r}}^{n}}{\Delta t} = L\kappa(\theta_{\mathbf{r}}^{n}) \nabla^2 \eta_{i,\mathbf{r}}^{n} - Lm \left(\left(\eta_{i,\mathbf{r}}^{n}\right)^3 - \eta_{i,\mathbf{r}}^{n} + 2\eta_{i,\mathbf{r}}^{n} \sum_{j \neq i}^{p} \gamma_{ij} \left(\eta_{j,\mathbf{r}}^{n}\right)^2 \right),$$

$$\mathbf{r} = (x, y, z) \in \Omega; \qquad i = 1, \ldots, p, \quad (8.2)$$

with

$$\kappa(\theta_{\mathbf{r}}^{n}) = \frac{\sum_{i=1}^{p} \sum_{j>i}^{p} \kappa_{ij} \left(\eta_{i,\mathbf{r}}^{n}\right)^2 \left(\eta_{j,\mathbf{r}}^{n}\right)^2}{\sum_{i=1}^{p} \sum_{j>i}^{p} \left(\eta_{i,\mathbf{r}}^{n}\right)^2 \left(\eta_{j,\mathbf{r}}^{n}\right)^2}, \quad (8.3)$$

$$\nabla^2 \eta_{i,\mathbf{r}}^{n} = \sum_{v=x,y,z} \frac{\eta_i^n(v + \Delta v) - 2\eta_i^n(v) + \eta_i^n(v - \Delta v)}{(\Delta v)^2}, \quad (8.4)$$

and Ω the system domain. Equations (8.2) can directly be applied to compute the values of the next time step as a function of the values on the previous time step.

However, the stability condition of explicit methods is generally very strict for the time step size Δt.

We have implemented an explicit time stepper based on scheme (8.2) that integrates into the extended bounding box algorithm, which was developed in Section 5.6 to simulate grain growth in a system with anisotropic boundary energy depending on the misorientation between neighbouring grains.

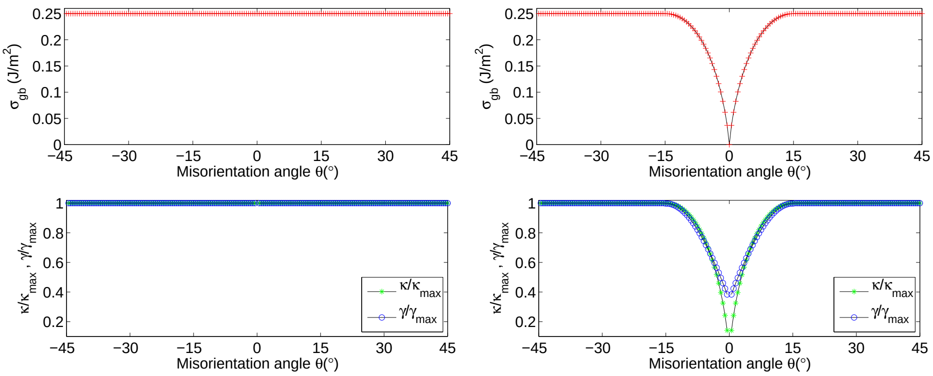
Remark

We have also applied the Backward Euler method to continuum field model (8.1). In combination with second-order central finite differences for the Laplacian, this method yields the following discretisation scheme:

$$\begin{aligned}\frac{\eta_{i,\mathbf{r}}^{n+1} - \eta_{i,\mathbf{r}}^{n}}{\Delta t} &= L\kappa(\theta_{\mathbf{r}}^{n+1})\nabla^2 \eta_{i,\mathbf{r}}^{n+1} \\ &\quad - Lm\left(\left(\eta_{i,\mathbf{r}}^{n+1}\right)^3 - \eta_{i,\mathbf{r}}^{n+1} + 2\eta_{i,\mathbf{r}}^{n+1}\sum_{j\neq i}^{p}\gamma_{ij}\left(\eta_{j,\mathbf{r}}^{n+1}\right)^2\right), \\ &\qquad \mathbf{r} = (x,y,z) \in \Omega; \qquad i = 1,\ldots,p, \quad (8.5)\end{aligned}$$

with $\kappa(\theta_{\mathbf{r}}^{n+1})$ defined as in (8.3) and Ω the system domain. In contrast to the explicit Forward Euler scheme (8.2), the Backward Euler scheme (8.5) is a system of nonlinear equations. In order to solve this system, a nonlinear solver was implemented based on the Gauss-Seidel-Newton method, which is similar to the nonlinear smoothing procedure described in Section 6.2.3. The convergence criterion of the Gauss-Seidel method is met when the norm of the residual scaled by the norm of the right-hand side falls below a tolerance value provided by the user. The maximal amount of iterations is set to 500. The Newton step is performed only once per unknown, per Gauss-Seidel iteration.

During a bounding box simulation, a grain region is allowed to grow with M grid points in each grid direction during a time step. So far, the value $M = 1$ has proved to be sufficient. When a fully implicit scheme is employed, a larger time step size Δt is possible. During such a large time step, a grain region might grow more than one grid point in each grid direction. In this case, the margin size M should be chosen larger than the value 1. Since our experience with the implicit solver is limited, in the next sections, we will only make use of the explicit solver.



(a) Isotropic boundary energy

(b) Anisotropic boundary energy

Figure 8.1: Grain boundary energy σ_{gb} and model parameters κ and γ as a function of the misorientation angle θ for (a) the isotropic boundary energy function, and (b) the anisotropic boundary energy function (8.7). The parameters κ and γ are scaled by their maximal values $\kappa_{\max} = 0.25$ and $\gamma_{\max} = 1.5$.

8.2.2 Anisotropic boundary energy formulation

As in [91], a fourfold symmetry of the crystallography is assumed. The orientations within one quadrant are discretised with an interspacing of $\Delta\theta = 90°/p$ and the discrete orientations are assigned to p phase field variables $\eta_1, \eta_2, \ldots, \eta_p$. The misorientation angle θ associated with the boundary between two neighbouring grains with orientations i and j is calculated as

$$\theta = \begin{cases} \Delta\theta \cdot |j - i|, & |j - i| \leq \frac{p}{2} \\ -90° + \Delta\theta \cdot |j - i|, & |j - i| > \frac{p}{2} \end{cases} \tag{8.6}$$

and ranges from $-45°$ to $45°$.

In this chapter, the effect of two different boundary energy formulations is studied. The first boundary energy function is isotropic and attributes a constant boundary energy $\sigma_{\text{gb}} = \sigma_m = 0.25\ J/m^2$ to all grain boundary types. Accordingly, the associated model parameters $\kappa(\theta)$ and $\gamma(\theta)$ are constant functions, which are plotted in Fig. 8.1(a).

The second boundary energy function is anisotropic and distinguishes between low-angle boundaries ($|\theta| < \theta_m$) and high-angle boundaries ($|\theta| \geq \theta_m$). The high-angle boundaries are attributed the same constant boundary energy σ_m. The concept of low-angle boundaries, or subgrain boundaries, was introduced in Section 2.2.3. There, it was explained that subgrain boundaries are defined by a rotation vector $\boldsymbol{\theta}$, a rotation angle θ, and the position of the boundary plane with respect to the rotation vector. In a microstructure with fibre texture, the rotation axis defined by $\boldsymbol{\theta}$ coincides with the fibre axis. Depending on the orientation of the boundary planes with regard to this axis, the low-angle boundaries can be divided into three different types: pure twist boundaries, partial twist boundaries, and tilt boundaries. As mentioned in Section 2.2.3, the boundary energy of these three boundary types can be described by the Read-Shockley dependence (2.10) if the values of the function parameters are modified correspondingly. In this work, we will use the same parameter values for all subgrain boundary types and thus assume the same boundary energy function for all low-angle boundaries. The resulting anisotropic boundary energy function is formulated as:

$$\sigma_{\mathrm{gb}}(\theta) = \begin{cases} \sigma_m \frac{|\theta|}{\theta_m}\left(1 - \ln\left(\frac{|\theta|}{\theta_m}\right)\right), & |\theta| < \theta_m \\ \sigma_m, & |\theta| \geq \theta_m \end{cases} \tag{8.7}$$

with $\sigma_m = 0.25\ J/m^2$ and $\theta_m = 15°$. Figure 8.1(b) plots $\sigma_{\mathrm{gb}}(\theta)$, as well as the corresponding model parameters $\kappa(\theta)$ and $\gamma(\theta)$, as a function of the misorientation angle θ. The parameter functions $\kappa(\theta)$ and $\gamma(\theta)$ are computed with the procedure developed in [89, 90], which allows to reproduce the grain boundary energy and mobility of a material for arbitrary misorientation and inclination dependence. This calculation of the model parameters furthermore permits to perform quantitative simulations with uniform stability and accuracy conditions.

For both boundary energy functions, a constant grain boundary mobility $\mu_{\mathrm{gb}} = 1 \times 10^{-6}$ m^{2}s/kg is assumed, while the grain boundary width is $\ell_{\mathrm{gb}} = 1.33 \times 10^{-6}$ m. Calculation of the other model parameters according to the procedure of [89, 90] gives $m = 1.125 \times 10^6$ J/m^3 and $L = 1$ ms/kg. Since both applied boundary energy functions are symmetric, the properties of the grain boundaries can be studied within a restricted misorientation range of $[\Delta\theta, 45°]$.

8.2.3 Simulation set-up

Simulations are performed on a 256 × 256 × 256 grid and start from the same microstructure, initialised with Algorithm 5.2 for $p = 500$ grain regions. One simulation run uses the isotropic boundary energy function, while the other simulation run employs the anisotropic boundary energy function (8.7). For both runs, the lattice spacing Δx is taken equal to 0.2×10^{-6} m and the time step size Δt is fixed to 0.015 s. Periodic boundary conditions are assumed.

In Chapter 5, the use of a threshold value of $\epsilon = 10^{-5}$ or 10^{-6} is advocated based on simulation results with the continuum field model (2.39). Numerical experiments with model (2.44) indicate the same sensitivity to the applied threshold value. In terms of computational requirements, a simulation run with $\epsilon = 10^{-5}$ takes approximately half of the execution time required by $\epsilon = 10^{-6}$ at the start. In the same way, the data files accompanying the former threshold value take 25% less storage space. The value $\epsilon = 10^{-5}$ is therefore a valid and practical choice for the simulations in this chapter. All simulations are run on four processors, with the parallel implementation of the extended bounding box algorithm, which was described in Section 5.6.

8.3 Effect of anisotropic boundary energy properties

8.3.1 Microstructure evolution

The images in Fig. 8.2 capture the evolution of a microstructure with anisotropic boundary energy function (8.7) at time points $t = 9$, 180, 360, and 720. For each time point, the microstructure is visualised by displaying the values computed for the function Ψ, which was defined in Chapter 5 as:

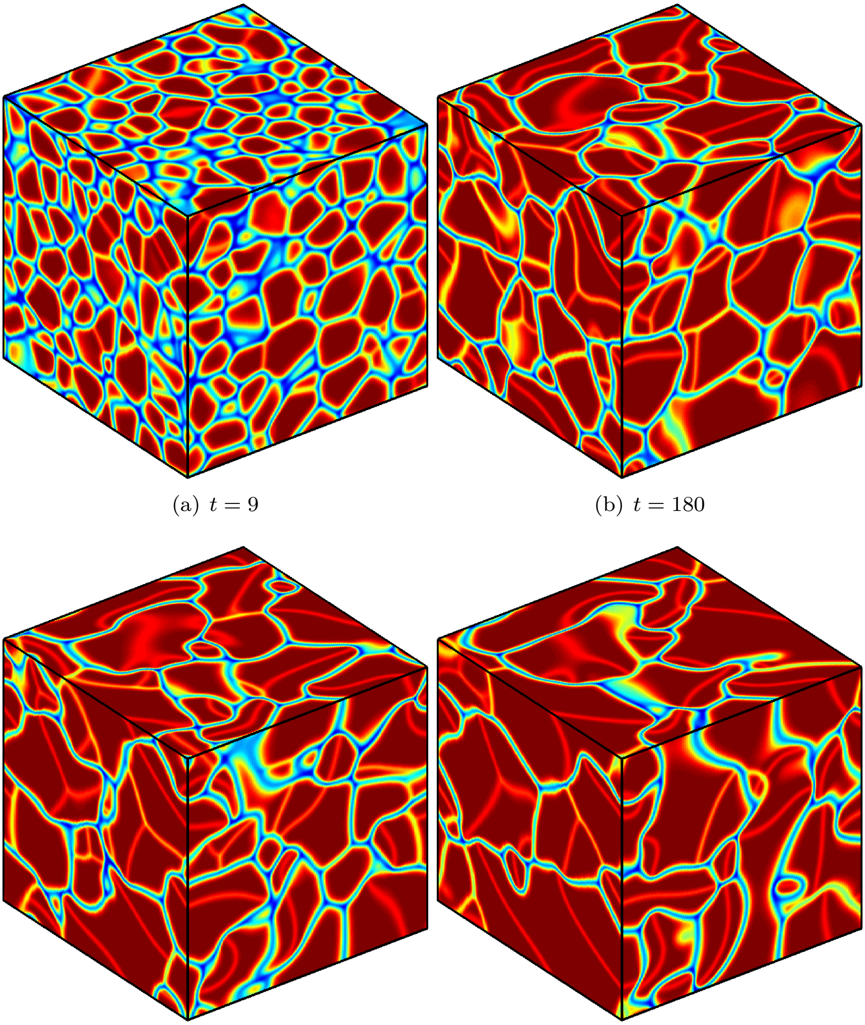
$$\Psi(\mathbf{r}, t) = \sum_{i=1}^{p} \eta_i^2(\mathbf{r}, t). \tag{8.8}$$

Within a grain, the function Ψ takes a constant value close to 1, while at grain boundaries, Ψ takes a lower value. The function Ψ distinguishes the low-angle boundaries (LABs) from the high-angle boundaries (HABs) by assuming a larger value at LABs than at HABs. Accordingly, the former boundary type is coloured light red in Fig. 8.2, while the latter type is coloured blue.

The network of HABs seems to evolve independently over the images of Fig. 8.2, whereas the LABs follow the movement of the HABs and enlarge accordingly. Also, it can be seen that the LABs meet the HABs almost perpendicularly, because of the large differences in boundary energy. The relative presence of the LABs furthermore seems to increase with time: the LABs seem to be preferred during grain growth.

8.3.2 Evolution of grain boundary characteristics

The misorientation distribution (MD) reflects the characteristics of the grain boundaries in a microstructure. In a fibre-textured microstructure, the MD describes the occurrence of each possible misorientation angle θ (8.6) as the misorientation between two grains. To calculate the area of a grain boundary, the following assumptions are made. The volume of a grain with crystallographic orientation i, or grain i, is defined as the ensemble of grid points where the phase field variable η_i has the highest absolute value of all phase field variables. The area of the grain boundary between two grains i and j is then computed by



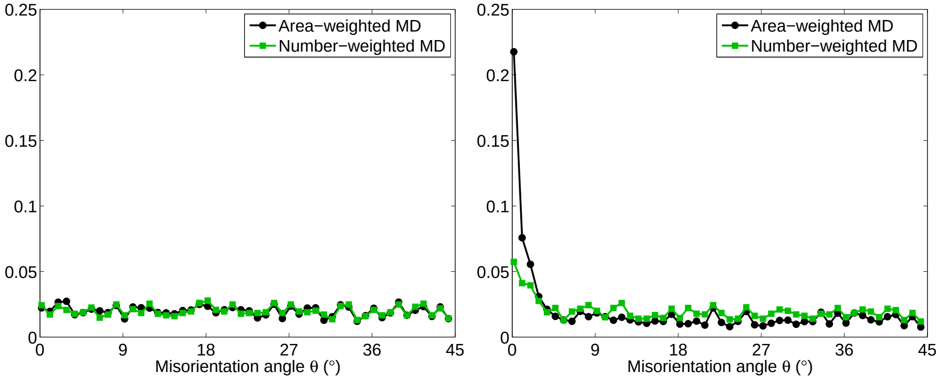
(a) $t = 9$

(b) $t = 180$

(c) $t = 360$

(d) $t = 720$

Figure 8.2: Simulation images of a microstructure with anisotropic boundary energy at (a) $t = 9$, (b) $t = 180$, (c) $t = 360$, and (d) $t = 720$, computed with threshold value $\epsilon = 10^{-5}$. The low-angle boundaries are coloured light red, while the high-angle boundaries are coloured blue.



(a) Isotropic boundary energy

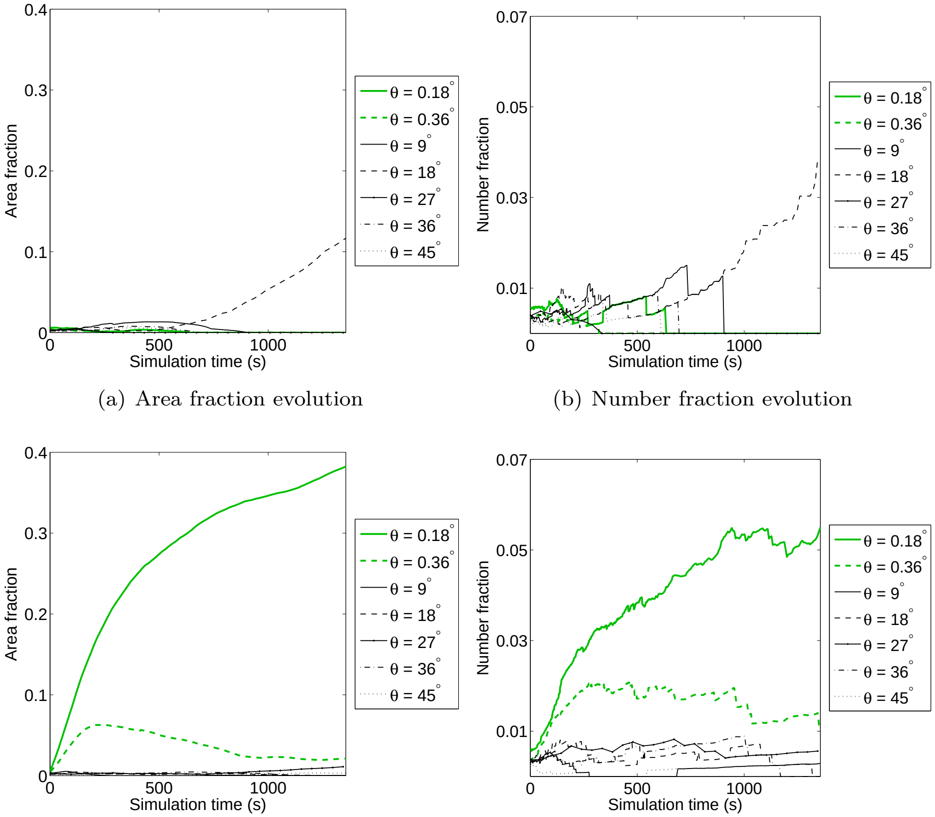
(b) Anisotropic boundary energy

Figure 8.3: Area-weighted and number-weighted MD (a) in a microstructure with isotropic boundary energy and (b) in a microstructure with anisotropic boundary energy (8.7), at $t = 108$.

counting all grid points of grain i that lie next to a grid point of grain j, plus all grid points of grain j that lie next to a grid point of grain i. In this view, grains and grain boundaries overlap. Since we are interested in relative quantities rather than absolute areas and volumes, this is a valid approach to construct the area-weighted MD. The number-weighted MD is computed by counting for each occurring misorientation the number of boundaries representing that particular misorientation.

Figure 8.3 plots the area-weighted and the number-weighted MD for the two simulated microstructures at one specific time point, namely $t = 108$. At this time point, there are 241 of the initial 500 grains left in the microstructure with isotropic boundary energy and 238 grains in the microstructure with anisotropic boundary energy. For both microstructures, the shape of the MDs is in keeping with the shape of the boundary energy functions in Fig. 8.1. The MDs of the microstructure with isotropic properties in Fig. 8.3(a) agree with the uniform boundary energy. Similarly, in the microstructure with anisotropic boundary properties, the area and number fractions of the HABS are approximately equally low, while the presence of the LABs increases for decreasing boundary energy. Note that the increase of the area fractions for decreasing misorientation angle θ is much more pronounced than the corresponding increase of the number fractions.

Figure 8.4 shows the evolution of the area and number fraction of boundaries with specific misorientations, among which the two lowest occurring misorientation angles, namely $\theta = \Delta\theta = 0.18°$ and $\theta = 2\Delta\theta = 0.36°$. The preference for the latter LABs in the microstructure with the anisotropic boundary energy function



(a) Area fraction evolution

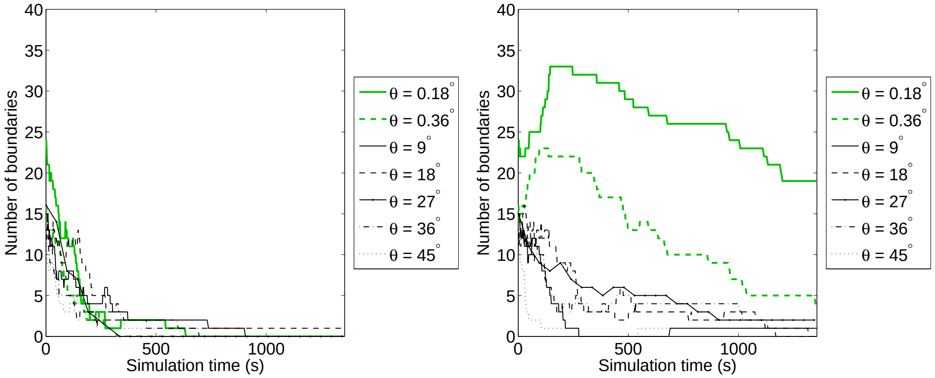
(b) Number fraction evolution

(c) Area fraction evolution

(d) Number fraction evolution

Figure 8.4: Evolution of the area and number fraction of boundaries with specific misorientations, defined by the misorientation angle θ, (a/b) in a microstructure with isotropic boundary energy and (c/d) in a microstructure with anisotropic boundary energy (8.7).

is clearly visible in Figs. 8.4(c) and 8.4(d): both the area and the number fraction of the two lowest-angle boundaries strongly increase during a simulation, while the presence of the HABs remains much lower. The graphs in Figs. 8.4(a) and 8.4(b) show that for the microstructure with the isotropic boundary energy, the area and number fraction remain approximately equally distributed over the range of the misorientation angle at the start of the simulation. At approximately $t = 800$, there are 92 of the initial 250 misorientations left in the isotropic microstructure. The relative presence of the boundaries with the remaining misorientations accordingly increases with time, as can be seen in Figs. 8.4(a) and 8.4(b). At the same time point, the anisotropic microstructure still contains



(a) Isotropic boundary energy

(b) Anisotropic boundary energy

Figure 8.5: Evolution of the number of boundaries with specific misorientations, defined by the misorientation angle θ, (a) in a microstructure with isotropic boundary energy and (b) in a microstructure with anisotropic boundary energy (8.7).

216 different misorientations.

The graphs in Fig. 8.5 illustrate the evolution of the absolute number of boundaries with misorientations corresponding to those studied in Fig. 8.4. In the microstructure with isotropic boundary energy, all boundary types decrease in number at approximately the same rate, as depicted in Fig. 8.5(a). This is not the case for the microstructure with anisotropic boundary energy, where the two lowest-angle boundaries remain present longer and even temporarily increase in number with respect to the initial state, which is illustrated in Fig. 8.5(b). The other boundary types all decrease at approximately the same rate.

Similar to Fig. 8.3, Figure 8.4 shows that the differences between the area fractions of the different boundary types within the same microstructure is larger than the corresponding differences between the number fractions. This is in keeping the observation made based on the microstructural images of Fig. 8.2, namely that the LABs seem to follow the movement of the HABs and enlarge.

8.3.3 Evolution of grain characteristics

Analogous to the calculation of the grain boundary area described in the previous section, the size of a grain with crystallographic orientation corresponding to phase field variable η_i is computed as the number of grid points where the value η_i has the highest absolute value of all phase field variables. The mean grain radius $\langle R \rangle$

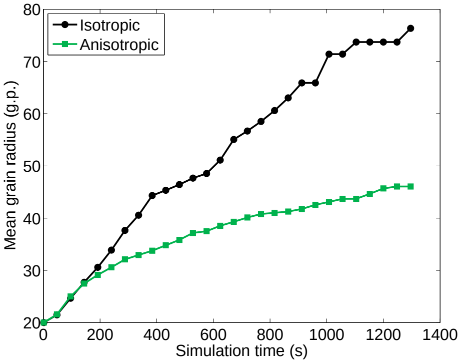


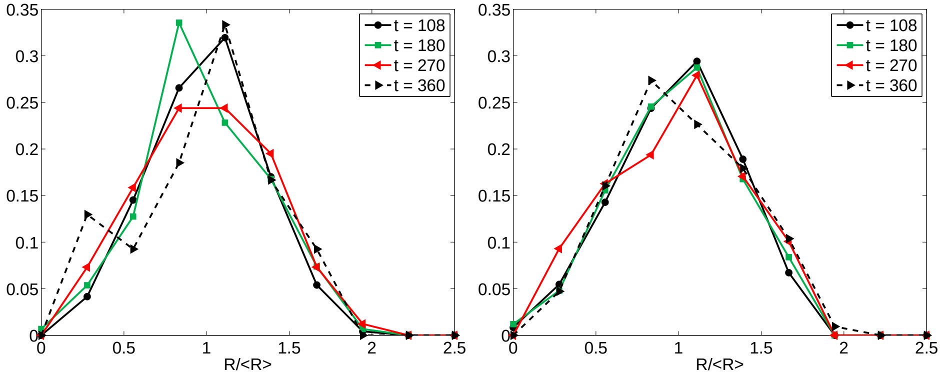
Figure 8.6: Time evolution of the mean grain radius $\langle R \rangle$ as computed for a microstructure with isotropic boundary energy and a microstructure with anisotropic boundary energy (8.7).

is computed as the radius of a sphere with a volume equivalent to that of the grain.

The time evolution of the mean grain radius $\langle R \rangle$ is depicted for both the microstructure with isotropic boundary energy and the microstructure with the anisotropic boundary energy in Fig. 8.6. The anisotropic boundary energy is found to slow down the growth kinetics in comparison to the isotropic boundary energy case. However, a larger initial amount of grains is required to come to reliable conclusions on long-term grain growth kinetics.

Figure 8.7 illustrates the evolution of the grain size distribution for both microstructures. For the isotropic microstructure, the shape of the grain size distribution is independent of time, which is the expected behaviour [58]. The dent in the grain size distribution at $t = 360$ is due to the small number of grains left in the microstructure. Figure 8.7(b) shows that the shape of the grain size distribution for the microstructure with anisotropic grain boundary energy is time invariant as well. The grain size distributions for the two simulated microstructures at time point $t = 108$ are compared in Fig. 8.8(a). There is no visible effect of the different grain boundary energy functions on the grain size distribution. In contrast, the distribution of the number of faces per grain, which is illustrated in Fig. 8.8(b), differs for the two studied microstructures. Overall, the number of faces per grain is higher when the anisotropic boundary energy function is applied.

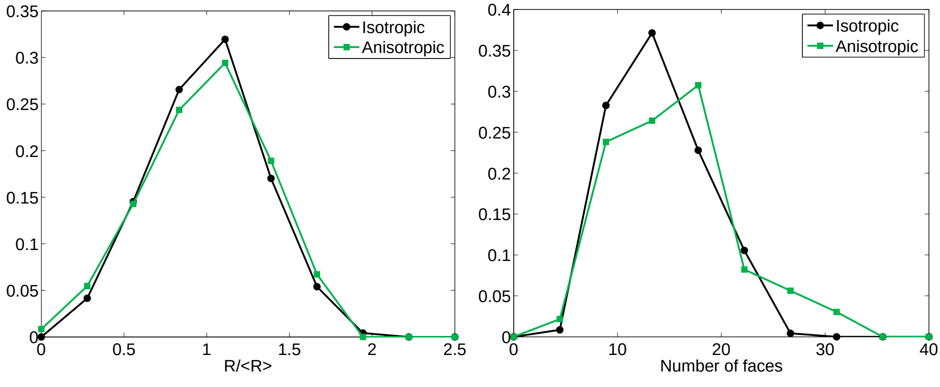
Recently, MacPherson and Srolovitz [80] suggested a relation between the growth rate of an individual grain and the number of grain faces. Under simplifying assumptions, this relation, which is the multi-dimensional extension of the von Neumann-Mullins equation [134] developed for two-dimensional structures, can be



(a) Isotropic boundary energy

(b) Anisotropic boundary energy

Figure 8.7: Evolution of the grain size distribution obtained from simulation results at $t = 108$ (a) of a microstructure with isotropic boundary energy and (b) of a microstructure with anisotropic boundary energy (8.7).



(a) Grain size distribution

(b) Grain face distribution

Figure 8.8: (a) Grain size distributions and (b) grain face distributions obtained from simulation results at $t = 108$ in a microstructure with isotropic boundary energy and in a microstructure with anisotropic boundary energy (8.7).

written as:

$$\frac{d\left(D(f)\right)^2}{dt} = c_1 \mu_{\text{gb}} \sigma_{\text{gb}} \left(6 - c_2 f^{\frac{1}{2}}\right), \tag{8.9}$$

where the quantity D is the linear dimension of a grain, which is computed as the diameter of a sphere with the equivalent volume of the grain, and f is the number of grain faces. Figure 8.9(a) plots the value of dD^2/dt, computed for the individual grains of the microstructure with isotropic boundary energy at time point $t = 108$, as a function of f. The simulation results are compared with a fit of the relation according to the theory of [80], as well as the theories of [95, 43, 53]. Relation (8.9) predicted by [80] shows the closest fit.

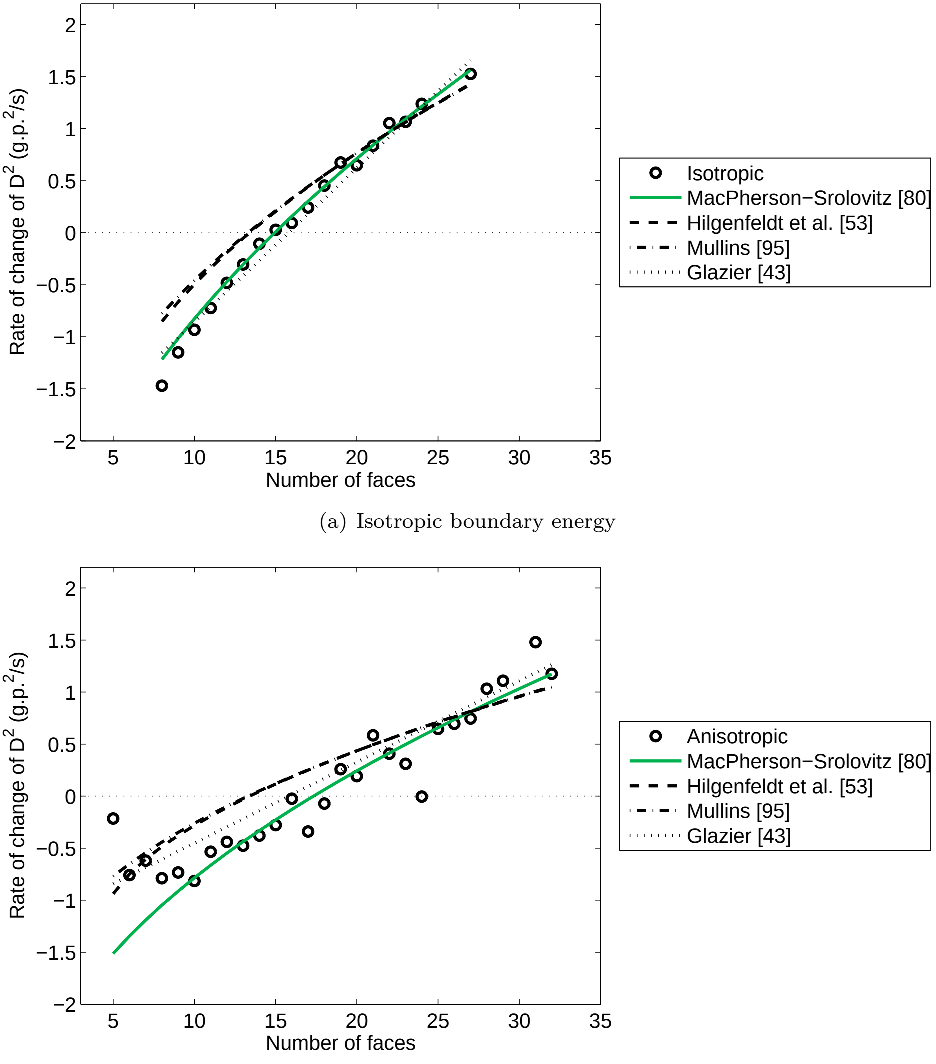
All theories mentioned above were developed for grain structures with isotropic boundary properties. Nevertheless, we have also tested the simulation results obtained from the microstructure with anisotropic boundary energy at $t = 108$ with the latter theories. Figure 8.9(b) illustrates how the growth rates for the different topological classes still approximately follow the relations predicted for grain structures with isotropic properties. However, there is much more scatter on the data than for the isotropic case in Fig. 8.9(a). Also, it can be seen that the range of the number of grain faces is larger in the microstructure with anisotropic boundary energy, which is in accordance with the grain face distribution shown in Fig. 8.8(b).

According to the theories of [95] and [53], the neutral number of faces, for which the grain size is stable, is $f_0 \approx 13.35$, while [43] predicts a stable grain size for $f_0 = 15.8$. For the microstructure with isotropic boundary energy, the fit with the relation of [80], which is the closest to the simulation data, indicates a stable topology class for $f_0 \approx 15$. In contrast, for the microstructure with anisotropic energy boundary, the simulation data in Fig. 8.9(b) displays a higher neutral number of faces, namely $f_0 \approx 17.3$.

8.4 Conclusion

During grain growth, grains with favourable properties grow at the expense of other grains, which results in an increase of the mean grain size. In particular, the evolution of the grain structure is influenced by the misorientation between the crystallographic orientations of neighbouring grains. In this chapter, preliminary results are presented on phase field simulation with continuum field model (2.44), which models grain growth in a microstructure with anisotropic boundary energy properties. To simulate with the latter nonlinear model, two numerical solvers have been implemented.

In order to gain some insight into the effect of crystallographic orientation dependent properties on microstructural evolution, three-dimensional grain growth simulations are performed in a microstructure with fibre texture, for two different boundary energy functions. The first boundary energy function is isotropic and attributes the same boundary energy to every boundary type. The second boundary energy function is anisotropic and attributes a constant boundary energy to all high-angle boundaries, while a Read-Shockley dependence is assumed for the low-angle boundaries. The model parameters are calculated with the procedure



(a) Isotropic boundary energy

(b) Anisotropic boundary energy

Figure 8.9: Plot of the value dD^2/dt as a function of the number of grain faces f, with D the linear dimension of a grain, computed at $t = 108$ (a) for a microstructure with isotropic boundary energy and (b) for a microstructure with anisotropic boundary energy (8.7). The simulation results are compared with fits of the theories of MacPherson and Srolovitz [80], Hilgenfeldt et al. [53], Mullins [95], and Glazier [43]. The intersections with the horizontal dotted line indicate for which number of faces the theories predict a stable grain size.

developed in [89, 90], which allows to reproduce any given boundary energy function and yields grain boundaries with homogeneous numerical properties. All simulations are performed with the parallel implementation of the extended bounding box algorithm, presented in Section 5.6.

The simulation results obtained for the microstructure with isotropic boundary energy properties are compared with those obtained for the microstructure with anisotropic boundary energy properties. The low-angle boundaries (LABs), which have a lower boundary energy, are clearly preferred during grain growth. The LABs remain much longer present in the microstructure, while the other boundary types disappear at approximately the same rate. The anisotropic boundary energy formulation is furthermore observed to change the individual growth rates of the grains as a function of the number of grain faces. The neutral number of faces, for which the grain size is stable, is found to be larger in the microstructure with anisotropic boundary properties than in the microstructure with isotropic boundary properties.

For realistic materials, a more general, three-dimensional description of the crystallography will provide more insight into the interplay of texture and grain growth. The formulation of such a crystallography in the context of phase field simulation is still subject to materials science research. Furthermore, a larger initial amount of grains is required to derive a reliable analysis of the effect of anisotropic boundary energy on grain growth from three-dimensional phase field simulations.

Chapter 9

Conclusions

9.1 Introduction

The microstructure of many materials consists of multiple grains with different crystallographic orientations. Under certain circumstances, such as increased temperature, the smaller grains will shrink and disappear under the influence of surface tension. This phenomenon, called grain growth, is thus characterised by an increase of the mean grain size. The study of the factors influencing grain growth is of great technological importance, because many material properties, such as strength and toughness, depend on the mean grain size and the grain size distribution.

Computer simulations are essential in the study of grain growth, since they enable parameter studies and provide three-dimensional insight into microstructural evolution, which can be difficult to obtain from experiments. One of the modelling techniques that has been successfully explored by many researchers for the simulation of microstructural evolution is phase field modelling.

The goal of this thesis was to design, analyse and implement efficient numerical solvers for grain growth simulation with two specific phase field models. With the newly developed software, we addressed certain questions with regard to the effect of second-phase particles and the presence of texture on grain growth. This concluding chapter starts with an overview of the results of this thesis in Section 9.2. In Section 9.3, we highlight the main contributions of this work. Section 9.4 ends the chapter with suggestions for future research.

9.2 Summary

In **Chapter 2**, the concepts of the grain growth phenomenon were introduced. After a discussion on the different existing types of grain growth models, two phase field models, namely the multi-phase field model (2.27) and the continuum field model (2.37) were presented, which are the main interest of this thesis. Both models represent a polycrystalline microstructure with a large set of phase field variables, where each variable corresponds to a single crystallographic orientation. In principle, the number of phase field variables included in a grain growth simulation with one of these models should equal or exceed the total number of grains, as in reality the number of possible orientations is infinite. Especially in three dimensions, a shortfall in the number of included phase field variables can cause incorrect growth kinetics. Realistic three-dimensional phase field simulations of grain growth thus demand significant amounts of computation power as well as data storage.

In **Chapter 3**, we have given an overview of the different acceleration techniques that have been applied to overcome the computational constraints of the multi-phase field model and the continuum field model. The strategies tested so far range from the application of different discretisation schemes to the use of sparse data structures and the development of efficient solvers. One of the discussed techniques is the application of IMEX time integration schemes to the continuum field model, which treat the diffusion part of the model with an implicit time integration method, while the reaction part is treated with an explicit time integration method. This approach is validated by the stability analysis in **Chapter 4**, where the amplification of perturbations consisting of Fourier modes is studied both theoretically and numerically.

In **Chapter 5**, a sparse bounding box algorithm is presented to perform efficient phase field simulations of grain growth in polycrystalline materials. The algorithm only solves the phase field equations locally, within bounding boxes that delimit active regions of the phase field variables. The bounding box algorithm shows significant improvements over existing techniques as its computational requirements scale with the grid size instead of with the number of crystallographic orientations involved. In contrast to other sparse algorithms, the bounding box data structure naturally allows for semi-implicit and implicit time integration. The bounding box algorithm is further accelerated through parallelisation and extended to perform simulations with a more complex phase field model, which is possible through its object-oriented design.

So far, multigrid methods developed for phase field simulation of microstructural evolution only considered a few phase field variables. In **Chapter 6** a nonlinear multigrid solver is constructed based on the FAS scheme to solve the multi-phase field model for multiple phase field variables. Experiments with this solver show that the convergence rates are independent of the grid size. However, the convergence properties of the multigrid solver depend on the number of involved phase field variables: for larger numbers, the convergence deteriorates.

The applicability of the bounding box algorithm developed in Chapter 5 is illustrated in Chapter 7 and Chapter 8. To gain more insight in the pinning effect of second-phase particles on grain growth, the effect of spheroid particles is studied in **Chapter 7** by three-dimensional phase field simulations with the continuum field model. The pinning effect of a particle distribution is found to be stronger for increasing volume fraction, and for increasing aspect ratio. A notable result of the analysis of the simulation results is that in spite of the increase of the number of particles located at boundary junctions with increasing aspect ratio, for all simulated aspect ratios, a volume fraction dependence of the limiting mean grain size of $1/f_{\mathrm{V}}^{0.93}$ is obtained, which follows theoretical predictions. Furthermore, based on the simulation results, a generalised Zener type relation is proposed.

Chapter 8 presents preliminary results on phase field simulation with continuum field model (2.44), which models grain growth in a microstructure with anisotropic boundary energy properties. Three-dimensional grain growth simulations are performed in a microstructure with fibre texture, for an isotropic and for an anisotropic boundary energy function. The latter boundary energy function describes the boundary energy of the low-angle boundaries by a Read-Shockley dependence, while the high-angle boundaries are attributed a constant, higher energy quantity. The simulation results show that the low-angle boundaries are clearly preferred during grain growth in the microstructure with anisotropic boundary properties. The anisotropic boundary energy formulation is furthermore observed to change the individual growth rates of the grains as a function of the number of grain faces. The neutral number of faces, for which the grain size is stable, is found to be larger in the microstructure with anisotropic boundary properties than in the microstructure with isotropic boundary properties. However, a larger amount of grains is required to derive a reliable analysis of the effect of anisotropic boundary energy on grain growth from three-dimensional phase field simulations.

9.3 Contributions

The main contributions of this thesis are:

- the development of a sparse bounding box algorithm to perform efficient phase field simulations of grain growth in polycrystalline materials. The bounding box algorithm shows significant improvements over existing techniques, as its computational requirements scale with the size of the microstructure instead of with the number of crystallographic orientations involved.

- a parallel implementation of the bounding box algorithm, which enables large-scale simulations that were infeasible before.
- the development and implementation of a nonlinear multigrid solver based on the FAS scheme with convergence properties that are independent of the grid size.
- a study of the effect of the aspect ratio and the volume fraction of spheroid second-phase particles with constant properties on grain growth, by three-dimensional phase field simulations.
- three-dimensional simulation results on the effect of anisotropic boundary energy formulation on grain growth in a fibre-textured microstructure

9.4 Future research directions

There are different possible directions for future research, both in the development of computational acceleration strategies and in materials science.

Computational acceleration strategies

As mentioned in Section 3.3.3, the sparse algorithm of [133] exploits the property that the evolution of the phase field variables in a grain growth simulation is limited to the grain boundary regions. The algorithm accordingly distinguishes the evolving phase field values at grain boundaries from the constant phase field values inside the grains. In the same way, in the context of the bounding box algorithm, it might be possible to give instructions to the solver not to compute the phase field values that are already at equilibrium.

In Section 3.2.2, we described the merits of adaptive time stepping. Thanks to the drastic reduction of the involved number of unknowns by the sparse data structure of the bounding box algorithm, in combination with the fact that the cuboid shape of the bounding boxes closely resembles the structure of the underlying grid, the use of adaptive time stepping now seems more appropriate.

The current parallel implementation of the bounding box algorithm partitions the simulation work by assigning approximately equal amounts of bounding boxes to each involved processor. An improvement on this partitioning would be to take into account the actual size of the boxes. Moreover, a graph cut algorithm could be applied in order to minimise the connections between the grains and thus the communication between the processors. Also, the current implementation would benefit from dynamic load balancing.

The numerical experiments with the implicit time stepper described in Section 8.2.1 are limited. It would be interesting to see how other nonlinear methods behave when solving the anisotropic continuum field model. Furthermore, we would like to explore the connection of the bounding box algorithm with the theory of local defect correction methods.

It was briefly mentioned in Section 6.3.2 how the multigrid solver developed in Chapter 6 could benefit of the concept of locally active phase field variables. There are several ways in which the ideas of the bounding box algorithm and the developed multigrid solver could be combined. The multigrid solver could be integrated with the bounding box algorithm to solve the equations locally. Another possibility is to use the idea of delimiters to specify to the multigrid solver where the active grain regions are located.

Materials science research

In Chapter 7, the pinning effect of second-phase particles with a spheroid shape was studied. For further research, it would be interesting to study this effect for a larger range of aspect ratios and volume fractions, and different orientation distributions of the particles. In addition, the effect of evolving second-phase particles could be studied by extending the model equations. Simulations with this extended model could also be performed within the framework of the bounding box algorithm.

The simulation results presented in Chapter 8 are focused on the effect of anisotropic formulation of the grain boundary energy, as a function of the misorientation angle of a boundary. Further investigations could include the effect of anisotropic formulation of the grain boundary mobility. Another interesting topic is the effect of the inclination dependence of both the boundary energy and the boundary mobility on grain growth.

The description of the crystallography of the three-dimensional microstructures in Chapter 8 is limited to microstructures with fibre texture. For realistic materials, a more general, three-dimensional description of the crystallography will provide more insight into the interplay of texture and grain growth.

Bibliography

[1] Abbruzzese, G. Computer simulated grain growth stagnation. *Acta Metall. 33* (1985), 1329–1337.

[2] Anand, L., and Gurland, J. The relationship between the size of cementite particles and the subgrain size in quenched-and-tempered steels. *Metall. Trans. A 6*, 4 (1975), 928–931.

[3] Anderson, M., Grest, G., Doherty, R., Li, K., and Srolovitz, D. Inhibition of grain growth by second phase particles: Three dimensional Monte Carlo computer simulations. *Scr. Metall. 23*, 5 (1989), 753–758.

[4] Anselmino, E. *Microstructural effects on grain boundary motion in Al-Mn alloys.* PhD thesis, Technische Universiteit Delft, 2007.

[5] Apel, M., Böttger, B., Rudnizki, J., Schaffnit, P., and Steinbach, I. Grain growth simulations including particle pinning using the multiphase-field concept. *ISIJ Int. 49*, 7 (2009), 1024–1029.

[6] Ascher, U. M., Ruuth, S. J., and Wetton, B. T. R. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal. 32*, 3 (1995), 797–823.

[7] Barmak, K., Emelianenko, M., Golovatky, D., Kinderlehrer, D., and Ta'asan, S. A new perspective on texture evolution. *Int. J. Numer. Anal. Model. 5* (2004), 93–108.

[8] Barmak, K., Kim, J., Kim, C.-S., Archibald, W., Rohrer, G., Rollett, A., Kinderlehrer, D., Ta'asan, S., Zhang, H., and Srolovitz, D. J. Grain boundary energy and grain growth in Al films: Comparison of experiments and simulations. *Scr. Mater. 54* (2006), 1059–1063.

[9] Bates, P. W., and Fife, P. C. Spectral comparison principles for the Cahn-Hilliard and phase-field equations, and time scales for coarsening. *Physica D 43*, 2-3 (1990), 335–348.

[10] Baňas, L., and Nürnberg, R. Finite element approximation of a three dimensional phase field model for void electromigration. *J. Sci. Comput. 37*, 2 (2008), 202–232.

[11] Beckett, G., Mackenzie, J. A., and Robertson, M. L. An r-adaptive finite element method for the solution of the two-dimensional phase-field equations. *Commun. Comput. Phys. 1*, 5 (2006), 805–826.

[12] Belytschko, T., Gracie, R., and Ventura, G. A review of extended/generalized finite element methods for material modeling. *Modelling Simul. Mater. Sci. Eng. 17* (2009), 043001.

[13] Bollman, W. *Crystal Defects and Crystalline Interfaces.* Springer-Verlag, 1970.

[14] Böttger, B., Eiken, J., and Steinbach, I. Phase field simulation of equiaxed solidification in technical alloys. *Acta Mater. 54* (2006), 2697–2704.

[15] Brahme, A., Fridy, J., Weiland, H., and Rollett, A. D. Modeling texture evolution during recrystallization in aluminum. *Modelling Simul. Mater. Sci. Eng. 17* (2009), 015005.

[16] Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Math. Comput. 31* (1977), 333–390.

[17] Braun, R. J., and Murray, B. T. Adaptive phase-field computations of dendritic crystal growth. *J. Cryst. Growth 174* (1997), 41–53.

[18] Bréchet, Y., and Militzer, M. A note on grain size dependent pinning. *Scr. Mater. 52* (2005), 1299–1303.

[19] Bronchard, Q., Bouar, Y. L., and Finel, A. Quantitative phase field modeling of precipitation processes. *Adv. Eng. Mater. 8* (2006), 1245–1248.

[20] Burke, J., and Turnbull, D. Recrystallization and grain growth. *Prog. Met. Phys. 3* (1952), 220–292.

[21] Ceniceros, H. D., and Roma, A. M. A nonstiff, adaptive mesh refinement-based method for the Cahn-Hilliard equation. *J. Comput. Phys. 225*, 2 (2007), 1849–1862.

[22] Chang, K., Feng, W., and Chen, L.-Q. Effect of second-phase particle morphology on grain growth kinetics. *Acta Mater. 57*, 17 (2009), 5229–5236.

[23] Chapa, M., Fernández, B., Medina, S. F., and López, V. Influence of Al and Nb on optimum Ti/N ratio in controlling austenite grain growth at reheating temperatures. *ISIJ Int. 42*, 11 (2002), 1288–1296.

[24] Chen, L.-Q., and Shen, J. Applications of semi-implicit Fourier-spectral method to phase field equations. *Comput. Phys. Commun. 108* (1998), 148–158.

[25] Chen, L.-Q., and Yang, W. Computer simulation of the domain dynamics of a quenched system with a large number of nonconserved order parameters: The grain-growth kinetics. *Phys. Rev. B 50*, 21 (1994), 15752–15756.

[26] Cherukuri, B., Srinivasan, R., Tamirisakandala, S., and Miracle, D. B. The influence of trace boron addition on grain growth kinetics of the beta phase in the beta titanium alloy Ti-15Mo-2.6Nb-3Al-0.2Si. *Scr. Mater. 60*, 7 (2009), 496–499.

[27] Cooley, J. W., and Tukey, J. W. An algorithm for the machine calculation of complex Fourier series. *Math. Comput. 19*, 90 (1965), 297–301.

[28] Copetti, M. I. M., and Elliot, C. M. Kinetics of phase decomposition processes: Numerical solutions to the Cahn-Hilliard equation. *Mater. Sci. Technol. 6* (1990), 273–283.

[29] Couturier, G., Doherty, R., Maurice, C., and Fortunier, R. 3D finite element simulation of the inhibition of normal grain growth by particles. *Acta Mater. 53*, 4 (2005), 977–989.

[30] Danilov, D., and Nestler, B. Phase-field modelling of solute trapping during rapid solidification of a Si-As alloy. *Acta Mater. 54*, 18 (2006), 4659–4664.

[31] Demirel, M. C., Kuprat, A. P., George, D. C., Straub, G. K., and Rollett, A. D. Linking experimental characterization and computational modeling of grain growth in Al-foil. *Interface Sci. 10* (2002), 137–141.

[32] Doherty, R. D., Srolovitz, D. J., Rollett, A. D., and Anderson, M. P. On the volume fraction dependence of particle limited grain growth. *Scr. Metall. 21* (1987), 675–679.

[33] Dorr, M., Fattebert, J.-L., Wickett, M., Belak, J., and Turchi, P. A numerical algorithm for the solution of a phase-field model of polycrystalline materials. *J. Comput. Phys. 229*, 3 (2010), 626–641.

[34] Elsey, M., Esedoglu, S., and Smereka, P. Diffusion generated motion for grain growth in two and three dimensions. *J. Comput. Phys. 228*, 21 (2009), 8015–8033.

[35] Fan, D., and Chen, L.-Q. Computer simulation of grain growth using a continuum field model. *Acta Mater. 45* (1997), 611–622.

[36] Fan, D., and Chen, L.-Q. Computer simulation of topological evolution in 2-D grain growth using a diffuse-interface field model. *Acta Mater. 45* (1997), 1115–1126.

[37] Fan, D., Chen, L.-Q., and Chen, S.-P. P. Numerical simulation of Zener pinning with growing second phase particles. *J. Am. Ceram. Soc. 81* (1998), 526–532.

[38] Feng, W. M., Yu, P., Liu, Z. K., Du, Q., and Chen, L.-Q. Spectral implementation of an adaptive moving mesh method for phase-field equations. *J. Comput. Phys. 220* (2006), 498–510.

[39] FFTW. `http://www.fftw.org/`.

[40] Gangulee, A., and D'Heurle, F. M. Anomalous large grains in alloyed aluminum thin films - 2. Electromigration and diffusion in thin films with very large grains. *Thin Solid Films 16*, 2 (1973), 227–236.

[41] Gao, J., Thompson, R. G., and Patterson, B. R. Computer simulation of grain growth with second phase particle pinning. *Acta Mater. 45*, 9 (1997), 3653–3658.

[42] Garcke, H., Nestler, B., and Stinner, B. A diffuse interface model for alloys with multiple components and phases. *SIAM J. Appl. Math. 64* (2004), 775–799.

[43] Glazier, J. A. Grain growth in three dimensions depends on grain topology. *Phys. Rev. Lett. 70* (1993), 2170–2173.

[44] Gruber, J., George, D., Kuprat, A., Rohrer, G., and Rollett, A. Effect of anisotropic interfacial energy on grain boundary distributions during grain growth. *Scr. Mater. 53* (2005), 351–355.

[45] Gruber, J., Ma, N., Wang, Y., Rollett, A. D., and Rohrer, G. S. Sparse data structure and algorithm for the phase field method. *Modelling Simul. Mater. Sci. Eng. 14*, 14 (2006), 1189–1195.

[46] Gruber, J., Miller, H., Hoffmann, T., Rohrer, G., and A.D.Rollett. Misorientation texture development during grain growth. Part I: Simulation and experiment. *Acta Mater. 57*, 20 (2009), 6102–6112.

[47] Guo, M. *Influence of Dissolved Cerium and Inclusion Containing Cerium on Solidification Structure and Austenite Grain Growth in Iron-Carbon Alloy.* PhD thesis, Tohoku University, 1998.

[48] Guo, M., and Suito, H. Austenite grain size distribution and topological properties in Fe-0.09 to 0.53 mass%C-0.02 mass%P alloys containing primary inclusions of Ce_2O_3 and CeS. *ISIJ Int. 39*, 12 (1999), 1297–1303.

[49] Harun, A., Holm, E. A., Clode, M. P., and Miodownik, M. A. On computer simulation methods to model Zener pinning. *Acta Mater. 54* (2006), 3261–3273.

[50] Hassold, G. N., Holm, E. A., and Miodownik, M. A. Accumulation of coincidence site lattice boundaries during grain growth. *Mater. Sci. Technol. 19* (2003), 683–687.

[51] Hellman, P., and Hillert, M. On the effect of second-phase particles on grain growth. *Scand. J. Metall. 4*, 5 (1975), 211–219.

[52] Hibbard, G. D., McCrea, J. L., Palumbo, G., Aust, K. T., and Erb, U. An initial analysis of mechanisms leading to late stage abnormal grain growth in nanocrystalline Ni. *Scr. Mater. 47*, 2 (2002), 83–87.

[53] Hilgenfeldt, S., Kraynik, A. M., Koehler, S. A., and Stone, H. A. An accurate von Neumann's law for three-dimensional foams. *Phys. Rev. Lett. 86* (2001), 2685.

[54] Hillert, M. On the theory of normal and abnormal grain growth. *Acta Metall. 13* (1965), 227–238.

[55] Hillert, M. Inhibition of grain growth by second-phase particles. *Acta Metall. 36*, 12 (1988), 3177–3181.

[56] Holm, E. A., Hassold, G. N., and Miodownik, M. On misorientation distribution evolution during anisotropic grain growth. *Acta Mater. 49* (2001), 2981–2991.

[57] Holm, E. A., Miodownik, M., and Rollett, A. D. On abnormal subgrain growth and the origin of recrystallization nuclei. *Acta Mater. 51* (2003), 2701–2716.

[58] Humphreys, F. J., and Hatherly, M. *Recrystallization and related annealing phenomena.* Elsevier, Oxford, 1995.

[59] Hunderi, O., and Ryum, N. On the stagnation of grain growth. *Acta Metall. 30* (1982), 739–742.

[60] Hundsdorfer, W., and Verwer, J. G. *Numerical solution of time-dependent advection-diffusion-reaction equations.* Springer-Verlag Berlin Heidelberg, 2003.

[61] Janssens, K. An introductory review of cellular automata modeling of moving grain boundaries in polycrystalline materials. *Math. Comput. Simul. 80*, 7 (2010), 1361–1381.

[62] Kay, D., and Welford, R. A multigrid finite element solver for the Cahn-Hilliard equation. *J. Comput. Phys. 212*, 1 (2006), 288–304.

[63] Kazaryan, A., Wang, Y., Dregia, S. A., and Patton, B. R. Grain growth in anisotropic systems: comparison of effects of energy and mobility. *Acta Mater. 50* (2002), 2491–2502.

[64] Kim, J., Kang, K., and Lowengrub, J. Conservative multigrid methods for Cahn-Hilliard fluids. *J. Comput. Phys. 193*, 2 (2004), 511–543.

[65] Kim, S. G., Kim, D. I., Kim, W. T., and Park, Y. B. Computer simulations of two-dimensional and three-dimensional ideal grain growth. *Phys. Rev. E 74*, 6 (2006), 061605.

[66] Kim, S. G., Kim, W. T., Suzuki, T., and Ode, M. Phase-field modeling of eutectic solidification. *J. Cryst. Growth 261*, 1 (2004), 135–158.

[67] Kim, Y., Hwang, S., Kim, M., Kwun, S., and Chae, S. Three-dimensional Monte-Carlo simulation of grain growth using triangular lattice. *Mater. Sci. Eng., A 408* (2005), 110–120.

[68] Kimpe, D. *On the Scalable Parallelization of Unstructured Mesh Applications*. PhD thesis, Katholieke Universiteit Leuven, 2008.

[69] Kinderlehrer, D., Livshits, I., and Ta'asan, S. A variational approach to modeling and simulation of grain growth. *SIAM J. Sci. Comput. 28*, 5 (2006), 1694–1715.

[70] Knowlton, B. D., Clement, J. J., and Thompson, C. V. Simulation of the effects of grain structure and grain growth on electromigration and the reliability of interconnects. *J. Appl. Phys. 81*, 9 (1997), 6073–6080.

[71] Kobayashi, R., Warren, J. A., and Carter, W. C. Vector-valued phase field model for crystallization and grain boundary formation. *Physica D 119* (1998), 415–423.

[72] Koch, C. C., Scattergood, R. O., Darling, K. A., and Semones, J. E. Stabilization of nanocrystalline grain sizes by solute additions. *J. Mater. Sci. 43*, 23-24 (2008), 7264–7272.

[73] Kornhuber, R., and Krause, R. Robust multigrid methods for vector-valued Allen-Cahn equations with logarithmic free energy. *Comput. Visual Sci. 9* (2006), 103–116.

[74] Krill III, C. E., and Chen, L.-Q. Computer simulation of 3-D grain growth using a phase-field model. *Acta Mater. 50* (2002), 3057–3073.

[75] Li, W.-B., and Easterling, K. E. The influence of particle shape on Zener drag. *Acta Metall. Mater. 38*, 6 (1990), 1045–1052.

[76] Longworth, H. P., and Thompson, C. V. Abnormal grain growth in aluminum alloy thin films. *J. Appl. Phys. 69*, 7 (1991), 3929–3940.

[77] Ma, E. Eight routes to improve the tensile ductility of bulk nanostructured metals and alloys. *JOM 58*, 4 (2006), 49–53.

[78] Ma, N., Kazaryan, A., Dregia, S. A., and Wang, Y. Computer simulation of texture evolution during grain growth: effect of boundary properties and initial microstructure. *Acta Mater. 52*, 13 (2004), 3869–3879.

[79] Mackenzie, J. A., and Robertson, M. L. A moving mesh method for the solution of the one-dimensional phase-field equations. *J. Comput. Phys. 181*, 2 (2002), 526–544.

[80] MacPherson, R. D., and Srolovitz, D. J. The von Neumann relation generalized to coarsening of three-dimensional microstructures. *Nature 446* (2007), 1053–1055.

[81] Manohar, P. A., Ferry, M., and Chandra, T. Five Decades of the Zener Equation. *ISIJ Int. 38*, 9 (1998), 913–924.

[82] McMahon, Jr., C., and Graham, Jr., C. *Introduction to Engineering Materials: The Bicycle and the Walkman*, 2 ed. Merion Books, 2000.

[83] Miodownik, M., Holm, E. A., and Hassold, G. N. Highly parallel computer simulations of particle pinning: Zener vindicated. *Scr. Mater. 42*, 12 (2000), 1173–1177.

[84] Miodownik, M. A. Grain boundary engineering with particles. *Scr. Mater. 54*, 40 (2006), 993–997.

[85] Moelans, N. *Phase-field simulations of grain growth in materials containing second-phase particles.* PhD thesis, Katholieke Universiteit Leuven, 2006.

[86] Moelans, N., Blanpain, B., and Wollants, P. A phase field model for the simulation of grain growth in materials containing finely dispersed incoherent second-phase particles. *Acta Mater. 53* (2005), 1771–1781.

[87] Moelans, N., Blanpain, B., and Wollants, P. Phase field simulations of grain growth in two-dimensional systems containing finely dispersed second-phase particles. *Acta Mater. 54* (2006), 1175–1184.

[88] Moelans, N., Blanpain, B., and Wollants, P. Pinning effect of second-phase particles on grain growth in polycrystalline films studied by 3-D phase field simulations. *Acta Mater. 55* (2007), 2173–2182.

[89] Moelans, N., Blanpain, B., and Wollants, P. Quantitative analysis of grain boundary properties in a generalized phase field model for grain growth in anisotropic systems. *Phys. Rev. B 78* (2008), 024113.

[90] Moelans, N., Blanpain, B., and Wollants, P. Quantitative phase-field approach for simulating grain growth in anisotropic systems with arbitrary inclination and misorientation dependence. *Phys. Rev. Lett. 101* (2008), 025502.

[91] Moelans, N., Spaepen, F., and Wollants, P. Grain growth in thin films with a fiber texture studied by phase-field simulations and mean field modeling. *Philos. Mag. 90*, 1 (2010), 501–523.

[92] Moelans, N., Wendler, F., and Nestler, B. Comparative study of two phase-field models for grain growth. *Comput. Mater. Sci. 46*, 2 (2009), 479–490.

[93] Morton, K. W., and Mayers, D. F. *Numerical Solution of Partial Differential Equations.* Cambridge University Press, 1994.

[94] MPI. `http://www.mpi-forum.org/docs/`.

[95] Mullins, W. Estimation of the geometrical rate constant in idealized three dimensional grain growth. *Acta Metall. 37* (1989), 2979–2984.

[96] Nes, E., Ryum, N., and Hunderi, O. On the Zener drag. *Acta Metall. 33*, 1 (1985), 11–22.

[97] Nestler, B. A 3D parallel simulator for crystal growth and solidification in complex alloy systems. *J. Cryst. Growth 275* (2005), e273–e278.

[98] Nestler, B., Danilov, D., and Galenko, P. Crystal growth of pure substances: Phase-field simulations in comparison with analytical and experimental results. *J. Comput. Phys. 207* (2005), 221–239.

[99] Nestler, B., Garcke, H., and Stinner, B. Multicomponent alloy solidification: Phase-field modeling and simulations. *Phys. Rev. E 71* (2005), 041609.

[100] Ohta, H., Inoue, R., and Suito, H. Effect of TiN precipitates on austenite grain size in Fe-1.5%Mn-0.12%Ti-Si($<$ 1.1%)-C(0.05 and 0.15%) alloy. *ISIJ Int. 48*, 3 (2008), 294–300.

[101] Ono, N., Kimura, K., and Watanabe, T. Monte Carlo simulation of grain growth with the full spectra of grain orientation and grain boundary energy. *Acta Mater. 47*, 3 (1999), 1007–1017.

[102] OpenMP. `http://openmp.org/`.

[103] Plapp, M. Three-dimensional phase-field simulations of directional solidification. *J. Cryst. Growth 303* (2007), 49–57.

[104] Provatas, N., Goldenfeld, N., and Dantzig, J. Efficient computation of dendritic microstructures using adaptive mesh refinement. *Phys. Rev. Lett. 80*, 15 (1998), 3308–3311.

[105] Provatas, N., Goldenfeld, N., and Dantzig, J. Adaptive mesh refinement computation of solidification microstructures using dynamic data structures. *J. Comput. Phys. 148* (1999), 265–290.

[106] Provatas, N., Greenwood, M., Athreya, B., Goldenfeld, N., and Dantzig, J. Multiscale modeling of solidification: Phase-field methods to adaptive mesh refinement. *IJMPB 19* (2005), 4525–4565.

[107] Read, W. T. *Dislocations in Crystals.* McGraw-Hill, 1953.

[108] Read, W. T., and Shockley, W. Dislocation models of crystal grain boundaries. *Phys. Rev. 78*, 3 (1950), 275–289.

[109] Riege, S. P., Thompson, C. V., and Frost, H. J. Simulation of the influence of particles on grain structure evolution in two-dimensional systems and thin films. *Acta Mater. 47*, 6 (1999), 1879–1887.

[110] Ringer, S. P., Li, W. B., and Easterling, K. E. On the interaction and pinning of grain boundaries by cubic shaped precipitate particles. *Acta Metall. 37*, 3 (1989), 831–841.

[111] Rosam, J., Jimack, P. K., and Mullis, A. M. A fully implicit, fully adaptive time and space descretisation method for phase-field simulation of binary alloy solidification. *J. Comput. Phys. 225* (2007), 1271–1287.

[112] Rosam, J., Jimack, P. K., and Mullis, A. M. An adaptive, fully implicit multigrid phase-field model for the quantitative simulation of non-isothermal binary alloy solidification. *Acta Mater. 56*, 17 (2008), 4559–4569.

[113] Ruuth, S. J. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *J. Math. Biol. 34*, 2 (1995), 148–176.

[114] Ryum, N., Hunderi, O., and Nes, E. On grain boundary drag from second phase particles. *Scr. Mater. 17* (1983), 1281–1283.

[115] Sandia National Laboratories. `http://www.sandia.gov/news/resources/releases/2006/nanosimulation.html`, June 2006.

[116] Smith, C. *Trans. Metall. Soc. AIME 175* (1948), 15.

[117] Song, K., and Aindow, M. Grain growth and particle pinning in a model Ni-based superalloy. *Mater. Sci. Eng., A 479*, 1-2 (2008), 365–372.

[118] Soucail, M., Messina, R., Cosnuau, A., and Kubin, L. P. Monte Carlo simulation of Zener pinning in two dimensions. *Mater. Sci. Eng., A 271*, 1-2 (1999), 1–7.

[119] Steinbach, I., and Pezzolla, F. A generalized field method for multiphase transformations using interface fields. *Physica D 134*, 4 (1999), 385–393.

[120] Steinbach, I., Pezzolla, F., Nestler, B., elberg, M. S., Prieler, R., Schmitz, G. J., and Rezende, J. L. L. A phase field concept for multiphase systems. *Physica D 94* (1996), 135–147.

[121] Sun, N., Patterson, B. R., Suni, J. P., Weiland, H., and Allard, L. F. Characterization of particle pinning potential. *Acta Mater. 54* (2006), 4091–4099.

[122] Suwa, Y., Saito, Y., and Onodera, H. Phase field simulation of grain growth in three dimensional system containing finely dispersed second-phase particles. *Scr. Mater. 55* (2006), 407–410.

[123] Suwa, Y., Saito, Y., and Onodera, H. Phase-field simulation of abnormal grain growth due to inverse pinning. *Acta Mater. 55* (2007), 6881–6894.

[124] Suwa, Y., Saito, Y., and Onodera, H. Three-dimensional phase field simulation of the effect of anisotropy in grain-boundary mobility on growth kinetics and morphology of grain structure. *Comput. Mater. Sci. 40* (2007), 40–50.

[125] Tan, Z., Tang, T., and Zhang, Z. A simple moving mesh method for one-and two-dimensional phase-field equations. *J. Comput. Appl. Math. 190*, 1 (2006), 252–269.

[126] Trefethen, L. N. *Spectral Methods in MATLAB.* Oxford University, 2000.

[127] Upmanyu, M., Hassold, G., Kazaryan, A., Holm, E., Wang, Y., Patton, B., and Srolovitz, D. Boundary mobility and energy anisotropy effects on microstructural evolution during grain growth. *Interface Sci. 10* (2002), 201–216.

[128] van Dam, A. *Go with the flow. Moving meshes and solution monitoring for compressible flow simulation.* PhD thesis, Universiteit Utrecht, 2009.

[129] Vanherpe, L., Moelans, N., Blanpain, B., and Vandewalle, S. Bounding box algorithm for three-dimensional phase-field simulations of microstructural evolution in polycrystalline materials. *Phys. Rev. E 76* (2007), 056702.

[130] Vanherpe, L., Moelans, N., Blanpain, B., and Vandewalle, S. Three-dimensional phase field simulations of grain growth in materials containing finely dispersed second-phase particles. *Proc. Appl. Math. Mech. 7* (2007), 2020001–2020002.

[131] Vanherpe, L., Moelans, N., Blanpain, B., and Vandewalle, S. Pinning effect of spheroid second-phase particles on grain growth studied by three-dimensional phase-field simulations. *Comput. Mater. Sci. 49*, 2 (2010), 340–350.

[132] Vanherpe, L., Wendler, F., Nestler, B., and Vandewalle, S. A multigrid solver for phase field simulation of microstructure evolution. *Math. Comput. Simul. 80*, 7 (2010), 1438–1448.

[133] Vedantam, S., and Patnaik, B. S. V. Efficient numerical algorithm for multiphase field simulations. *Phys. Rev. E 73* (2006), 016703.

[134] von Neumann, J. *Metal interfaces.* American Society for Metals, 1952.

[135] Wang, D. Variable step-size implicit-explicit linear multistep methods for time-dependent PDEs. Master's thesis, Simon Fraser University, 2005.

[136] Warren, J. A., Kobayashi, R., Lobkovsky, A. E., and Carter, W. C. Extending phase field models of solidification to polycrystalline materials. *Acta Mater. 51* (2003), 6035–6058.

[137] Wen, Y., Wang, B., Simmons, J., and Wang, Y. A phase-field model for heat treatment applications in Ni-based alloys. *Acta Mater. 54* (2006), 2087–2099.

[138] Weygand, D., Bréchet, Y., and Lépinoux, J. Inhibition of grain growth by particle distribution: effect of spatial heterogeneities and of particle strength dispersion. *Mater. Sci. Eng., A 292*, 1 (2000), 34–39.

[139] Wise, S., Kim, J., and Lowengrub, J. Solving the regularized, strongly anisotropic Cahn-Hilliard equation by an adaptive nonlinear multigrid method. *J. Comput. Phys. 226*, 1 (2007), 414–446.

[140] Zöllner, D., and Streitenberger, P. Three-dimensional normal grain growth: Monte Carlo Potts model simulation and analytical mean field theory. *Scr. Mater. 54*, 9 (2006), 1697–1702.

Scientific résumé

Education

2005-2010: PhD in Applied Sciences and Engineering: Computer Science
Department of Computer Science, Katholieke Universiteit Leuven, Belgium
PhD thesis: *Acceleration strategies for phase field simulation of grain growth in polycrystalline materials*

2002-2006: Academic Teacher Training: Applied Sciences and Engineering
Faculty of Engineering, Katholieke Universiteit Leuven, Belgium

2002-2005: Master of Applied Sciences and Engineering: Computer Science
Department of Computer Science, Katholieke Universiteit Leuven, Belgium
Master's thesis: *Numerical modelling and simulation of grain growth*

2000-2002: Bachelor of Applied Sciences and Engineering
Faculty of Engineering, Katholieke Universiteit Leuven, Belgium

Teaching

2005-2009: Teaching Assistant of *Modellering en Simulatie* (G0Q57A)

2006-2009: Teaching Assistant of *Numerieke Wiskunde* (H01D8A)

2005-2008: Ombudsperson of Master *Computerwetenschappen*

2006-2009: Ombudsperson of Master *Wiskundige Ingenieurstechnieken*

Articles in internationally reviewed scientific journals

- L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Pinning effect of spheroid second-phase particles on grain growth studied by three-dimensional* *phase field simulations*, Computational Materials Science, 49(2):340–350 DOI:10.1016/J.COMMATSCI.2010.05.020

- L. Vanherpe, F. Wendler, B. Nestler, S. Vandewalle, *A multigrid solver for phase field simulation of microstructure evolution*, Mathematics and Computers in Simulation, 80(7):1438–1448, 2010
  DOI:10.1016/J.MATCOM.2009.10.007
- L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Three-dimensional phase field simulations of grain growth in materials containing finely dispersed second-phase particles*, Proceedings in Applied Mathematics and Mechanics, 7(1):2020001–2020002, 2008
  DOI:10.1002/PAMM.200700007
- L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Bounding box algorithm for three-dimensional phase field simulations of microstructural evolution in polycrystalline materials*, Physical Review E, Statistical, Nonlinear and Soft Matter Physics, 76(5):056702, 2007
  DOI:10.1103/PHYSREVE.76.056702

Publications in international conference proceedings

- N. Moelans, A. Serbruyns, J. Heulens, B. Blanpain, P. Wollants, L. Vanherpe, S. Vandewalle, B. Rodiers, *Quantitative phase-field simulations of coarsening and growth in complex systems*, Materials Science and Technology, 494–505, 2008

Presentations at conferences and symposia

- International Congress on Computational and Applied Mathematics, Leuven, Belgium, 5-9 July 2010
  TALK: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Three-dimensional phase field simulations of grain growth in a polycrystalline microstructure with anisotropic grain boundary properties*
- 2nd Symposium on Phase-Field Modelling in Materials Science, Aachen, Germany, 30 August - 2 September 2009
  TALK: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *The effect of ellipsoid second-phase particles on grain growth studied by three-dimensional phase field simulations*
- SIAM Conference on Mathematical Aspects of Materials Science, Philadelphia (PA), USA, 11-14 May 2008

Co-organiser of Mini-symposium: *Towards Realistic Three-dimensional Phase-field Simulations of the Evolution of Polycrystalline Structures*
Talk: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Three-dimensional Phase Field Simulations of Grain Growth in Anisotropic Systems*

- International Congress on Computational and Applied Mathematics, Gent, Belgium, 7-11 July 2008
  Talk: L. Vanherpe, F. Wendler, B. Nestler, S. Vandewalle, *A multigrid solver for phase-field simulation of microstructure evolution*

- Workshop *Phase-field models for the evolution of complex structures*, Institut Henri Poincaré, Paris, France, 4-6 June 2007
  Poster: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Bounding box algorithm for three-dimensional phase field simulations of grain growth*
  Poster: N. Moelans, L. Vanherpe, B. Blanpain, P. Wollants, *Phase field simulations of grain growth in the presence of second-phase particles that evolve in time*

- 6th International Congress on Industrial and Applied Mathematics, Zürich, Switzerland, 16-20 July 2007
  Talk: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Numerical simulation of three-dimensional grain growth in materials containing finely dispersed second-phase particles with the phase field method*

- Symposium *High Performance Computing @ K.U.Leuven*, Kasteel Arenberg, Heverlee, Belgium, 27 April 2007
  Talk: L. Vanherpe, N. Moelans, B. Blanpain, S. Vandewalle, *Three-dimensional simulations of grain growth in polycrystalline materials with the phase field method*

- Polymorphism in Condensed Matter, Max Planck Institute for the Physics of Complex Systems, Dresden, Germany, 13-17 November 2006
  Poster: L. Vanherpe, S. Vandewalle, B. Blanpain, *Numerical modeling and simulation of grain growth*

- Thirty-first Conference of the Dutch-Flemish Numerical Analysis Communities, Zeist, The Netherlands, 11-13 October 2006
  Poster: L. Vanherpe, S. Vandewalle, B. Blanpain, *Numerical modeling and simulation of grain growth*

- IAP (Interuniversity Attraction Pole) Study Day, Château-Ferme de Profondval, Court St. Etienne, Belgium, 16 May 2006
  Poster: L. Vanherpe, S. Vandewalle, B. Blanpain, *Numerical modeling and simulation of grain growth*

Research visit

- Research visit to the Institute of Computational Engineering at the Karlsruhe University of Applied Sciences, with Prof. Britta Nestler, Germany, 16 March - 3 May 2008
  Travel grant: Travel grant of the FWO (Research Foundation Flanders, Belgium)
  Resulting paper: L. Vanherpe, F. Wendler, B. Nestler, S. Vandewalle, *A multigrid solver for phase field simulation of microstructure evolution*, Mathematics and Computers in Simulation, 80(7):1438–1448, 2010

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science
Research group Scientific Computing
Celestijnenlaan 200A box 2402, BE-3001 HEVERLEE (Belgium)

KATHOLIEKE UNIVERSITEIT
LEUVEN

ASSOCIATIE K.U.LEUVEN