Novel Cellular-Automata (CA) based modelling for dynamic recrystallization and grain growth in materials - theory and applications

Department of Automatic Control & Systems Engineering

The University of Sheffield

By:

Daliya Aflyatunova

Supervisor:

Prof. Mahdi Mahfouf

Submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

October 2018

Dedicated to the memory of
my beloved grandparents.

Vita

2007-2011	Theoretical Physics (BSc)
	The Udmurt State University, Russia
2011-2013	Material Science (MSc)
	The Udmurt State University, Russia
2014-2018	PhD sudent
	The University of Sheffield, UK

Publications and Presentations

1. D. Aflyatunova, M. Mahfouf. A validated cellular-automata model for normal and abnormal grain coarsening using local transition functions with optimal cell connectivity. *8th International Conference on Multiscale Materials Modeling, 9–14 October 2016, Dijon, France* (abstract);

2. D. Aflyatunova, M. Mahfouf. A Cellular-automata Model for Dynamical Recrystallization with Using the Cell-orientated Grain Boundary Velocities, *ICME 2017, Ypsilanti, Michigan, USA* (abstract, oral presentation).

Abstract

The mechanical properties of alloys depend on their microstructure, or grain sizes, which are obtained during processes of the grain growth and recrystallization. However, experiments do not allow to observe all details of the process and there is no theoretical model to predict these. This is why computer modelling allows for the prediction of microstructure. Cellular Automata (CA) is known to be used to for the simulation of the grain growth. However, (CA)-based models usually work only on the specific range of parameter, because there is no linear relationship exists between cellular automata and the physical parameters.

This thesis describes a newly developed space and time realistic Cellular Automata technique for the solving of various differential equations. Five types of differential equations, which have been considered in this work, describe the grain boundary movement in the processes of recrystallization and grain growth. These differential equations arise from a combination of the main driving forces: curvature and stored energy difference. The Cellular Automata technique is designed to be space and time realistic and is capable of taking into account the movement of cells in the triple junctions. The thesis includes numerous simulation results which have been compared with the analytical solutions for each type of equation. The problems associated with the reversibility of Cellular Automata are discussed with their relationships with the errors of the simulations. The thesis contains several original contributions in the various aspects of Cellular Automata simulation principles, including a new type of Margolus of neighbourhood.

It has been shown that the model is capable of solving the associated differential equations, which describe grain boundary movement, but with the errors that are related to the square grid. The newly suggested type of neighbourhoods allowed to eliminate the effects of the grid and obtain better prediction results.

Acknowledgements

I would like to express my sincerest gratitude to my supervisor Professor Mahdi Mahfouf for his support, patience and guidance during the course. Thank you for your emails, meetings and discussions. I cannot thank you enough for helping me improve.

I gratefully acknowledge all the academic and administrative staff of the Department of Automatic Control and Systems Engineering, the University of Sheffield for their support and Faculty of Engineering for providing me a scholarship.

I am especially grateful to Dr Mikhail Krivilev who taught me most valuable skills during my undergraduate studies. I also wish to thank my friends in Sheffield, who made my time during the course truly enjoyable.

To my mother and my aunt, thank you so much for your love, endless support and for having faith in me. I love you very much. Thank you.

Contents

List of Figures

Chapter 2 7

Chapter 3 57

Chapter 4 **111**

Chapter 5 **137**

List of Tables

Abbreviations

AGG Abnormal Grain Growth

CA Cellular Automata

CAS Cellular Automata Step

CDRX Continuous Dynamic Recrystallization

DDRX Discontinuous Dynamic Recrystallization

DFA Deterministic Finite Automaton

DRX Dynamic Recrystallization

DSMC Direct Simulation Monte Carlo

EBSD Electron Backscatter Diffraction

KM Kocks-Mecking

KWC Model of Kobayashi, Warren and Carter

LGCA Lattice Gas Cellular Automata

LTF Local Transition Function

MPF Multi-Phase-Field

PCA Probabilistic or Stochastic Cellular Automata

SFE Stacking-fault energy

SRX Static Recrystallization

STF State Transition Function

JMAK Johnson-Mehl-Avrami-Kolnogorov

Nomenclature

d_{cell}	cell size
n_{cells}	number of cells per initial grain
ρ	dislocation density
v	grain boundary velocity
p	driven force
f_i^n	fraction for ith neighbour at nth iteration
Δt_n	time increment for nth iteration
r	radius
t	time

Chapter 1

Introduction

1.1 Motivation

When it comes to the solving of the differential equations, cellular automata (CA) is not the first in the list to think about. However, CA techniques have been used for those problems that involve differential equations for a long time. For example, the lattice gas model has found application in solving the Navier-Stokes equations. Others applications of CA are processes of diffusion, reaction-diffusion equations, wave equations. Cellular automata have been widely used for the modelling of the grain growth and recrystallization in materials.

The mechanical properties of steels, such as hardness, strength, toughness, brittleness etc. strongly depend on their microstructure (grain size distribution), which is normally obtained during thermomechanical processing. Experiments do not allow to observe other important details of the process (for example, the atomistic phenomena involved in the boundary migration that occurs at high temperature) because of the associated rapid dynamics. Therefore, the most appropriate instrument for studying the grain growth may rest in computer-based modelling.

Direct simulations via the Monte Carlo method or cellular automata (CA) allow to include the topological requirements inherently, which would give them the advantage over the statistical and theoretical methods. CA is considered as the preferred method, as it is scalable, flexible and versatile to arbitrary time and space systems. CA [2] is the method for computer modelling that is based on applying local transition rules, which determine the future state of the cell, according to the states of its neighbours. One of the recently developed approaches consists of applying local transition functions (LTF) [3] as well as using Monte Carlo steps for the decision of changing a cell at each time step.

The main factors influencing the grain growth that should be considered are temperature, solutes and particles, specimen size and texture [4]. The driving force for grain growth is based on the grain boundaries surface energy. The total surface energy is lowered, because the grain boundary area decreases during grain growth due to the increase in their sizes and the reduction of their numbers [5]. This is the ideal case, as grain boundary motion is driven only by local curvature. The grain growth law follows from the proportionality of the growth rate $\frac{dD}{dt}$ to the curvature of the cell walls $c \sim \frac{1}{D}$:

$$\frac{dD}{dt} = K'c = \frac{K}{D}, \tag{1.1}$$

where K is constant of proportionality. As the grain boundary movement proceeds through the diffusion of atoms, that is the energy activated process, the constant K can be expressed through the temperature T and the activation energy Q as follows:

$$K = K_0 e^{-Q/RT}. \tag{1.2}$$

These statements allow to connect the modelling and physical parameters and calibrate the simulation results.

The more difficult aspect for investigation and control purposes is the abnormal grain growth (AGG), which is called also secondary recrystallization, which is related to the growth of the ordinary grains when they slowed-down because of one or several factors [6]: inclusions, locking points, preferred orientation or sheet thickness. There are two cases that arise when such a growth occurs [7]: 1) preferred orientation, when the mobility and the driving force of the low-angle boundaries are lower; 2) the presence of inclusions at the temperature which allows their slow dissolution. However, the nucleation stage is worth a closer investigation. There are four distinguished models of nucleation [8]: classical, martensitic, subgrain coarsening and bulging mechanism model. To date, the initial nucleation stage was considered in CA models using non-realistic approaches. However, the suitable choice for these models defines the future grain growth.

The interaction between two grains is governed by the differential equations which are introduced later, and it is possible to have analytical solution of the single grain growth kinetics. Unfortunately, there is no such analytical solution for the real structures where a big number of grains interact with each other. Therefore, cellular automata has been used for decades for the grain growth and recrystallization simulations. Cellular automata is a discrete model which allows to predict the behaviour of the system with the large number of components, such as grains.

The accuracy of the cellular automata, as the differential equations solver, is highly important. The common way to increase it is to validate its results via scaling solution to the one obtained in experiments. However, sometimes, this method works only on a small range of variables, which requires to revalidate results with new experiments. The reason of that that there is no linear relationship between iterations and physical time.

The novel CA technique developed in this research solves differential equations, that describe grain boundary movement in the recrystallization and grain growth processes. The technique has been designed from the principles of the time and space realism, which means that no further scaling to the experiments is required. A particular attention is given to the study of the errors that are produced with a developed CA technique.

1.2 Aims and Objectives

The aim of this research is to develop novel techniques for the simulation of the grain growth and recrystallization. For this purpose a single grain growth has been simulated and compared with the analytical solution of the differential equations. Simulations have been performed in MATLAB on Iceberg HPC Cluster and for visualisation ParaView has also been used. The efficiency of the performance has been estimated by finding errors and their convergence have been studied. Consistency of the model parameters has also been checked on the range of the cell sizes.

1.3 Outline of Thesis

Thesis is organised as follows. Chapter 2 introduces the reader with the history of applications of CA in material science, and briefly describes theories of recrystallization and cellular automata. It also describes the modification of the LTF that has been suggested for improvement of the shapes of the growing grains. CA is used for solving differential equations, describing grain bound-

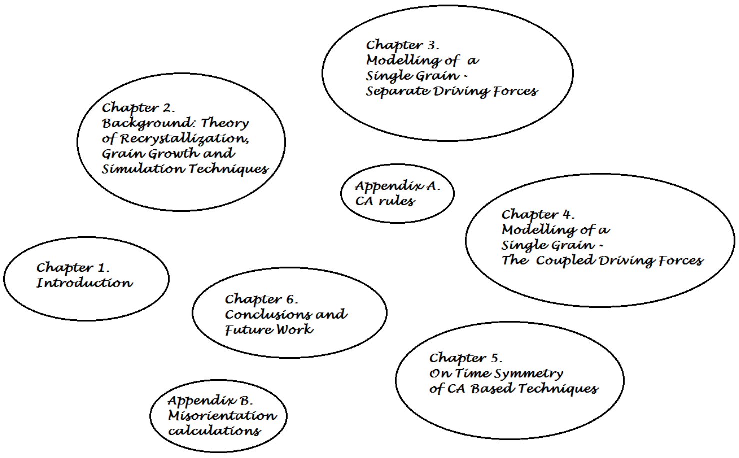


Figure 1.1: The outline of the thesis.

ary movement under driven forces acting separately in Chapter 3, and the coupled driven forces acting simultaneously in Chapter 4. Chapter 5 investigates the nature of the errors in simulations. It considers various methods to reduce them including corrected Moore neighbourhood and new random Margolus neighbourhood. Finally, Chapter 6 concludes the work in this thesis and suggests some future research directions. The explanation of the new CA developed technique is given in Appendix A. Appendix B includes the misorientation calculations which are required for multigrain calculations. The scheme of the thesis structure is shown in Fig. 1.1.

1.3.1 Summary of Contributions

The contributions of the research presented in this thesis are as follows:

1. Modified local transition functions have been suggested in order to improve the shape of the growing grains (Chapter 3);

2. A novel time-realistic deterministic Cellular Automata algorithm has been designed which is capable of considering the movement of cells in triple junctions (Chapter 3);
3. A developed CA technique alongside with its probabilistic version has been applied as a numerical solver for non-linear differential equations which describe the grain boundary movement (Chapter 3,4);
4. The errors of the simulations are extensively investigated and the nature of these are explained; The new random Margolus neighbourhood has been implemented to reduce errors caused by grid. (Chapter 5)

Chapter 2

Background: theory of recrystallization, grain growth and simulation techniques

2.1 Nucleation

Although nucleation is an important phenomenon since it leads to an initial size distribution for the following grain growth, it is very difficult to measure it experimentally, because it occurs very quickly and the structure obtained after nucleation disappears during further recrystallization and grain growth.

2.1.1 Mechanism of nucleation for dynamic recrystallization

The classical fluctuation theory of the nucleation does not work for recrystallization because of the low driving force and the high interface energy [9]. There are three recrystallization nucleation models, as described in [9]:

1. For the small strains and hot working the nucleation occurs due to the strain induced by the pre-existing grain boundary migration; this mechanism, which is also called a bulging, is mostly observed during the dynamic recrystallization (DRX);
2. The nucleation by the low angle boundary migration occurs at the high strains, high annealing temperatures, in the low stacking-fault energy (SFE) metals and alloys with the heterogeneous subgrain size distribution;
3. The nucleation by the subgrain coalescence, which occurs during static recrystallization (SRX), is possible in metals and alloys with the transition bands, high SFE, at the moderate strains, low temperatures and large spread in the distribution of subgrain angles. SRX with the nucleation by the subgrain coalescence has been simulated by Muramatsu [10] using the model of Kobayashi, Warren and Carter (KWC) phase field method.

The first mechanism, DRX, which is investigated in the present work, is the most suitable for the high temperature plastic deformation of metals with the low-to-medium stacking fault energy materials where recovery proceeds slowly during hot working (e.g. austenitic steels, γ -iron, copper, nickel, etc.) [4]. The main differences between DRX and SRX are as follows:

- The modelling of DRX requires a multiscale approach, the coupling of the microstructural evolution and macroscopic mechanical behaviour;
- The dislocation density in SRX recrystallized grains is almost zero, whereas it increases with the deformation during DRX;

- During DRX, the nucleation occurs only on the pre-existing grain boundary surfaces [11]. The nucleation rate laws have been derived by Cahn [12];
- The required dislocation density for the initiation dynamic recrystallization is much higher than for the SRX because of the driving force reduction.

2.1.2 Dislocation density evolution: The KM model

The dislocation density depends on the two processes, occurring simultaneously: 1) the plastic deformation, which has a hardening effect, generates dislocations; 2) the dynamic recovery annihilating dislocations that leads to the softening effect. Therefore, the change of the dislocation density with the true strain can be expressed as follows:

$$\frac{\partial \rho}{\partial \varepsilon} = \left(\frac{\partial \rho}{\partial \varepsilon}\right)_{hard} + \left(\frac{\partial \rho}{\partial \varepsilon}\right)_{soft} \tag{2.1}$$

Kocks and Mecking [13] proposed the following phenomenological equation for the variation of the mean dislocation density:

$$\frac{\partial \rho}{\partial \varepsilon} = k_1 \sqrt{\rho} - k_2 \rho, \tag{2.2}$$

where k_1 is a constant related to the hardening and $k_2(\varepsilon, T) = \alpha \mu b k_1 / \sigma_{st}$ is the softening parameter. This simplification in the literature is often called KM model.

The steady-state stress [4] is described by the following expression:

$$\sigma_{st} = (A_1 \dot{\varepsilon} \exp(Q_A/RT))^{1/A_2}, \tag{2.3}$$

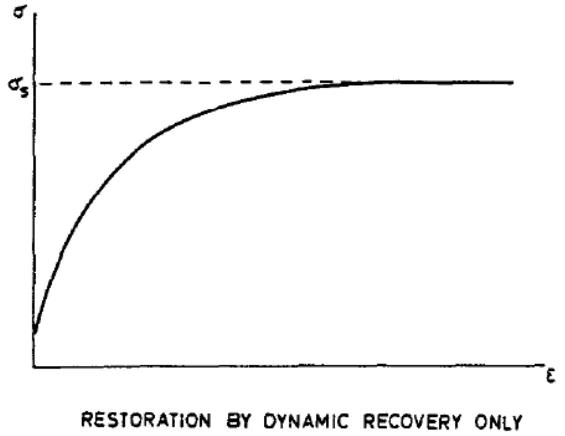


Figure 2.1: Stress-strain curves for dynamic recovery [11].

where Q_A is the activation energy.

The high-temperature flow stress σ is related to the average dislocation density $\bar{\rho}$ [4] as follows:

$$\sigma = \alpha\mu b\sqrt{\bar{\rho}}, \tag{2.4}$$

$$\bar{\rho} = \frac{1}{n}\sum_{i=1}^{n}\rho_i, \tag{2.5}$$

where α is a dislocation interaction coefficient and for most metals approximately equals 0.5 [4], b is the magnitude of the Burgers vector, μ is the shear modulus, n is the total number of cells. The combination of these two equations, Eq.(2.4) and Eq.(2.5), leads to a macroscopic stress-strain curve as shown in Fig. 2.1.

2.1.3 Critical dislocation density

The critical dislocation density for the initiation of DRX during hot working was proposed by Roberts and Ahlblom [11] by neglecting dynamic recovery:

$$\rho_c = \left(\frac{20\gamma\dot{\varepsilon}}{3\mathbf{b}lM\tau^2}\right)^{1/3}, \tag{2.6}$$

where γ is the grain boundary energy, $\dot{\varepsilon}$ is the strain rate, $\mathbf{b}$ is the Burgers vector of glide dislocations, l is the size of subgrains developed during deformation or dislocation mean free path [14] and can be expressed as follows [15]:

$$l = \frac{K\mu b}{\sigma}, \tag{2.7}$$

where constant K is about 10 for the most metals. Eq.(2.4) for the flow stress σ gives $l = \frac{K}{\alpha\sqrt{\rho_{ini}}} = \frac{10}{0.5\sqrt{\rho_{ini}}}$, M is the boundary mobility, τ is the dislocation line energy being as follows [11]

$$\tau = \mu b^2/2, \tag{2.8}$$

μ is the shear modulus and the critical nucleus size being as follows [11]:

$$d_c = 2r_c = \frac{6blm\tau}{5\dot{\varepsilon}}\rho_c^2 \tag{2.9}$$

The kinetics of the grain boundary nucleation was derived by Cahn as follows: [12]:

$$X = 1 - \exp[-b_s^{-1/3} \cdot f(a_s)], \tag{2.10}$$

where X is the volume fraction transformed in time t,

$$f(a_s) = a_s \int_0^1 1 - \exp\left\{-\pi a_s^3 \left[\frac{1-x^3}{3} - x^2(1-x)\right]\right\} dx, \tag{2.11}$$

where $x = \frac{y}{Gt}$, G is the growth rate, y is distance,

$$a_s = (I_s G^2)^{1/3} t,$$
$$b_s = \frac{I_s}{8S^3G},$$

where I_s is the specific grain boundary nucleation frequency per unit area and S is the grain surface area per unit volume.

The time required for site saturation is $t_{ss} = (I_s G^2)^{-1/3}$ and the estimated dynamically recrystallized grain size can be written as follows [11]:

$$D = d_c + 2Gt_{ss} = d_c + 2Y m\tau\rho_{0c}t_{ss}, \tag{2.12}$$

where Y is within 1.05-1.25 depending on Zener-Holomon parameter, $Z = \dot{\varepsilon}\exp\left(\frac{Q}{RT}\right)$, where Q is the activation energy $Q = -R\left[\partial \ln \dot{\varepsilon}/\partial(1/T)\right]_{\sigma_{max}}$.

Takaki et al. [16], [17] developed a multi-phase-field (MPF) model firstly proposed by Steinbach and Pezzolla [18] to simulate microstructural evolution during DRX. They used a computational algorithm as suggested by Kim et al. [19] and Kocks-Mecking model [13] for the modelling of the dislocation density evolution. The coupling of CA method with the KM model was performed earlier by Ding and Gao [14].

2.1.4 Nucleation rate

When the dislocation density reaches the critical value $\rho_c = \left(\frac{20\gamma\dot{\varepsilon}}{3blm\tau^2}\right)^{1/3}$, the nuclei are placed on the grain boundaries (or internal defects, e.g. impurities, shear bands).

The nucleation rate $\dot{n}$ per unit area of a grain boundary can be written as a function of the temperature T and the strain rate $\dot{\varepsilon}$ [14] as follows:

$$\dot{n}(\dot{\varepsilon}, T) = C\dot{\varepsilon}^d \exp\left(-\frac{Q_A}{RT}\right), \tag{2.13}$$

where C is constant, d was set as 1 in [14, 16, 17] and Q_A is the activation energy.

2.2 Recrystallization

Recrystallization can be defined as a volume stored energy driven process relating to the formation of the new grain structure in the plastically deformed materials [20]. The two associated types exist: 1) static recrystallization, which occurs after deformation; 2) dynamic recrystallization, where the new grains are nucleated and grow during deformation. In addition, the latter can be also divided into two groups: discontinuous (DDRX) and continuous (CDRX) dynamic recrystallization. During the first one, the nucleation of the new grains occurs at high-energy sites in the microstructure, primarily along the boundaries. The second is accompanied by the formation of the long-angle boundaries from the dislocation networks.

The dynamic recrystallization occurs when the dislocation density reaches a critical value. At the lower process temperatures T or high strain rates $\dot{\epsilon}$ the flow stress oscillates forming serrations, and the higher T or lower $\dot{\epsilon}$ lead to these smoothing of the oscillations (see Fig. 2.2). The modelling of recrystallization allows to optimise these parameters that helps to create a material with improved properties.

The annealing of the deformed metal occurs in three stages as was defined by Turnbull [21]: 1) recovery, characterized by continues changes in

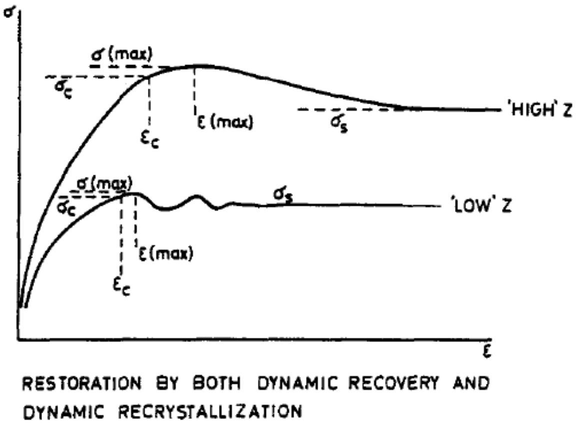


Figure 2.2: Stress-strain curves for dynamic recrystallization [11].

physical properties, 2) recrystallization, which implies nucleation and nuclei growth, and 3) grain growth, occurring by the migration of the grain boundaries, which existed before annealing. Metals, which have already undergone recrystallization may incur secondary recrystallization or abnormal grain growth (AGG).

2.2.1 Mean field theories

Several theories were introduced to describe the change of the grain size distribution $f(R)$ with time t, which considered the average effect of the whole ensemble of the grains on the embedded isolated grain and were classified by Atkinson [22] as mean field theories.

It is convenient to consider them from the viewpoint of Hunderi and Ryum [23]. They considered the normal grain growth as the motion of the individual grains and they demonstrated that the change of the grain size $\frac{\partial R}{\partial t}$ accompanied by the grain boundary motion can be caused by two processes: 1) a diffusion-like process characterized by diffusion constant $Diff$ depended on the specific grain-boundary mobility, 2) by velocity $v = dR/dt$ which is a drift velocity induced by the driving force F. Therefore the continuity equation for the grain flux j can be written as follows:

$$j = -Diff \frac{\partial f(R,t)}{\partial R} + vf(R,t), \tag{2.14}$$

where the distribution function $f(R,t)$ is defined as follows:

$$\frac{\partial f(R,t)}{\partial t} = \frac{\partial}{\partial R}(-j) = \frac{\partial}{\partial R}\left(Diff \frac{\partial f(R,t)}{\partial R}\right) - \frac{\partial}{\partial R}(vf(R,t)) \tag{2.15}$$

Hence, the mean field theories can be divided into groups the theories concentrating on:

1. the drift-velocity terms [24],[25]: $Diff = 0, \dfrac{\partial f}{\partial t} + \dfrac{\partial}{\partial R}(fv) = 0$;
2. diffusion, Louat [26]: $v = 0, \dfrac{\partial f}{\partial t} = Diff \dfrac{\partial^2 f}{\partial R^2}$

The first type methods are, in turn, divided into two approaches developed by: 1) Feltham [25], who obtained the drift velocity v using the experimentally determined form of the grain size distribution f (he asserted it as log-normal) and 2) Hilert [24], who suggested the particular expression for v and then obtained f. These methods are described next in more detail.

2.2.1.1 ***Feltham***

Feltham [25] neglected the diffusion term and by using the time invariant properties of the grain ensemble distribution, he first obtained the drift velocity dependence by claiming that the experimental grain size distribution $f(R,t)$ is log-normal: $\dfrac{\partial D_i^2}{\partial t} = C \ln \dfrac{D_i}{D'}$, where C is a constant, D_i is the diameter of ith grain, D' is the most probable spatial grain diameter. Assuming

$D = D_{max} = 2.5D'$ or time-invariance of the log-normal distribution Feltham obtained a parabolic grain growth law $D'^2 - D_0'^2 = \left(\frac{\lambda V a \sigma}{h}\right) e^{-H/Kt}$ for pure metals, where t is the time of isothermal annealing, λ is a numerical constant of order unity, a is a jump distance, σ is a specific surface energy, h is a Plank constant, and H is the activation energy for grain-boundary self-diffusion.

However, as was noticed by Srolovitz [27], the real F distribution was skewed and has upper cut-off in contrast to the symmetric log-normal function, whose tails approach infinity, and whose peaks of log-normal curve are wider than in their simulation.

2.2.1.2 *Hillert*

Another approach was realized by Hillert [24], where he suggested a particular expression for a drift velocity as follows:

$$v = g\frac{dR}{dt} = \alpha M \sigma \left(\frac{1}{R_{cr}} - \frac{1}{R}\right), \tag{2.16}$$

where α is dimensionless constant, M is the boundary mobility and σ – the surface energy of the boundary, R_{cr} is the critical size at which switch from the shrinkage to the growth occurs and proportional to the mean radius $\overline{R}$. Using this expression, he derived a steady state solution distribution function $f(R,t)$. However, as was noted by Hunderi [23], this distribution is much more tapering than experimentally obtained and also it is difficult to check experimentally this suggested relationship between drift velocity and grain size. The more empirical relations were reviewed by Gao [28].

Hillert also suggested a defect model in accordance with the grain growth occurring by the dislocation climb and obtained a parabolic growth law calculated from the number of the defects per grain c as follows:

$$\frac{dR_{cr}^2}{dt} = \frac{1}{3} M \sigma \sum_p p \cdot c_p, \tag{2.17}$$

where c_p is the concentration of grains with $n = 6 - p$ neighbours and $\sum_p p \cdot c_p$ is the total concentration of defects. Hillert's model considers change of the topological parameters only in the nearest neighbour cells. As the number of defects is constant, therefore their concentration will grow during grain growth approaching a high value required for the normal grain growth. Hillert showed that this concentration should be equal $\frac{3}{4}$. However there maybe annihilation of the defects, which will decrease number of them. This should lead to a constant remaining number of the defects during the grain growth.

Hillert's model was extended to textured microstructure, by introducing the dependence of R_{cr} on the orientation by Abbruzzese and Lücke in [29]. These texture changes are rather rules than exceptions, because the primary recrystallization almost leads to the preferred orientations. Abbruzzese and Lücke define normal grain growth as continuous and abnormal (secondary recrystallization) as discontinuous, at the same time noting that they are not principally different processes, but are distinguished only by the values of the input parameters, such as diffusivities, fractions of grains with different orientations, mean radii and the standard deviations of the grain size distribution for orientation H. They developed a statistical model by making assumptions on the superposition of grain curvatures, homogeneity and of random grain distribution. In contrast to the textureless case, they derived a different radius $\tilde{R}_c^H$ and the diffusivity M^H, for each orientation class. Their model predicted deviations from the parabolic grain growth law, and a stepwise growth is observed, that corresponds to the different time dependencies at the different time stages of grain growth.

2.2.1.3 *Louat*

An alternative method was to consider a boundary motion as a diffusion process, excluding drift velocity process, which was done by Louat [26] as follows:

$$\frac{\partial f}{\partial t} = Diff \frac{\partial^2 f}{\partial R^2}. \tag{2.18}$$

Louat made the assumption that the grain growth is a random process, statistically determined. Assuming that the diffusion coefficient $Diff$ is independent of t, f, R, and given the boundary condition $f(0) = 0$ he obtained the following solution:

$$f(R,t) = \frac{CRe^{-R^2/4Difft}}{Difft^{3/2}}, \tag{2.19}$$

where C is a constant chosen to give distribution at time $t = t_0$.

2.2.1.4 *Hunderi and Ryum*

Hunderi and Ryum [23] suggested a generalized approach to find either v or f, where by using appropriate scaling for R, v and f, the Hillert distribution can hence be obtained. They also found that the drift velocity derived by Feltham is first term in a series expansion.

2.2.2 Driving forces for recrystallization

During recrystallization and grain growth, grain boundary is moving with velocity as follows:

$$v = Mp, \tag{2.20}$$

where M – grain boundary mobility and p – driving pressure.

There main source of driving forces for grain boundary migration during primary recrystallization is the stored deformation energy (10 $[MPa]$), which is caused by the dislocations motion. The driving force for grain growth is the grain boundary energy (10^{12} $[MPa]$). However, there are also other driving forces with lower magnitude such as surface energy ($2 \cdot 10^{-2}$ $[MPa]$), chemical driving force ($6 \cdot 10^{2}$ $[MPa]$), magnetic field ($3 \cdot 10^{-5}$ $[MPa]$), elastic energy ($2.5 \cdot 10^{-4}$ $[MPa]$) and temperature gradient ($4 \cdot 10^{-5}$ $[MPa]$) [30] but their estimated magnitude are significantly lower than for recrystallization and grain growth and therefore they were not considered in the present work. Therefore, only the pressures arising from curvature given by Gibbs-Thomson relationship [4] $p_C = -2\gamma/r$ and dislocation density difference $p_D = \tau[\rho]$, were considered (the pinning pressure was no considered yet) in the present research. Here, the dislocation line energy $\tau = \alpha\mu b^2$, α is a constant of the order of 0.5, b is a magnitude of Burgers vector, μ - shear modulus and jump in dislocation density across the boundary $[\rho]$ is the dislocation-density difference between grains $\delta\rho = \rho_i - \rho_j$,

$$p = p_C + p_D \tag{2.21}$$

2.2.3 Mobility and boundary migration

The grain boundary movement following Cole, Feltham and Gilliam [31] can be considered as transferring atoms from one grain to the adjoining by diffusion jump across the boundary. In the presence of the driving force the activation energy for migration or boundary self-diffusion Q is decreased in one direction and increased in another. Therefore, a net flow of atoms can be written as follows [30]:

$$v = bv_0 c_{vg} \left\{ \exp\left(\frac{-G_m}{kT}\right) - \exp\left(-\frac{G_m + pb^3}{kT}\right) \right\}, \tag{2.22}$$

As $pb^3 \ll kT$,

$$v \cong bv_0 c_{vg} \exp\left(-\frac{G_m}{kT}\right) \left\{1 - 1 + \frac{pb^3}{kT}\right\} = b^4 v_0 c_{vg} \frac{1}{kT} \exp\left(-\frac{G_m}{kT}\right) \cdot p = mp, \tag{2.23}$$

where G_m is free activation energy, c_{vg} is vacancy concentration, b is the distance by which atom moves the boundary (about atomic diameter), v_0 is atomic vibration frequency [31, 32, 6, 30].

Taking into account the Nersnt-Einstein relationship and Arrhenius type dependence on temperature of diffusion in solids the mobility can be written as follows [32, 30]:

$$m = \frac{b^2 D_m}{kT} = \frac{b^2 D_0}{kT} \exp\left(-\frac{Q_m}{kt}\right), \tag{2.24}$$

or if Q_m is activation energy of grain boundary diffusion per mol

$$m = \frac{b^2 D_0}{kT} \exp\left(-\frac{Q_m}{RT}\right) = m_0 \exp\left(-\frac{Q_m}{RT}\right), \tag{2.25}$$

in this expression D_0 is the pre-exponential of lattice diffusion and has dimension of $[m^2 s^{-1}]$. It was used in the works of Davies [33], Lan [34], Zheng [35, 36], Han [37]. Other authors such as Ding [14, 38], Kugler [39] used the expression by setting $\delta D_{ob} = bD_0$

$$M = \frac{\delta D_b}{kT} b,$$

$$\delta D_b = \delta D_{ob} \exp\left(-\frac{Q_b}{RT}\right), \tag{2.26}$$

$$M = \frac{b\delta D_{ob}}{kT} \exp\left(-\frac{Q_b}{RT}\right), \tag{2.27}$$

here, δ is characteristic grain boundary thickness, D_b is boundary self-diffusion coefficient, and δD_b has dimension of $[m^3 s^{-1}]$.

The orientation dependent expression for mobility was firstly introduced by Humphreys [40] and used by Hallberg [41], Han and co-workers [37] as follows:

$$m(T, \Theta) = m_0(T)\left(1 - \exp\left[-B\left(\frac{\Theta}{\Theta_m}\right)^k\right]\right), \tag{2.28}$$

where $m_0(T)$ is the mobility at high angle boundary described above, $k = 4$, $B = 5$.

The shear modulus has a temperature dependence, which was used by Hallberg[42, 41] based on the relationship given by Frost and Ashby in [43], for example:

$$\mu(T) = \mu_0\left[1 + \left(\frac{T - 300}{T_m}\frac{T_m}{\mu_0}\frac{d\mu}{dT}\right)\right], \tag{2.29}$$

where μ_0 is shear modulus at 300 K and $\frac{T_M}{\mu_0}\frac{d\mu}{dT}$ is the temperature dependence of modulus and T_m is a melting temperature.

2.2.4 Primary recrystallization kinetics

The phenomenological model known as The Johnson-Mehl-Avrami-Kolnogorov (JMAK) [44, 45, 46] equation is often used to quantitatively describe recrystallized fraction X as a function of time t:

$$X = 1 - \exp\left\{ - \left(\frac{t}{t_R} \right)^q \right\}, \tag{2.30}$$

where t_R is the characteristic time for recrystallization.

2.3 Grain growth following recrystallization

2.3.1 The Turnbull rate equation

Turnbull's work [21] was devoted to the study of the underlying physical principles of the recrystallization and grain growth. They formulated a phenomenological rate equation, describing grain-boundary segment motion under the influence of the exerted by surface curvature on isolated section boundary pressure p or driving force, which was considered as the reduction in the energy and therefore a decrease in the grain boundary.

This pressure p is related to the grain boundary velocity v via the following expression: $v = Mp$, where the grain boundary mobility M varies with the temperature T according to: $M = M_0 \exp\left(-\frac{Q_a}{RT}\right)$. Here, Q_a is the activation energy required for the boundary movement, R – ideal gas constant, T – absolute temperature and $M_0 = \dfrac{\overline{b}^2 D_0}{KT}$ – pre-exponential factor, $\overline{b}$ – value of Burgers vector, D_0 – diffusion constant, k – Boltzmann constant. The pressure p can be expressed through the boundary curvature κ as $p = \gamma\kappa$, where $\kappa = \dfrac{1}{r_1} + \dfrac{1}{r_2}$, r_1 and r_2 are the principle radii of the curvature and supposing that the boundary is the sector of sphere radius of which is $R = r_1 = r_2$ then $p = \dfrac{2\gamma}{R}$. Up to misorientation of 15^o the grain-boundary energy is γ_m and at the misorientation $\Theta_m = 15^o$, corresponding to high angle grain boundary – the grain boundary free energy γ can be obtained through the Read-Shockley equation [47]:

$$\gamma = \gamma_m \left(\frac{\Theta}{\Theta_m}\right)\left[1 - ln\left(\frac{\Theta}{\Theta_m}\right)\right], \tag{2.31}$$

where Θ is the grain boundary misorientation, beyond $\Theta = 15^o$ the grain boundary energy is assumed constant.

For the isothermal conditions linear rate of growth or grain boundary velocity $\dot{x}$ can be simplistically expressed as $\dot{x} = \dfrac{K'\sigma V}{R}$, where σ is the surface energy of the boundary, V is the grain atomic volume, rate constant $K' = K'_0 \exp\left(-Q_G/RT\right)$, where Q_G is the activation energy for grain growth [21]. In the case of the isothermal annealing condition the following statements are assumed: 1) boundary surface tension σ, and therefore the grain boundary free energy γ is independent of the grain size and time, 2) boundary curvature r is proportional to the mean diameter D or radius R of an individual grain, and so 3) the net increase $\dfrac{dR}{dt}$ is proportional to $\dot{x}$: $\dfrac{dR}{dt} \sim \dfrac{K'\sigma V}{R}$, therefore one can derive the parabolic growth law for the isolated section of the grain boundary

$$\overline{R}_t^2 - \overline{R}_0^2 = kt, \tag{2.32}$$

where $\overline{R_0}$ is the initial mean radius, $\overline{R}_t$ is the mean grain radius at the time t and k is constant. In the limit $\overline{R}_t \gg \overline{R}_0$, $\overline{R}_t^2 = kt$ and $\overline{R}_t = kt^n$, here $n = 0.5$ is also termed as grain growth exponent. However, experiments usually show it to be less than 0.5 [48].

The reason for the discrepancy is that their equation describes the kinetics of only individual grains, but does not consider the influence of neighbouring grains, the topological requirements, which were studied by **Smith** [49], although he also ignored kinetics or how quickly topological transformations and overall grain growth can occur. Therefore, the grain growth should be described in terms of the interaction between topological space-filling and force driving boundary migration. This attempt was made by **Rhines and Craig** [50], who included two new parameters: the *sweep constant* Θ which corresponds to the number of grains vanishing when boundaries sweep through the unit volume and the *structural gradient* $\sigma = \dfrac{M_V S_V}{N_V}$, where N_V is the number of grains per unit volume, M_V is the total curvature per cm^3, S_V – the total area of grain boundary per cm^3. They obtained the grain growth exponent of 1/3. The average grain volume increases with the rate proportional to the mobility μ, the grain boundary surface tension γ and the structural gradient σ:

$$\overline{V} = \frac{\Theta^* \mu \gamma M_V}{N_V} t + \overline{V_0}, \tag{2.33}$$

where $\overline{V_0}$ is the mean grain volume at the time $t = 0$.

A review of theories for the normal grain growth in single phase materials, which attempted to merge these two above approaches together, is represented by Atkinson [22]. Atkinson emphasizes that the normal grain growth which is characterized by the time-invariant size distribution, its range being relatively narrow.

2.4 Theory of Cellular Automata

2.4.1 Definition of CA

Cellular automata (CA) is a discrete (in space and time) model consisting of a regular lattice, each site of which can take one of a finite set of possible discrete states. These states are simultaneously upgraded according to local rules, which define the state variable of each site as a function of the state of the neighbouring sites (local rules) and its previous state. This technique allows to analyse and simulate the most sophisticated behaviour of nonlinear dynamical systems, for example evolution of hydrodynamic flows [51].

S. Wolfram [52] defined the elementary cellular automaton as a one-dimensional CA, each state of which takes one of two possible states and have two adjacent neighbours (on the right and left). The rules are usually represented as a decimal number by reading its rule icon in base 2. Examples of some rules are shown in Fig.2.3.

The configuration of neighbouring influences on the final state because of the discretization of space. There are two types of neighbouring: type I – neighbours which one coordinate differs by one unit and type II – none of its coordinates differ by more than one unit. Conway's game of life is the example of type two-dimensional CA and belongs to the "totalistic" rules. The value of the cell in totalistic CA depends on the sum of the values of its neighbours at the last step [52].

Three sites of elementary CA (left, central and right) can be in $2^3 = 8$ different configurations and there are $2^8 = 256$ different rules, determining the next state of the site. Usually, two requirements are used, that give just 32 possible rules called "legal": 1) rules must be symmetric (100=001, 110=011); 2) initial zero state should be unchanged. This gives general form of rules: $a_1 a_2 a_3 a_4 a_2 a_5 a_4 0$ [53]. Cellular automata increases complexity with higher dimensions.

In addition, two broad classes of CA may be identified : 1) those which lead to simple, homogeneous final configurations, 2) CA is also able to form fairly complicated structures despite simple initial states. There are different types of CA's, such as the PCA (Probabilistic or Stochastic CA), the LGCA (Lattice Gas Cellular Automats) and the DSMC (Direct Simulation Monte Carlo),

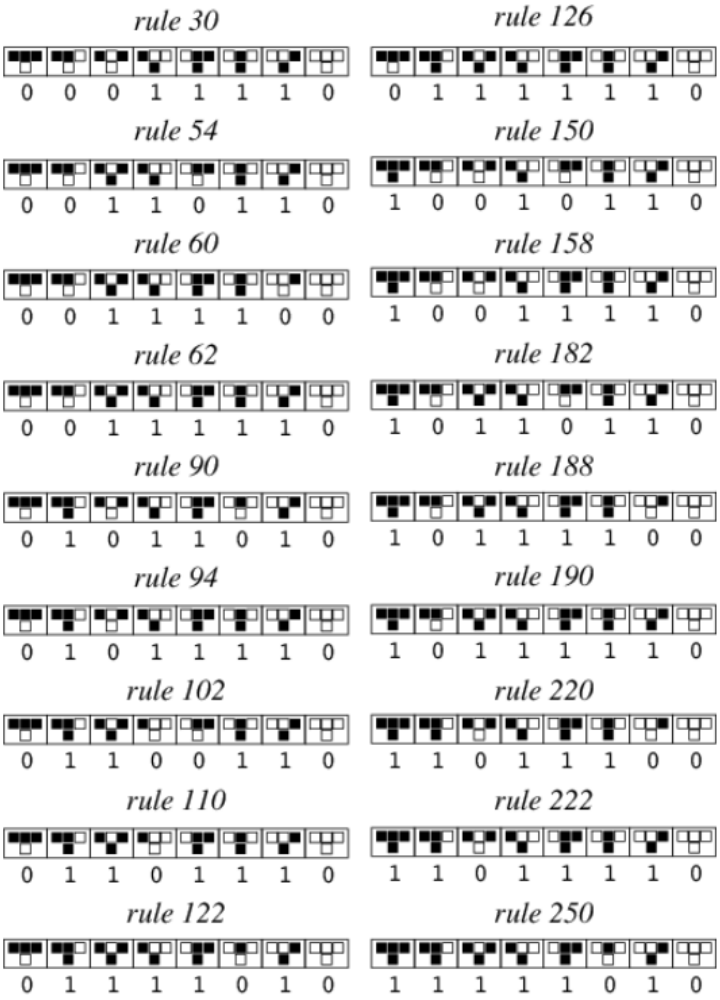


Figure 2.3: Elementary CA rule icons.

short overview of which is given in [54] with an example of the simulation of recrystallization. Vanag [55] compared these methods on the example of the Williamowski-Rossler model.

2.4.2 Properties of CA

In contrast to the Monte Carlo model, that reflects real microscopic models, cellular automato-based simulations are *scalable* to arbitrary systems, but only by achieving an appropriate scaling. The CA approach is *flexible* due to the fact that automata can involve any relevant rules of transformation. These properties make CA *versatile* for the various space and time scales in the simulation of microstructure. Depending on the type of rules, cellular automata can be **deterministic** or **probabilistic**. It must be noted at this stage that there are two different ways to make a cellular automata non-deterministic. In the first approach lattice cells are selected randomly, but deterministic transformation rules are used. In the other method, the probabilistic transformation is applied, but with systematic analysis of all lattice positions in the sequential order.

In [52] details about the behaviour of elementary CA with analyses of some of their local statistical properties, such as density and two-point correlation function, are given. It is shown that these characteristics cannot sense the structures obtained via a complex CA. The **sequence density** $Q_i(n)$ – density of sequences consists of n adjacent sites with value i. These sequence densities, normally obtained from configurations after the application of local cellular automata rules, differ from the sequence density of its initial disordered configuration. This is an exhibition of *self-organization*. Another reflection of self-organization is the difference in the probabilities for initially equiprobable configurations. In addition, global properties (statistical properties not of the set of cells in individual CA configuration, but ensemble CA) such as **Hamming distance** $H(s_1, s_2)$ are considered. The Hamming distance indicates the number of sites which are different between the binary sequences of s_1 and s_2. For additive rules, the behaviour of function is defined by the evolution of the initial difference for τ time steps. In contrast, for nonadditive any small change in initial steps linearly increases the number of the uncorrelated sites.

The most of the elementary CA are *irreversible,* since the final configuration can be obtained from several different initial configurations. There are 6 CA rules among Wolfram's elementary rules which are reversible: rule 204 (identical), rule 51 (complement), rule 170 (shift-left), rule 240 (shift-right) and their invert analogues rule 85 and rule 15. Fig. 2.4 shows the evolution of a single seed and random initial configuration according to these rules. Rules 105 and 150 become reversible when periodic boundary are used.

The extension to k possible states per site and d dimensions is also considered [52]. The time and length scales of appearing self-similarity are increasing with k, and in the limit $k \to \infty$ self-similarity may not be obvious at any finite time. These CA can form an "impermeable membrane" protecting the sites from external fluctuations. In the case of a five-site neighbourhood (the site itself, its nearest neighbours and its next-nearest neighbours) self-similar patterns become less apparent than in three-site neighbourhood for disordered initial states. CA can be considered as computers, where input data represent the initial states. Complicated CA which can simulate any computable function are called **universal**. Such CA are apparently capable of computing any physical system.

Discussed global properties determined in algebraic properties of the polynomials representing the additive CA time evolution rule are discussed in

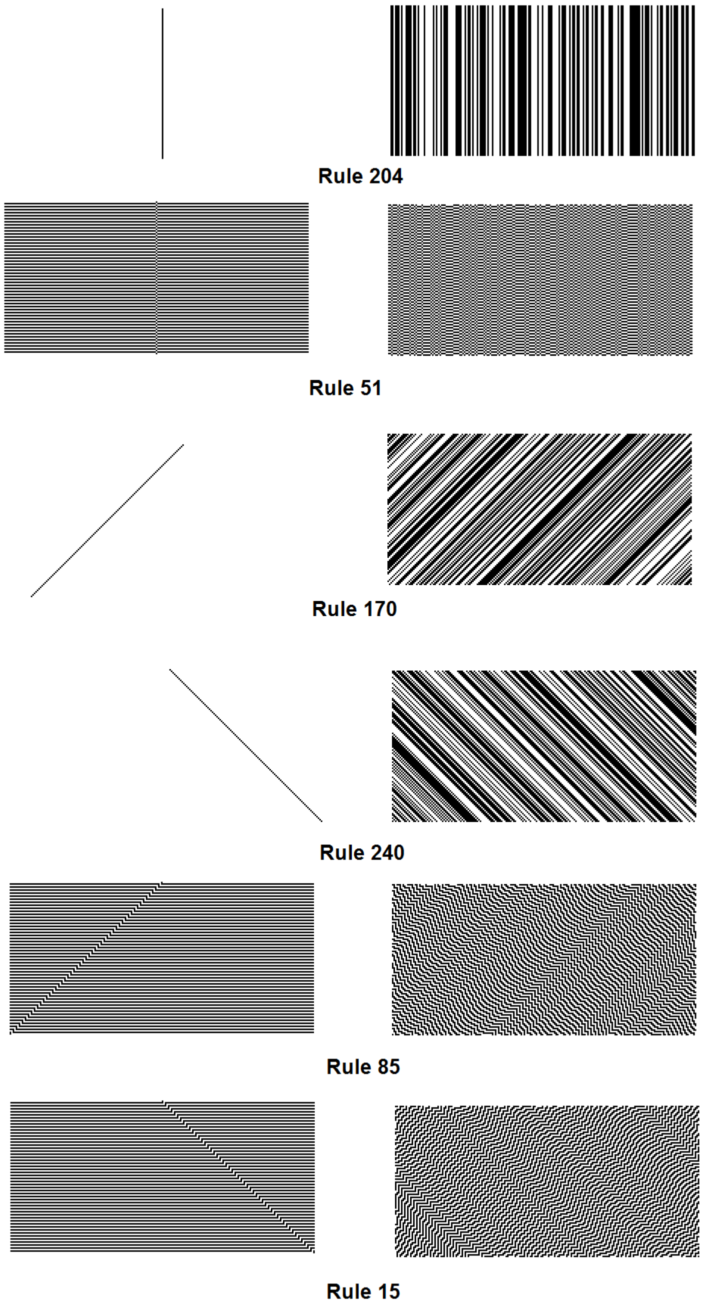


Figure 2.4: First 100 steps of evolution a a single seed and random initial structure according to reversible elementary CA rules.

[56]. It is shown on the example of the rule "modulo-two" that for the additive CA global properties depend not only on the value of size N, but also on its number theoretical properties of N. For example, irregular dependence on N of cycle lengths has been found. The maximum possible length of cycle, which grows on average exponentially with the size N of CA, is considered to be of the order of the square root of the entire number of possible configurations, hence, there is the possibility of rather long cycles. State transition can be represented by graphs. As shown by the diagram in Fig. 2.5 the nodes joined by directed arcs represent possible configurations of states. The trees representation (irreversible transactions) will have their roots in the cycles of the state configuration which maybe visited repeatedly. Irreversibility is exhibited by merging distinct branches in the trees, which is accompanied by a decrease in the entropy; irreversibility can be measured by the fraction of configurations which may be reached after one time step. Such cycles represent "*attractors*", on which characterization based classification is defined by dynamical systems theory as denoted in [57]. The exception is the "reducible" CA, which generate patterns, containing properties of several classes. Some quantitative characteristics such as propagation speed λ, spatial set and measure (in terms of probabilities) entropies, $s^{(x)}(X)$ and $s_{\mu}^{(x)}(X)$, are introduced respectively, including the corresponding dimensions $d^{(x)}$ and $d_{\mu}^{(x)}$.

2.4.3 Classification of CA

There are four basic qualitative classes of CA established during statistical studies. In **Class I** the outcome of the evolution is an homogeneous state, so it is independent of the initial state and prediction is trivial. Its attractor can be considered as the "limit points" observed in the phase space. Therefore,

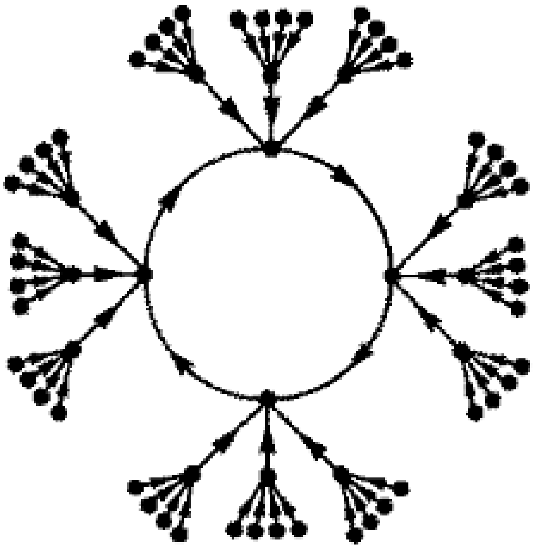


Figure 2.5: State transition diagram [56].

its dimensions are zero for such attractors. In **Class II** the final states are stable or of periodic persistence structures. For Class II CA maximum average propagation speed $\overline{\lambda}_{+}$ and thus the temporal dimension vanish. **Class III** CA evaluates to chaotic patterns and any changes in the initial states tend to propagate with a finite speed. For those CA which generate more regular patterns the length of transients are shorter and the fractions of states on cycles are much smaller than those which generate more irregular patterns. Class III CA exhibit decreasing entropy with time, which the indicates irreversibility. Infinite CA **Class IV** are found to be probably capable of universal computation, namely they are capable of evaluating any computable function, and a finite CA is only a subset of these functions. The result of the evaluation of CA from this class cannot be predicted and its behaviour may be determined only by explicitly running it. Sets of the configurations of the first two classes can be considered as sentences in regular grammar. CA are considered from this point of view in [53], where each word of the language is represented by a CA configuration. CA are represented by state transition graphs, where nodes represent pairs of initial states. The total numbers of states $\Xi^{(t)}$ (size) in the minimal DFA (deterministic finite automaton) gives a "measure of the complexity".

A computation theory features of the above classes is proposed. The limit sets for classes I and II CA lead to regular languages with a constant complexity or increasing as polynomials in t. For classes III and IV one can observe rapidly-increasing complexities, so the limit sets do not form a regular language: limits sets for Class III CA correspond to languages called "context-sensitive", for IV – to "general". "Regular" languages are fairly elementary, so their characterizations may be computable via finite calculating processes. In the case of the limiting properties determined by an infinite computational procedures, there are possible "short cuts" in order to obtain finite description of the limiting properties. In contrast, for more complicated languages the behaviour can be defined only via explicit simulation. This fact leads to the non-computability and undecidability, which are suggested to be the common feature for the systems investigated in mathematics, computation theory and theoretical physics [58].

To evaluate real physical systems, such as dendritic crystal growth or reaction-diffusion process, it is convenient to use two-dimensional CA , whose global properties are very close to those of one-dimensional CA [59]. Two-dimensional CA in most features are similar to one-dimensional CA, for example they can be considered in the classifications mentioned above, however, they have some new properties, for example forming dendritic and corrugated boundaries, which are observed in natural systems.

To characterize limiting structures of patterns a *growth dimension* is hereby introduced, which is a noninteger for dendritic patterns, for example spatial $D_x = \lim_{t\to\infty} \frac{\log n}{\log t}$, where n is the total number of sites, which are arranged in the patterns, and $\overline{D_x} = \lim_{t\to\infty} \frac{\log \overline{n}}{\log t}$ for the boundaries, where $\overline{n}$ is the number of sites situated on the boundary. Boundary is defined as the set of the sites, which can be obtained via the arbitrary path started from infinity and which does not intersect any nonzero sites. These properties with some exceptions do not depend on the form of the initial states for the resulting patterns, obtained in the large times.

CA of Class III or IV may correspond to the physical process that seems random or chaotic. These CA evolving from the random initial state exhibit random behaviour. Such a type of system is called homoplectic. They also can generates randomness even from a simple initial state, and these type of system is called autoplectic [60].

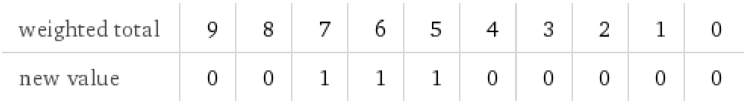
2.4.4 2D CA

One of famous examples of two dimensional cellular automata is Conway's Game of Life. It is an outer totalistic CA, where cell change its state as follows depending on the sum of values of neighbours and on the state itself as follows:

$$a_{i,j}^{(t+1)} = \tilde{f}(a_{i,j}^{(t)}, a_{i,j+1}^{(t)} + a_{i+1,j}^{(t)} + a_{i,j-1}^{(t)} + a_{i-1,j}^{(t)}) \tag{2.34}$$

with a rule specification by a code as follows:

$$C = \sum_n f[n]k^{kn} \tag{2.35}$$



weighted total	9	8	7	6	5	4	3	2	1	0
new value	0	0	1	1	1	0	0	0	0	0

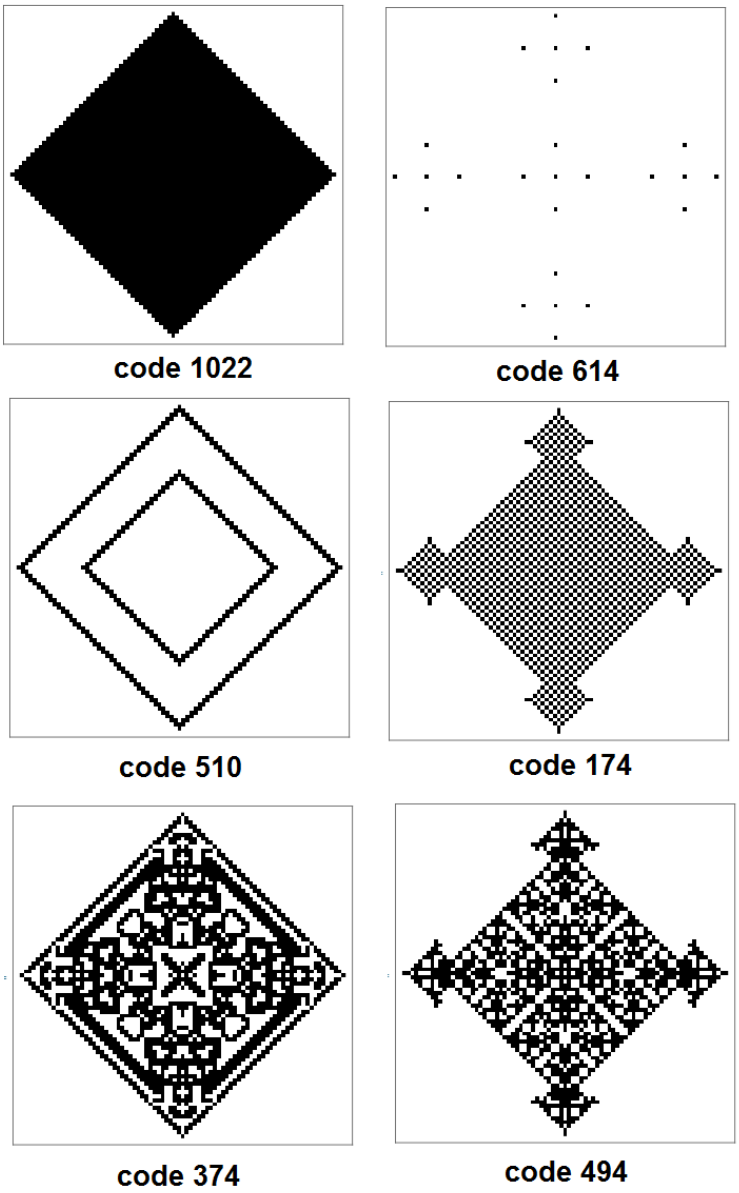
Figure 2.6: Rule table for Game of Life.

Rule table for Game of Life is shown in Fig. 2.6. According to Eq.(2.35) the code rule is $C = 2^7 + 2^6 + 2^5 = 128 + 64 + 32 = 224$.

Some examples of structures generated by evolution of 2D outer totalistic CA from a single grain are shown in Fig 2.7.

As has been shown by Wolfram and Packard [59], 2D CA global properties are similar to 1D CA. However some phenomena such as boundaries and interfaces have no analogue in one dimension. An article [59] contains an extensive study on two-dimensional CA by direct simulation of randomly sampled CA rules. An interesting fact has been observed, some rules lead to the diffusive growth of polygons or almost circular shapes, despite of regular lattice. Some examples of such structure are shown in Fig. 2.8. Examples of 2D patterns in this section were generated using Wolfram Language.

For 2D CA Neumann neighbourhood (Fig. 2.9 a) consists of 5 cells and Moore neighbourhood (Fig. 2.9 b) contains 9 cells. Therefore, there $2^5 = 32$ and $2^9 = 512$ different configurations which means 2^{32} and 2^{512} possible rules for Neumann and Moore neighbourhoods respectively. And 3D CA will increase computational costs drastically.



code 1022

code 614

code 510

code 174

code 374

code 494

Figure 2.7: Structures obtained by evolution of 2D outer totalistic CA after 40 iterations.

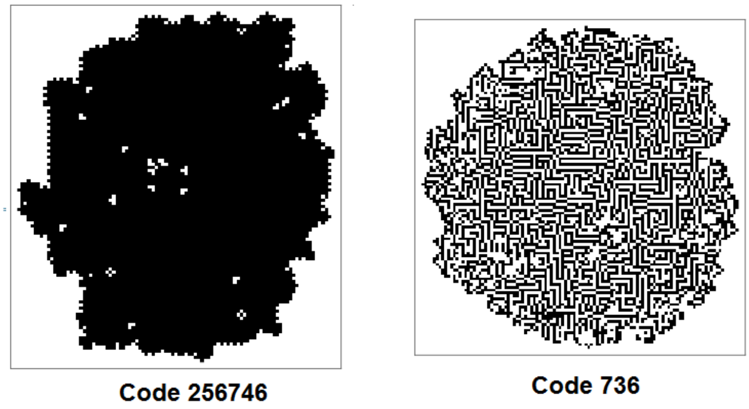


Figure 2.8: Structures obtained by evolution of 2D 9-neighbour outer totalistic CA after 150 iterations with non-regular or round shapes.

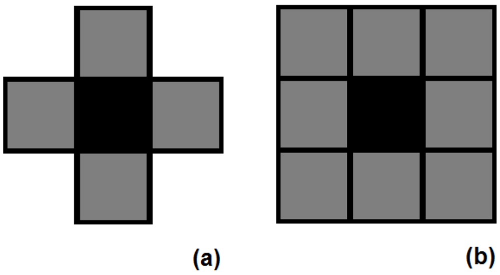


Figure 2.9: Neighbourhoods for 2D cellular automata : Neumann neighbourhood (a); (b) Moore neighbourhood.

2.5 Historical review of CA models applied for computer simulation of the recrystallization and grain growth processes

As was noted by Atkinson [22], all models, called mean field theories led to the grain growth exponent of 2, due to neglecting of the topological requirements. He gives the following classification for computer simulation models of soap froth or polycrystalline grain growth: the model can be statistical or representing direct simulation, which in turn is distinguished between deterministic and probabilistic. The deterministic type implies solving the motion equation for velocity boundary, but one may define a separate computer procedure for simulating a whole evolving structure. The most common probabilistic method is the Monte Carlo simulation method, although there are also non-Monte-Carlo models. Among the works devoted to computer simulations, Atkinson highlights the Monte Carlo simulations performed by Exxon group which focus on the study of grain growth [27, 61, 29], abnormal grain growth [62] and recrystallization [63] using Monte Carlo simulation on a discrete lattice, which considered topological requirements automatically, as they are the direct consequences of the evolving model. The results differ from the empirical theories, for instance,where the grains can grow or shrink, though not always in good agreement with experimental data [61].

Comparison of the main simulation methods such as Monte Carlo Potts method, cellular automata, phase field method, vertex and level set models is presented in [20]. Cellular automata are widely used due to their good scalability, high spatial resolution, ability for efficient parallelization. Previously the cellular automata approach has been used in different areas of material science, such as solidification, primary static recrystallization and recovery, nucleation and grain coarsening.

2.5.1 Solidification processes

Spittle and Brown, 1989 adapted the probabilistic rule-based Monte Carlo method, developed by Srolovitz *et. al.* [27, 61, 29, 63], to model the process of the solidification (columnar and equiaxed grain structures) based on assumed nucleation behaviour [64], with the following attempts to consider latent heat absorption during remelting, grains movement, heat conduction, and therefore to predict qualitatively the influence of the superheat and mould temperature [65]. However the model still did not take into account the solute redistribution, as only uniform composition was assumed. The solute redistribution, which depends on the alloy composition and the constitutional supercooling parameter has been introduced in the Spittle's next work [66]. Later the authors used probabilistic CA model [67, 68] to predict kinetics of the steady-state "free" dendrite growth with incorporating latent heat evolution and heat diffusion. The investigation of the influence of the initial undercoolings and solid-liquid interfacial energy on the final morphology of dendrites was carried out in [69] using this method. An attempt to quantifying results was made by calculating the hydraulic radius. Using CA for steady-state growth of columnar dendrites in binary alloys was qualitatively demonstrated in [70].

Zhu and Smith, 1992 [71] used the nucleation site distribution for heterogeneous nucleation and probabilistic growth, including changes in bulk free energy and interface energy. Therefore, compared to earlier models, columnar or equiaxed grain structures will be evolved based on the free energy change, i.e. for itself. Also, they converted simulation time (Monte Carlo Steps) to real time.

Rappaz and Gandin, 1993 developed two- [72] and simplified three-dimensional [73] cellular automata model for solidification of dendrite alloys, using probabilistic heterogeneous nucleation in the bulk of the liquid and at the mould wall with two different nucleation site distributions deduced from the experimental data and probabilistic growth of dendrite tips, which was calculated via KGT (Kurz-Giovanola-Trivedi) model. The principle of "best alignment" criterion of crystallographic orientation of dendrite grains was taken into account which allowed the model to predict quantitatively columnar to equiaxed transition. In order to make the orientation of growing dendrite tips independent on CA grid, but to keep the original anisotropy, the dendrite tip correction has been made. The limitation of these models was the assumption of the uniform temperature field. Therefore, the suggested CA model [72] was extended to the case of non-uniform temperature was coupled with Finite Element (FE) heat flow calculations [74] (2D), [75, 76, 77](3D) where latent heat realized by the cells is fed back into the FE model. The square growth and "decentred octahedron" algorithms, respectively, were used to avoid the bias of the results due to anisotropy associated with the network of CA.

Cortie, 1993 applied Cellular Automata technique to simulate the solidification of a hypothetical liquid [78] by taking into account the anisotropy properties of the orthogonal lattice.

2.5.2 Static and dynamic recrystallization

The first primary recrystallization simulations (grain nucleation and grain growth) using two-dimensional cellular automata were implemented by **Hesselbarth and Göbel, 1991** in [79]. Results were in agreement with the classical JMAK theory.[1] They used the following neighbourhoods: Moore, Neumann, 7-cells- and 25-cells-neighbourhoods. In contrast to the nucleation process, grain growth depends on the neighbourhood. In Hesselbarth's work, ho-

[1] It is considered the extended volume fraction $x_e = kt^n$ which would be recrystallized if the entire matrix was still unrecrystallized, i.e. in the absent of overlap. Assuming $\dfrac{dx}{dx_e} = 1-x$, where $x(t) = \dfrac{V_{rec}}{V_{tot}}$ is the recrystallized matrix fraction or the degree of recrystallization, the characteristic equation for JMAK theory can be written as $X = 1 - \exp(-kt^n)$, where n is Avrami exponent.

mogeneous and heterogeneous recrystallization were distinguished, and they studied the former type. Firstly, a cell switches its state if it has at least one recrystallized cell in its neighbourhood. Three different model assumptions of inhomogeneous recrystallization were considered: a) impeded impingement, where adaptive lattice is used; b) variation of nucleation rate; and c) variation of the grain growth rates over computed grid. The results are presented in the form of the double logarithmic Avrami-plot with $\log(t)$ along $x-$axes and $\ln\ln\dfrac{1}{1-x}$ along $y-$axes. The authors considered the effect of the impingement of grains that allowed them to explain the deviation of the experimental results from JMAK-theory.

Hesselbarth showed that the concept of neighbourhood influences only the shape of the recrystallization grains, but not the JMAK time exponent; the same result but with 95% accuracy was obtained by **Davies, 1995** [80]. He investigated the influence of the neighbourhood on the coefficients in the JMAK equation and found an effect on the Avrami exponent n, but reported no effect on the nucleation, growth and geometry of grains dependent constant k. The results are in agreement with JMAK theory as simulated recrystallization was implemented in particle-free matrix.

Although results of classical JMAK theory are not changed in the presence of immobile discontinues particles with isotropic impingement, however, as shown by **Pezzee and Dunand, 1994** [81], oriented continues particles due to anisotropic impingement have an influence on the recrystallization kinetics and topology. The authors modified the Hesselbarth's model to investigate the geometric impingement influence of immobile particles on the properties of the recrystallized grains and kinetics of recrystallization. The kinetics are slowed (Avrami exponent n falls) and hence the mean grain size reduces accompanied by a corpuscle size reduction and the growing of the particle fraction and aspect ratio. Also, it was found that the growing corpuscle aspect ratio, the grain size and the area fraction increase the grain average aspect ratio. The results mentioned above are more significant for particles which have the maximum aspect ratio, for example elongated fibres, as they interrupt grain growth in the perpendicular directions orthogonal to the fibres. At high and low recrystallized proportions deviations from JMAK-theory were found, as well as in the Hesselbarth's model [79]. The reason for these deviations is thought to be in the stochastic scatter due to lacking statistics (small number of growing grains). **Shelton and Dunand, 1996** [82] extended the model to the case of mobile particles and showed the existence of an effect of their clustering along boundaries of growing grains.

Among the deficiencies of the aforementioned models one can underline that they were not related to any real physical systems, were not calibrated in space and time. **Davies, 1997** [33] using two different approaches for calculation the interface migration rate (the theoretical constant velocity model and the Cahn-Hagel interface averaged migration rate) applied the 3D CA for study kinetics of recrystallization of pure single crystal iron calibrated it to real time. The parameters were based on the experimental data of the recrystallization of single crystal iron. Used 3-dimensional models gave a more adequate reflection of the physical characters, such as boundary mobility, as it depends on misorientation between grains and orientation in its space (boundary plane inclination). Further, **Davies and Hong, 1999** [83] studied the static recrystallization of three classes of texture using 3D CA, where nucleation and growth probabilities based on experimentally determined nucleation rates and Cahn-Hagel interface migration rates were introduced.

Independently, **Marx, 1996** with co-workers, developed a deterministic 3-dimensional CA (described in [84, 85]) for simulation texture evolution during primary static recrystallization and static recovery in fcc and bcc alloys. They modified it in order to use less computer resources: in the main memory only information about boundaries was kept, as informational content about interior of nucleus expected to be low. The CA model included 3 sequentially called routines: recovery, which decreases dislocation density inside cells; nucleation and nucleus growth. At the same time, this model was used by **Raabe, 1996** and co-authors [86, 87] where they introduced a probabilistic three-dimensional CA model to simulate primary static recovery and recrystallization in real time and space scales by using experimental grain boundary mobility and energy. Their approach allowed to investigate the influence of the orientation dependence of recovery on the final microstructure. The probabilistic analogue of Turnbull rate equation [21] was used as local transformation rule in order to calculate the switching probability. The switching equation was scaled, normalized and was expressed through the local values of mobility depending on local misorientation, driving force depending on the dislocation density and maximum statistical variance. The decrease of the cell size leads only to increasing in the time step required for one switch, but does not change switching probability. The decision about cell switch based on local switching probability was made by evaluating it using the Monte Carlo algorithm.

A study by **Goetz and co-workers, 1998** [88] is dedicated to static recrystallization with both heterogeneous and homogeneous recrystallization resulted in site-saturation and continues (nucleation with constant rate) nucleation on 2-D and 3-D. The results match the JMAK theory in the case of homogeneous nucleation, both with saturation and constant rate. However, in this work they did not take into account the misorientation angles. In the heterogeneous recrystallization Goetz et al. explored the nucleation at the parent boundaries with extension to the case when first wave of nucleation occurs at the parent grain boundaries, and the following – at the recrystallized/unrecrystallized grain boundaries. To analyse such results they used the Vandermeer's microstructural path method [2] [89] and showed agreement with it. A lamellar microstructure was obtained during site saturation and transition from clusters at low nucleation rate to necklace-type microstructure at high rates.

Dynamic recrystallization was firstly simulated using CA by **Goetz and Seetharaman, 1998** [90] and then by **Ding and Guo, 2001** [14] where they combined the CA method which was used to predict grain growth and metallurgical principles for DRX to determine the variations of dislocations density, the growth kinetics of dynamically recrystallizing grains, the flow stress and the nucleation rate of DRX. The last was supposed to depend on the deformation temperature and the strain stress and can be estimated from the experimentally obtained percentage of DRX. The growth kinetics for each recrystallized grain is described by the driving force, and the transformation probability for the sites lying on the allowable growth distance is calculated by the CA. Simulation was implemented at various temperatures and strain rates. The simulated and experimental percentage of DRX are correlated, differences are explained by the fact that there had not considered another types of nucleation sites during the simulation. The mean size and percentage of the recrystallized grains are reduced with increasing strain rate and decreasing temperature.

[2]

$$S_V = C_1(1-X)t^m$$
$$S_V = C_2(1-X)[-\ln(1-X)]^q,$$

where C_1, C_2, q, m – constants, S_V is the interface area between recrystallized and unrecrystallized regions per unit volume

Raabe [91] emphasises that the main disadvantage of mesoscale CA method is the inability to reflect physics of the nucleation, such as in Ginzburg-Landau phase field method with respect to the spinodal decomposition. Therefore, Raabe continues, that it would be useful to combine cellular automata and Ginzburg-Landau-type phase field methods.

A developed method in [87, 92, 91] was successfully applied to Ti stabilized interstitial free (IF) steel sheets treated by cold rolling [93]. The texture was divided by $\alpha-$ (cold rolling) and $\gamma-$(annealing) fibres, which differed in the stored dislocation densities. An assumption was made in that the deformation energy related to the dislocation density is inversely proportional to the experimentally obtained EBSD image quality. It was observed that areas with high accumulated local dislocation density recrystallized first that was caused by the localized nucleation. This effect is called 'cluster phenomena' and it apparently disappears with increasing threshold value. Two types of nuclei have been identified: potential and successful densities, the last being located inside or close to inherited grain boundaries of the deformed grains. Also, two types of unrecrystallized areas were found: 1) thermodynamic, for the areas with low dislocation density, due to insufficient driving forces, 2) kinetic, for the areas with high driving forces, but surrounded by almost immobile boundaries. Investigations of the influence of particles on the inherited boundaries, called the 'Zener pining', were also conducted. The dependence of the recrystallization kinetics and the final microstructure on the precipitated volume fraction volume and the mean radius of particles with three pronounced regimes was revealed.

One of the benefits of using discrete models such as CA is the possibility to account for materials heterogeneities. It was shown that this method can be used for the simulation of the primary static recrystallization in deformed aluminium by using the combination of the probabilistic CA method with crystal plasticity finite element method. This coupling approach is described in more detail in [92]. Information about microtexture and accumulated shear that was translated into state variables of CA model (crystal orientation and stored elastic energy (dislocation density)) using as starting data, can be obtained from the experiment (orientation imaging microscopy) or crystal plasticity finite-element simulation as in the present work. For translation, a Finite Element (FE) mesh was mapped onto the cellular automata regular lattice, which was later filled by corresponding values of the translating variables within the Wigner-Seitz cells. These data were used as input for the yield surface calculations [94]. Also, the given coupling approach is considered in another review paper [91].

In these works, the kinetic and thermodynamic instability criteria of nucleation are considered, and nucleation is assumed to be occurred in the regions with high misorientations and local driving force between subgrains. Therefore, the kinetic criteria suggests the formation of large-angle mobile grain boundary and thermodynamic – stored energy changes across this newly formed boundary, that in turn contributes in the driving force allowing boundary movement towards the deformed matrix. At the same time a nucleation model was chosen which does not create new orientations, but only proceeds assigning zero dislocation density to the cell which had dislocation density above some critical value. After that, misorientations were calculated in order to check the kinetic criteria, and only for those cells that meet requirement of minimum misorientation of 15^o were subsequently treated by the growth step using switching equation. The heterogeneity caused the deviation from the classical JMAK theory – the Avrami exponent n was found lower than expected for site-saturated nucleation. This behaviour corresponded to earlier growth competition.

During the primary recrystallization there are two competing processes: 1) shrinkage due to curvature driven force and 2) volume boundary motion caused by stored energy difference. The growth occurs when the grain reaches the critical size. Monte Carlo (MC) Potts model is applicable for modelling the first process and CA is satisfactory for the latter. **Rollett and Raabe** [95] developed the hybrid model, combining Monte Carlo Potts and CA method for considering both these types.

Zhang et al., 2002 [96] applied CA for the grain growth in combination with recrystallization procedure for simulation of both static and dynamic recrystallization of C-Mn steels during hot deformation, that leads to austenite grains refinement. The hexagonal array of cells was used in order to represent fcc-lattice of austenite.

2.5.3 Nucleation and grain growth (normal and abnormal)

One of the first work devoted to study coupled radial grain growth in two-phase systems occurring in varying transformations, such as eutectic or eutectoid growth by application three-dimensional CA was implemented by **Spittle and Brown, 1994** [97]. Hence the experiments to study these transformations are difficult to implement, computer simulations help to understand how evolution of growing two-phase equiaxed grains occurs. They included in the CA model procedures of growth, rejection of atomic or molecular species and diffusion through interface.

Liu et al., 1996 [98] are the first to use cellular automata to simulate the curvature driven grain growth, taking into account only the activation free energy. They obtained a microstructure evolution which was very similar to the experimental data. **Kumar et al., 1998** [99] investigated the nucleation and initial growth of ferrite in austenite using two-dimensional CA. **Brown, 1998** [100] applied combined 3D cellular automaton finite difference method for the simulation of solute diffusion controlled coupled two-phase unidirectional growth. In contrast to [98], where the grain boundary energy did not depend on the misorientation of the grains, **Geiger et al., 2001** [101] studied the influence of orientation value (defining the energy of the grain boundaries) on the normal grain coarsening. They considered the sum of the thermal energy, governed by Maxwell-Boltzman distribution, and boundary energy represented as Hamiltonian depending on the misorientation angle which are included in model through maximum oriental number q_{max}. Therefore, it is shown, that at small q_{max} non-realistic structure is formed and variation of high q_{max} does not affect the rate of coarsening and relative grain area distribution.

The investigation made by **Yu, 2002-2005** [102, 103, 104] was devoted to the application of the two-dimensional CA for the simulation of the normal and abnormal grain growth during reheating under different conditions, such as temperature and reheating time, as these factors are the most important for controlling at manufactures, although the initial microstructure, composition, reheating rate also influence on the grain coarsening kinetics. This model requires the knowledge of the following parameters: abnormal grain growth temperature, fraction of randomly selected grains f, which become abnormal, and two probabilities P_{low} and P_{high}, corresponding to the normal and abnormal growing grains. The Yu's results of the simulation was scaled and compared with the experiments for Nb steels and showed a good agreement.

Hillert [24] defines the normal grain growth as the steady-state, where a number of the defects remains constant and abnormal growth as initial stage before normal. The first stage of the simulation was implemented by two simultaneous processes: 1) nucleation, defined by using probability P_{n1} that each cell with initial zero state will change its state into one of Q possible orientations represented by nonzero value and 2) nuclei growth defined by probability P_{n2} that cell with zero state will be consumed by the cell with nonzero, namely will assign the same value. This stage continues until full vanishing of all zero cells. Depending on the choice of P_{n1} and P_{n2} one can obtain different microstructures, because in this model P_{n1} corresponds to the nucleation rate and P_{n2} – to the nuclei growth rate: if the former is bigger, then the nucleus will be formed more rapidly than will grow and there will be fine-grained microstructure, otherwise grain size will be large.

The obtained microstructure was used as the initial stage for the next stage – the normal grain coarsening. It was conducted through the probabilistic CA method represented by using the following three rules: 1) Estimate which area the cell belongs to: within the grain or on the boundary. If the cell is inner, then the next cell is estimated, otherwise the second rule is applied; 2) Realization of the grain curvature effect; 3) The probabilistic rule, which defines the probability P_{low} to change the state, which is related to the grain coarsening kinetics. Here, homogeneous distribution of the grain boundary energy is assumed, therefore equal probability was set for all grains. It was found that the growth kinetics is increased with the temperature, which was introduced into the model via the probability.

Also, a new method was adopted for abnormal grain coarsening, where by changing the third rule for heterogeneous distribution, which is supposedly caused by non-uniformly distributed pinning pressure due to heterogeneous distribution of particles (their progressive dissolution and coarsening) was applied. The abnormal grain coarsening was simulated in the next steps: 1) firstly, the normal grain growth was implemented as described above; 2) when the temperature reaches the abnormal grain coarsening stage, randomly chosen grains with a certain fraction are assigned a high probability $P_{high} > P_{low}$, P_{high} was chosen as $A \cdot \exp\left(-\frac{Q}{8.3FT}\right)$ and $P_{low} = A \cdot \exp\left(-\frac{Q}{8.3IT}\right)$, where IT and FT are initial and final temperatures, respectively. Results showed non invariant grain growth, that can be divided into three consistent regimes, which are also observed in the experiments of isochronal grains coarsening: 1) suppressed normal grain coarsening occurs at low probability, showing unimodal distribution of the initial grains; 2) once temperature exceeds the certain critical point, abnormal grain growth temperature, those grains which have higher probability grow faster, and duplex region with bimodal distribution is observed, corresponding retained primary grains and new abnormally growing grains. Abnormally coarsening grains have higher mobility, hence they consume other grains until their final disappearance and 3) there comes a stage of normal, but accelerated grain coarsening, as now all grains have the same, higher mobility.

A method for model correlation and experimental scaling was suggested. Despite the fact that this work showed a good agreement with experimental data, it has several limitations, such as:

1. The CA method is two-dimensional, while 2D and 3D structures have different topologies, and 3D is more realistic and comparable with experiments;
2. The model has difficulty with directly coupling the pinning effect, and here it was simply assumed to be equivalent to the heterogeneous distribution of the grain energy;
3. No suitable method was developed for defining fraction of grains,

which turn into abnormal and probability assigning them (selecting probabilities P_{low} and P_{high}) which should be extracted from the boundary energy of the abnormal and primary grains;

4. Also, there is a problem in consistency of the initial stage of isothermal grain coarsening with experiment at temperature 1100 ^{o} C, because of the fact that in simulation grains need more time to show larger grain sizes;
5. Although the model considered particle dissolution, it did not consider the effect of precipitate dissolution on the grain coarsening process directly.

Several other factors, such as the curvature effect, the interface migration and the overlapping of diffusion fields are still in need of consideration in particle coarsening and particle dissolution modelling.

Lan et al., 2006 [34] adopted their CA model [105] to simulate the two-dimensional curvature driven normal austenite growth. The simulation of the whole process of the austenite recrystallization during multi-pass hot rolling with consideration of both dynamic and static recrystallization, and meta-dynamic recrystallization (the process that takes place in the inter-pass zone during at dynamic recrystallization), recovery and curvature driven grain coarsening was implemented by **Zheng, 2008** [106]. The curvature-driven grain coarsening occurred after the impingement of the recrystallized fronts. Additionally simulation of deformation was carried out. However, in these works [34, 105, 106], not absolutely accurate calculation of the grain-boundary curvature κ was carried out.

Janssens, 2010 [1] gives an example of the simulation of the curvature driven grain growth using 2D probabilistic CA ($P = \frac{k}{k_{max}}$, where k_{max} is maximum curvature) with von Neumann neighbourhood. The microstructure obtained in recrystallization simulation was taken as the initial distribution. Simulation was implemented on the periodic grid and this was the reason for the deviation from the analytical prediction. The problem, mentioned also in [95], is how to perform scaling and couple the real world space-time and simulation time steps. This is impossible in periodic grid, because of its equidistantial aspect which leads to the influence of the orientation of the CA grid on the velocity of a grain boundary. Using the irregular grid can help to solve this problem due to its statistical isotropy of the neighbourhood [107, 108]. In [109], this method was expanded to the 3D case with random grid, that helped to improve anisotropy of the computation lattice, that was observed in the previous works. Given randomness of a particular neighbourhood, it must be stored and calculated for each cell. To reduce computational resources, cells from the inner part are removed from the data structure in the case when sub-processes are either neglected or assumed constant. Also it is proposed to circumvent the updating of those cells, which have already recrystallized, using the "inverse" method [3]. However, this is only applicable to recrystallization, where grain growth causes a large energy release, this type is called primary growth. During the secondary growth, occurring due to a curvature of the grain boundaries, movement of the grain boundary may reverse in time, because first grains may shrink. Therefore, the inverse updating scheme is not applicable. A developed method was implemented in [1],

[3] The most straightforward optimization is a simple preselection of the cells: those having a status recrystallized are excluded from the updating scheme performing recrystallization. But especially of the beginning of the recrystallization a lot of cells find no recrystallized cells in their neighbourhood and thus are evaluated for updating without resulting effect. The “inverse” method: if a cell has a recrystallized status then update the status of all cells in its neighbourhood to the status recrystallized.

where Janssens gives an example of the modelling of two different processes such as solute diffusion and grain growth which occur simultaneously. He uses an approach whereby both processes are implemented sequentially on infinitesimally small time-steps. To simulate abnormal grain growth sequentially calculates the grain growth step, the solute diffusion and the precipitate dissolution step.

2.6 State-of-the-Art

Vertyagina, 2013-2015 [110, 111] suggested a crystal lattice depending method, based on the differentiated defining priority of the interaction between the core cell and its Moore neighbourhood: it is believed that the Neumann neighbourhood (6 cells in the 3-dimensional model) should be considered primarily. The probability of the cells to influence the core cell via edges (12 cells) is probability p_2, and via vertices (8 cells) – p_3, they are significant for fcc – and bcc – structure respectively.

Instead of the local transition rules, local transition functions (LTF), which are inherently probabilistic, were used, as they include real physical laws and allow to take into account real parameters that truly influence the evolving system. At the same time, as was noticed by Vertyagina et al. [110], LTF requires defining its appropriate mathematical form and the area of the application function. At each time-step the value of LTF f_{step} is calculated and its normalized value $f = f_{step}/f_{max}$ with its associated comparison with the random value in the interval [0,1] thus implementing the real Monte Carlo step. LTF were defined as follows: 1) 2D case: $f_{max} = (4p_1 + 4p_2)^2$, $f_{step} = (N_1p_1 + N_2p_2)^2$; 2) 3D case: $f_{max} = (6p_1 + 12p_2 + 8p_3)^3$, $f_{step} = (N_1p_1 + N_2p_2 + N_3p_3)^3$, where N means the number of cell surrounding the core cell and belonging corresponding group. The probabilities p_2 and p_3 were introduced via the ratio of v to v_{max}: $p_{2,3} = \frac{v}{v_{max}} = \left(\frac{D_0}{D}\right)^m$, where D_0 is initial grain size, m was chosen $m = 2$. The probability p_1 was asserted to be not related to the crystal structure, but it depends on the temperature T: $p_1 = A \exp\left(-\frac{Q}{RT}\right)$.

The model allows to extract the grain boundary velocity v (first derivative of the grain size), driving force $p = \alpha\frac{\gamma}{D}$ and the grain boundary mobility (was calculated by the equation $v = Mp^m$), where γ is the grain boundary energy per unit area, α and m are constants, D is the mean grain size. Calibration operations were performed on the simulated temperature, time and space with experimental data using method described by Yu [102]. Results of temperature calibration despite the general good agreement showed some deviations from the experimental points.

Verification has been made via comparison of the experimental and calculated values of data, based on the points of line intersections for the different types of lattice for Gibbs energy. This model showed a good agreement with experimental data, however there are deviations at the temperatures lower than 1373 K. The authors suggested that it was caused by the incorrect choice of parameters in LTF and can be explained by the fact that another mechanism controlling the grain growth may exist.

Zhu et al., 2014 [112] investigated by means of the two-dimensional CA the microstructural evolution during hot stamping of the ultra-high strength steel (UHSS), consisted of pearlite and ferrite at room temperature. The following stages were modelled: 1) nucleation and growth of austenite in pearlite; 2) diffusion-controlled growth of austenite in ferrite which occurs at higher temperature, the steel was reduced to ternary system Fe-Mn-C and chemical driving force was calculated using FDM. For the first process, growth is assumed to be driven by the free energy reduction, the second – is induced by chemical potentials difference of iron in austenite and ferrite and carbon diffusion was taken into account; 3) curvature driven coarsening of austenite. The real micrograph of UHSS was used as initial input structure. The calculated mean grain size deviated from the measured, according to authors because of neglecting of solute drag effect and the pinning. The fact that the switching probability for the pearlite-to-austenite transformation depended on the lattice size, caused influence of the lattice on the early stage of the austenization. In addition, usage of FDM techniques for simulation of carbon diffusion and the algorithm for separating different grains led to large calculation time.

Han et. al, 2015 [37] obtained a modified Zener pinning relation (with the consideration of the initial grain size) based on simulations of grain growth in the presence of second-part inert particles, which did not evolve, by using realistic mobilities and driving forces, that led to real time and space. The influence of initial grain sizes and particles fraction and sizes on microstructural evolution was investigated using deterministic CA firstly described by **Zheng, Raabe and Li, 2012** [113]. The new discrete disk template method with a diameter of a 15 cells was used for a calculating boundary curvature, which was found to be much effective than in the previous works [34]. Despite this attempt to include second-part phase particles by **Han et. al, 2015** [37], there is still lack of CA models considering the influence of defects and precipitation on the microstructural evolution, that will be in a good agreement with experimental data.

As can be seen from the considered above works, the initial grain distribution influences on the final microstructure. However, by now no physically realistic methods were used for nucleation stage. Due to difficulties of the obtaining the experimental micrograph, the realistic prediction of initial nucleation stage remains important problem to be solved.

2.7 Summary

This chapter has provided an extensive literature survey on the history of using the CA techniques for modelling of recrystallisation and grain growth phenomena in metals. The introduction to the theory of CA has been provided in order introduce the reader with the different classification of CA techniques and its application to the material science. The chapter also gives briefing to the theory of recrystallization and grain growth. It has been shown that there is still lack of the theory for nucleation processes and mechanisms forming grain growth. The literature review has shown that the latest research on CA technique has achieved high precision of prediction the real processes of the grin growth.

Literature review has shown benefits of usage of CA in the modelling of complex dynamic systems such as grain growth and recrystallization. It is a discrete method, which allows to reflect any physical mechanism by an appropriate choice of rules and have a high spatial and time resolution. Therefore CA approach has been chosen as a modelling technique for modelling grain growth and recrystallization in the present research.

Chapter 3

Modelling of a Single Grain - Separate Driving Forces

3.1 Modified LTF functions

During the present research method suggested by Vertyagina (LTF [110, 111]) has been modified and called modified LTF. The original code has been written in Delphi. It has been optimized and rewritten in Matlab for faster calculations, but with savings of the original algorithm.

The cells attached to the central by edges are called as Group I and represent Neumann neighbourhood, and cells within rotated Neumann or attached by vertices are called Group II (see Fig. 3.1). The original method suggests that a cell can be transformed only to a cell from Group I , without consideration of Group II.

The modified approach considers both groups, but with a different weights. The core cell will change its state to another state which is the most frequent state, even if this cell lies in Group II. The difference is in the way of calculat-

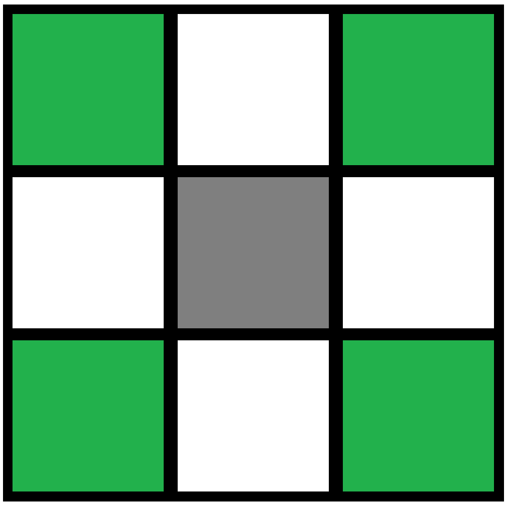


Figure 3.1: Positions of Group I (white cells), Group II (green cells) and core cell(grey). [110, 111].

ing numbers N_1 and N_2. The original method defines N_1 and N_2 as follows:

$$N_1 = (number\ of\ the\ cells\ in\ Group\, 1\ with\ the\ state\ A) - 1, \tag{3.1}$$

where A is the most frequent state in Group 1.

$$N_2 = number\ of\ cells\ in\ Group\, 2\ with\ state\ A. \tag{3.2}$$

The following modifications of these definitions N_1 and N_2 have been made:

$$W_i = g_1 + 0.5\,\frac{g_2(g2+1)}{2}, \tag{3.3}$$

where g_1 is the number of cells with state A_i in Group 1, g_2 is the number of cells with state A_i in Group 2, A_i is the unique state in Group I and II.

$$N_1 = \ (number\ of\ cells\ with\ the\ state\ A\ in\ Group\ 1), \qquad (3.4)$$

$$N_2 = a\ (number\ of\ cells\ with\ the\ state\ A\ in\ Group\ 2), \qquad (3.5)$$

where A is the state corresponding to the maximum W_i, $a = 0.3$ is a fitting parameter that will be describe below. The principle of randomness also been added the code: if the weights of two different states are equal then the states are chosen as the next step state of the core cell with the equal probabilities.

As a result, the structure of the grown grains become more similar to the observed in the experiments. Fig. 3.2(a-b) simulated microstructures obtained via original and modified method from the same initial microstructure shown in Fig. 3.2c. The simulations have been run until the average grain size become more than $d = 20$. The modified method allows to obtain grains with smoother boundaries. The 5-side form, which is inherent with experimentally observed structures [114] (see Fig. 3.2d), has become more distinguishable.

3.1.1 Comparison with the experimental data

The 2D simulations of the normal grain growth have been compared with the experimental data on the austenite grain growth in the precipitate-free variant of APIX60 steel performed by Zhou [114]. In order to quantitatively compare simulation results with the experiments, the data from article [114] has been digitized, and is shown in Table 3.1. The corresponding plot of the data is shown in Fig.3.3. The obtained grain growth exponents n at the different annealing temperatures T are also given in the Table 3.1.

It has been found that a influences the range of possible p_1: grain growth exponents showed values from 0.42 to 0.47 at varying p_1 from 0 to 0.9. Therefore, the series of the simulations have been carried out in order to choose the

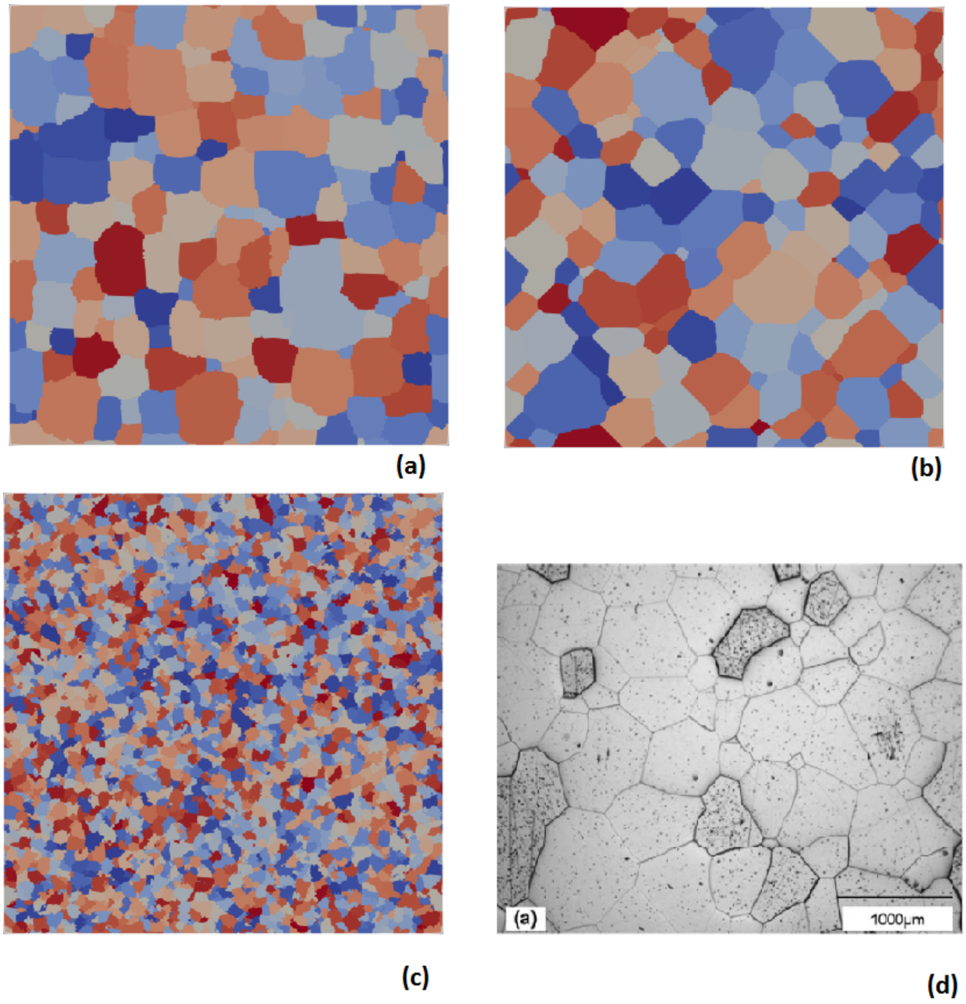


Figure 3.2: Simulated microstructure using original (a) and modified LTF (b) method obtained from the same initial structure (c) in comparison with the experimental structure (d) [114].

Temperature, K	Average diameter, mm	Time, s	Grain growth exponent
1473	144	305.141	0.3852
	201	902.156	
	285	1797.68	
	366	3601.99	
1573	330	298.507	0.5222
	552	590.381	
	741	895.522	
	825	1200.66	
	1110	2706.47	
1673	375	119.403	0.5489
	642	305.141	
	960	597.015	
	1110	902.156	

Table 3.1: Average grain diameter with time at different temperatures in the austenite grain growth experiments by Zhou [114] and obtained grain growth exponents.

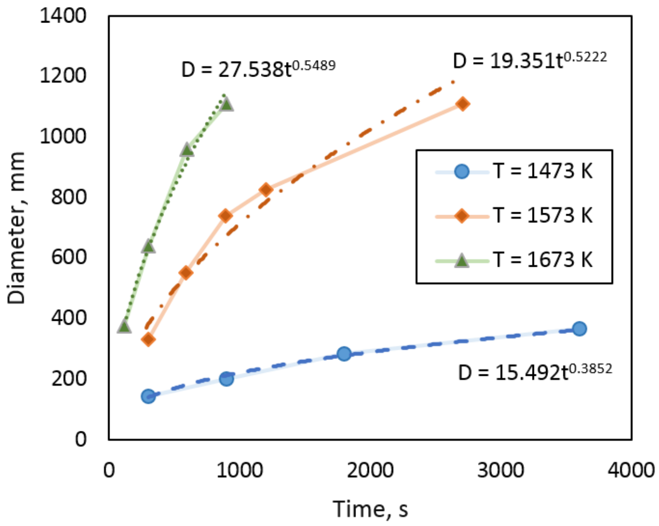


Figure 3.3: The kinetics of the austenite grain growth at the different temperatures obtained in the experiments by Zhou [114]. The dashed lines show regression curves with the corresponding regression equations.

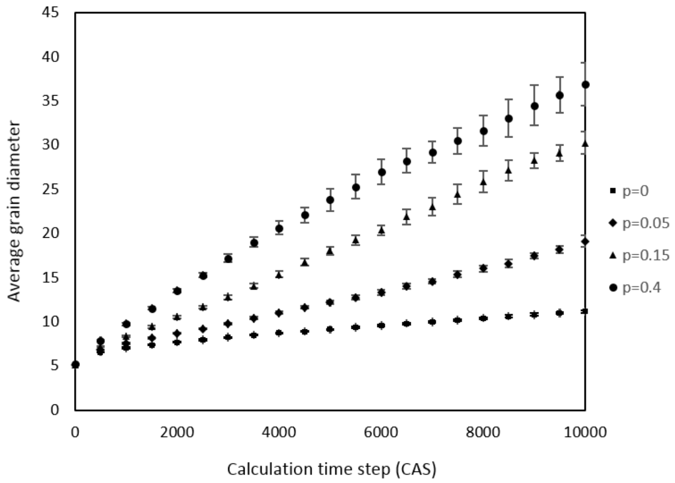


Figure 3.4: Simulated average grain size dependence on the probability p_1.

appropriate value of a. Parameter $a = 0.3$ gives sufficient range of magnitudes of grain growth exponents. Calculations were implemented on the 250x250 cells area and the calculation time was taken to be 10000 steps. The grain size evolution is shown in Fig. 3.4 at the different values of probabilities p_1 which is correlated with the temperature T. Each simulation has been repeated more than 15 times. As it can be seen, the deviations grows with increasing probabilities.

The dependence of the grain growth exponent n on the value of probability p_1 at $a = 0.3$ is shown in Fig. 3.5. The experimental grain growth [114] lies in the range of $n = 0.3852$ at $T = 1473\,K$ and $n = 0.5489$ at $T = 1673\,K$, that corresponds to the model probabilities $p_1 = 0.056$ and $p_1 = 0.25$.

Given the following form of equation:

$$p_1 = A \exp\left(-\frac{Q}{RT}\right), \tag{3.6}$$

one can write the following system of two equations:

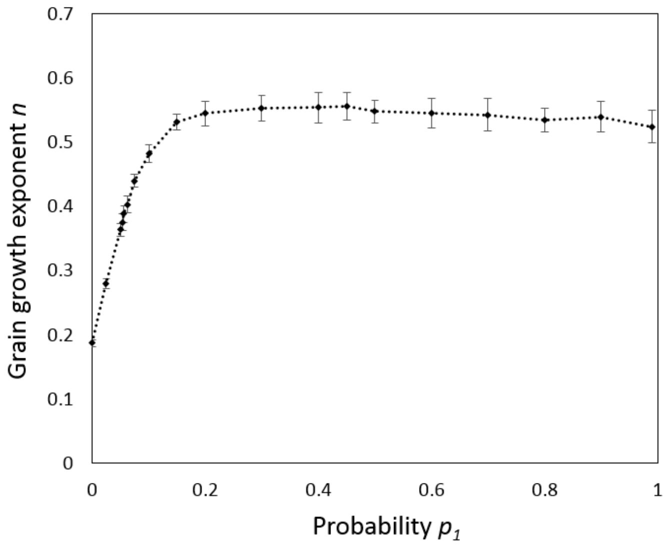


Figure 3.5: Simulated grain growth exponent dependence on the probability p_1.

$$0.056 = A \exp\left(-\frac{Q}{8.31 \times 1473}\right), \tag{3.7}$$

$$0.25 = A \exp\left(-\frac{Q}{8.31 \times 1673}\right), \tag{3.8}$$

that gives $A = 15253.28$, $Q = 153.19\ kJ/mole$. Therefore, the relationship between the real temperature and modelling probability p_1 can be written as

$$p_1 = 15253.28 \exp\left(\frac{153190}{8.31T}\right). \tag{3.9}$$

Substituting temperature $T = 1573$ K gives $p_1 = 0.124$. The experimental grain growth at the given temperature is $n_{exp} = 0.5222$, simulation is $n_{sim} = 0.4928$, or 5.63% of discrepancy. In comparing with the previous results obtained in the original model [110], where n was about of 0.415, the

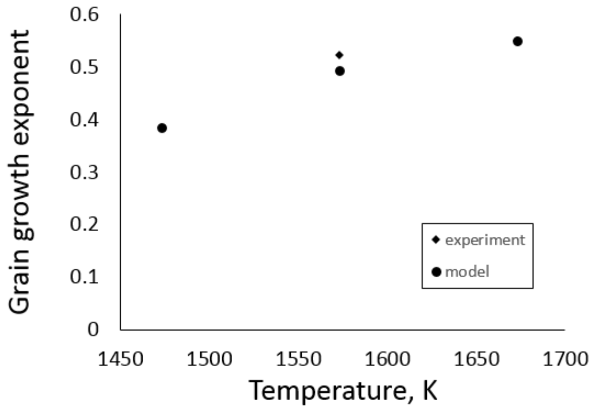


Figure 3.6: Results of the calibration of the simulated data with experiments [114].

present result is closer to the experimental data. These results are shown in Fig. 3.6.

3.1.2 Motivation of study on a single grain

At the beginning of the present research the modified LTF CA has been suggested and has shown the simulated structures are more similar to the 5-side form of the grains, observed in the experiments than with the original method. The calibration with experimental data showed a discrepancy equal to 5.63% that is smaller in comparison with the original model. However as can be seen from this method, the real experimental data is required to calibrate simulation parameters. In the light of the lack of the sufficient experimental data, it has been decided to conduct the future research on the grain boundary movement from the first mechanical principles and compare simulation results with mathematical solutions instead of experimental.

3.2 Analytical solutions

The dynamics of the single spherical grain shrinkage simulated by CA can be compared with the analytical solution, where the radius of the sphere or the disk for the 2D case is a function of time. The grain boundary velocity magnitude is given by following formula:

$$v = Mp, \tag{3.10}$$

where p is the driving pressure, the mobility $M = \frac{b\delta D_{ob}}{kT} \exp\left(-\frac{Q_b}{RT}\right)$ (see Eq.(2.27)).

In this work the pressure $p = p_C + p_D$ arising from the curvature is given by the Gibbs-Thomson relationship $p_C = -2\frac{\gamma}{r}$, and the dislocation density difference $p_D = \tau[\rho]$ (the pinning pressure was no considered yet) [4]. Here, γ is a grain-boundary energy, the dislocation line energy $\tau = \alpha\mu b^2$, α is a constant of the order of 0.5, b is magnitude of Burgers vector, μ is shear modulus Eq. (2.29) and the jump in the dislocation density across the boundary $[\rho]$ is the dislocation-density difference between grains, $\triangle\rho = \rho_i - \rho_j$.

3.2.1 Constant dislocation density jump

Let us consider the grain boundary motion of the single grain driven only by the dislocation density jump,

$$v = v_D = Mp_D, \tag{3.11}$$

$$p_D = \tau[\rho], \tag{3.12}$$

where the kinetics of the grain radius r can be written as follows:

$$\frac{dr}{dt} = M\tau[\rho] \tag{3.13}$$

$$\int dr = \int M\tau[\rho]dt \tag{3.14}$$

and then the analytical solution is as follows:

$$r(t) = r_0 + M\tau[\rho] \cdot t \tag{3.15}$$

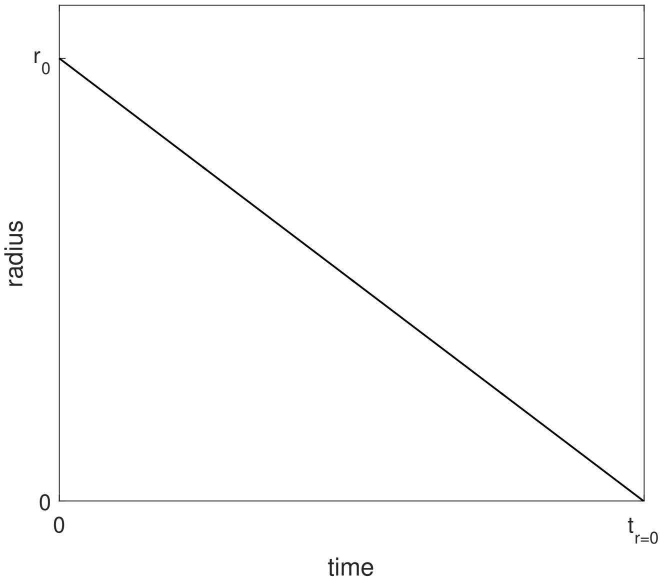


Figure 3.7: The change of the radius of a sphere shrinkage with a constant velocity due to dislocation density jump.

In the case of a negative $[\rho]$, the grain is shrinking with a constant velocity. Schematically, the evolution of the grain radios with time is shown in Fig.3.7.

3.2.2 Evolving dislocation density jump

3.2.2.1 Notation for time

Each cell (i, j) if its 2D domain or (i, j, k) for 3D is assigned at the beginning $(t = 0)$ the initial dislocation density ρ_{ini}. The KM model, described in literature review, has been used for the simulations dislocation density. Every time step, with time increment Δt, at each point (i, j) dislocation density and strain are calculated by following system of equation:

$$\begin{cases} \frac{d\rho}{d\varepsilon} = k_1\sqrt{\rho} - k_2\rho, \\ d\varepsilon(t) = \dot{\varepsilon}dt \end{cases} \tag{3.16}$$

The first equation of the system Eq. (3.16) (KM model [13]) is a nonlinear first order ODE expressing relationship between local dislocation density ρ_i in the each ith grain and true strain ε, that physically related to the accumulation of the dislocations due to: 1) deformation during work hardening – the first term on the RHS of equation, where k_1 is a constant related to hardening; 2) DRV — the second term, where $k_2 = \frac{\alpha\mu b k_1}{\sigma_{st}}$ and steady-state stress σ_{st} is a function of temperature T and strain rate $\dot{\varepsilon}$: $\sigma_{st}(T, \varepsilon) = \left(A_1\dot{\varepsilon}e^{Q_A/RT}\right)^{1/A_2}$, α is the dislocation interaction coefficient, μ is the shear modulus, b is the magnitude of the Burgers vector and Q_A is the activation energy. The constant strain increment is described by the second equation of the system Eq. (3.16), where $\dot{\varepsilon}$ is the strain rate. The system of these equations can be rewritten as follows:

$$\frac{d\rho}{\dot{\varepsilon}dt} = k_1\sqrt{\rho} - k_2\rho \tag{3.17}$$

The initial condition $\rho(0) = \rho_{ini}$ gives the solution $\rho(t)$ as follows:

$$\rho(t) = \frac{1}{k_2^2}\left(\sqrt{\rho_{ini}}k_2e^{-\frac{1}{2}k_2\dot{\varepsilon}t} - k_1e^{-\frac{1}{2}k_2\dot{\varepsilon}t} + k_1\right)^2, \tag{3.18}$$

or

$$\rho(t) = \frac{1}{k_2^2}\left((\sqrt{\rho_{ini}}k_2 - k_1)e^{-\frac{1}{2}k_2\dot{\varepsilon}t} + k_1\right)^2, \tag{3.19}$$

where the time is $t = \Delta t \cdot n_{step}$ and the strain ε is evaluating at each time step n_{step} as $\varepsilon = \dot{\varepsilon}\Delta t n_{step}$, assuming zero initial strain is $\varepsilon|_{t=0} = 0$.

Each cell is being checked as to whether it lies on the border or not. If it does and the dislocation density at this cell has reached critical value ρ_c then the nucleation is happening. Once the cell has become a new nucleus at certain time t which will we call as t_{ini}, its dislocation density value is returning to its initial value ρ_{ini}.

The fact that the newly formed nucleus will have initial dislocation density ρ_{ini} means that in Eq. (3.17) the initial condition $\rho|_{t=0} = \rho_{ini}$ should be changed to $\rho|_{t_{ini}} = \rho_{ini}$. The more general solution of Eq. (3.17) will be as follows:

$$\rho(t, t_{ini}) = \frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2}{k_2^2} \tag{3.20}$$

The usage of this form of equation allows to calculate dislocation density implicitly in the simulations and obtain the analytical values at any iteration regardless of the size of time increment. As can be seen this solution leads to the particular case Eq. (3.19) at $t_{ini} = 0$.

The condition for the nucleation $\rho = \rho_c$ can be hence written as follows:

$$\frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2}{k_2^2} = \rho_c \tag{3.21}$$

From this equation $t - t_{ini} = \Delta t_c =$ can be found, which will be used in the CA simulations:

$$\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}\Delta t_c}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2 = \rho_c k_2^2 \tag{3.22}$$

$$e^{-\frac{1}{2}k_2\dot{\varepsilon}\Delta t_c}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1 = \pm\sqrt{\rho_c}k_2 \tag{3.23}$$

$$e^{-\frac{1}{2}k_2\dot{\varepsilon}\Delta t_c} = \frac{k_1 \pm \sqrt{\rho_c}k_2}{k_1 - \sqrt{\rho_{ini}}k_2} \tag{3.24}$$

$$-\frac{1}{2}k_2\dot{\varepsilon}\Delta t_c = \ln\frac{k_1 \pm \sqrt{\rho_c}k_2}{k_1 - \sqrt{\rho_{ini}}k_2} \tag{3.25}$$

$$\Delta t_c = -2\frac{1}{k_2\dot{\varepsilon}}\ln\frac{k_1 \pm \sqrt{\rho_c}k_2}{k_1 - \sqrt{\rho_{ini}}k_2} \tag{3.26}$$

3.2.2.2 Analytical solution of single grain shrinkage driven by evolving dislocation density driving force

The solution Eq. (3.15) has been obtained for the constant dislocation density jump $[\rho]$. Let us consider the case of dislocation density evolving according to Eq. (3.19). The equation for grain radius is the same as Eq. (3.13), but the variable $[\rho]$ depends on time t:

$$\frac{dr}{dt} = -M\tau[\rho](t), \tag{3.27}$$

where the negative sign in (3.27) shows that equation represents the shrinkage of the deformed grain, instead of the growth of the recrystallized grain, $[\rho](t)$ is the difference between dislocation densities of the deformed and the recrystallized grains: $[\rho](t) = \rho_d(t, t^d_{ini}) - \rho_r(t, t^r_{ini})$. $\rho_d(t, t^d_{ini})$ and $\rho_r(t, t^r_{ini})$ can be rewritten using Eq. (3.20) as follows:

$$\rho_d(t, t^d_{ini}) = \frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2}{k_2^2}, \tag{3.28}$$

$$\rho_r(t, t^d_{ini}) = \frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2}{k_2^2}, \tag{3.29}$$

Therefore,

$$\begin{split}[\rho](t) = \rho_d(t, t^d_{ini}) - \rho_d(t, t^r_{ini}) = \frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d_{ini})}(-\sqrt{\rho_{ini}}k_2+k_1)-k_1\right)^2}{k_2^2} - \\ - \frac{\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r_{ini})}(-\sqrt{\rho_{ini}}k_2+k_1)-k_1\right)^2}{k_2^2},\end{split} \tag{3.30}$$

which leads to the following differential equation:

$$\begin{split}\frac{dr}{dt} = -\frac{M\tau}{k_2^2}\Bigg(\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2 - \\ - \left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2\Bigg),\end{split} \tag{3.31}$$

$$\begin{split}r = \int -\frac{M\tau}{k_2^2}\Bigg(\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2 - \\ \left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2\Bigg)dt + C\end{split} \tag{3.32}$$

$$\begin{split}r = \frac{M\tau}{k_2^3\dot{\varepsilon}}\big[(-\sqrt{\rho_{ini}}k_2 + k_1)^2\left(e^{-k_2\dot{\varepsilon}(t-t^d)} - e^{-k_2\dot{\varepsilon}(t-t^r)}\right) + \\ +4k_1(-\sqrt{\rho_{ini}}k_2 + k_1)\left(-e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d)} + e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r)}\right) + k_1^2k_2\dot{\varepsilon}(t^d - t^r)\big] + C\end{split} \tag{3.33}$$

The constant C can be found from the initial condition $r(0) = r_{ini}$:

$$\begin{split}C = r_{ini} - \frac{M\tau}{k_2^3\dot{\varepsilon}}\big[(-\sqrt{\rho_{ini}}k_2 + k_1)^2\left(e^{k_2\dot{\varepsilon}t^d} - e^{k_2\dot{\varepsilon}t^r}\right) + \\ +4k_1(-\sqrt{\rho_{ini}}k_2 + k_1)\left(-e^{\frac{1}{2}k_2\dot{\varepsilon}t^d} + e^{\frac{1}{2}k_2\dot{\varepsilon}t^r}\right) + k_1^2k_2\dot{\varepsilon}(t^d - t^r)\big]\end{split} \tag{3.34}$$

Then the solution can be written as follows:

$$r(t) = \frac{M\tau}{k_2^3\dot{\varepsilon}}\Bigg(4k_1\left(-\sqrt{\rho_{ini}}k_2 + k_1\right)\left(e^{\frac{1}{2}k_2\dot{\varepsilon}t^d_{ini}} - e^{\frac{1}{2}k_2\dot{\varepsilon}t^r_{ini}}\right)\left(1 - e^{-\frac{1}{2}k_2\dot{\varepsilon}t}\right) + \\ + \left(-\sqrt{\rho_{ini}}k_2 + k_1\right)^2\left(e^{k_2\dot{\varepsilon}t^r_{ini}} - e^{k_2\dot{\varepsilon}t^d_{ini}}\right)\left(1 - e^{-k_2\dot{\varepsilon}t}\right)\Bigg) + r_{ini} \tag{3.35}$$

Let us consider the radius with time of the deformed grain surrounded by newly recrystallized. The initial conditions for the dislocation density are as follows: at time $t = 0$ s the deformed grain has critical dislocation density $\rho^d|_{t=0} = \rho_c$ and the recrystallized grain has zero initial dislocation density $\rho^r|_{t=0} = 0$ (or $t_{ini^d} = \Delta t_c$, see Eq. (3.26), and $t_{ini^r=0}$). The parameters for the dislocation density calculations are shown in Table 3.2 and other parameters are the same as in Table 4.1. Fig. 3.8a shows the radius $r(t)$ of a single grain during the shrinkage due to evolving dislocation density difference compared with the driven by the constant dislocation density jump, which has been discussed in the previous section. As can be seen from the figure, the grain has not disappeared by the time $t = 500$ s in contrast the case of a constant $[\rho]$, but approaches the limit $r_{lim} \approx 0.59 \times 10^{-4}$. This can be explained by the fact that the dislocation density difference has decreased with time (see Fig. 3.8b) or in other words the driving force tends to zero.

Parameter	Value	Description
$\dot{\varepsilon}$	2×10^{-3} $1/s$	Strain rate
k_1	2.97×10^{8} m^{-1} [39]	Constant related to the hardening
A_1	1) 3.59×10^{44} [14], 2) 2×10^{44} [39]	Constant
A_2	7.58 [14], 7.6 [39]	Constant
Q_A	1) 261×10^{3} J/mol [14], [42], [41] 2) 275×10^{3} J/mol [39], [17], [16],	Activation energy
ρ_c	5.51×10^{13} m^{-2}	Critical dislocation density
ρ_{ini}	0 m^{-2}	Initial dislocation density

Table 3.2: Parameters.

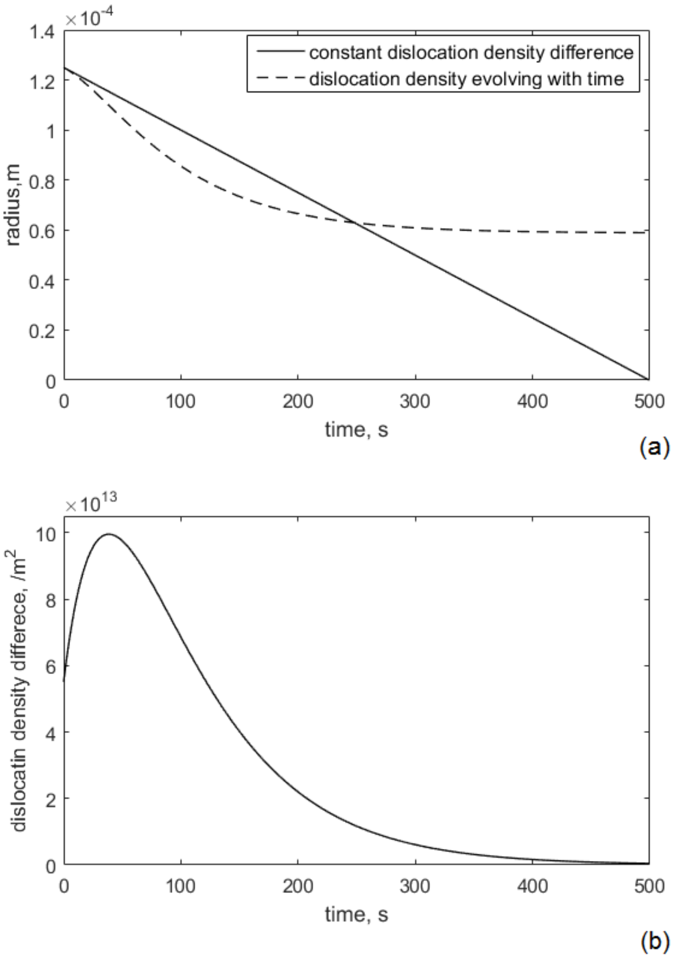


Figure 3.8: Radius evolution during the grain shrinkage due to dislocation density difference: constant and evolving with time. Dislocation density evolution.

3.2.3 Curvature driven grain shrinkage

Let us now consider the case of when the grain is shrinking only by curvature, so that $p_D = 0$.

$$v = v_C = Mp_C, \tag{3.36}$$

$$p_C = -\frac{2\gamma}{r}, \tag{3.37}$$

Therefore, the equation for the grain boundary motion $v = \frac{dr}{dt}$ can be written as follows:

$$\frac{dr}{dt} = -M\frac{2\gamma}{r}, \tag{3.38}$$

$$r \cdot dr = -\int 2M\gamma dt, \tag{3.39}$$

This is a 1st order linear differential equation with the initial condition $r(0) = r_0$.

$$\frac{r^2}{2} = -2M\gamma t + \frac{r_0^2}{2}, \tag{3.40}$$

$$r^2 = -4M\gamma t + r_0^2 \tag{3.41}$$

Therefore, the kinetics of the grain radius can be described theoretically as follows:

$$r(t) = \sqrt{r_0^2 - 4M\gamma t} \tag{3.42}$$

Schematically, the evolution of the grain radius with time is shown in Fig.3.9.

Ding [14] did not calculate curvature, but instead he used directly radius of grains r_i. However, the radius of each grain is maybe unknown in the

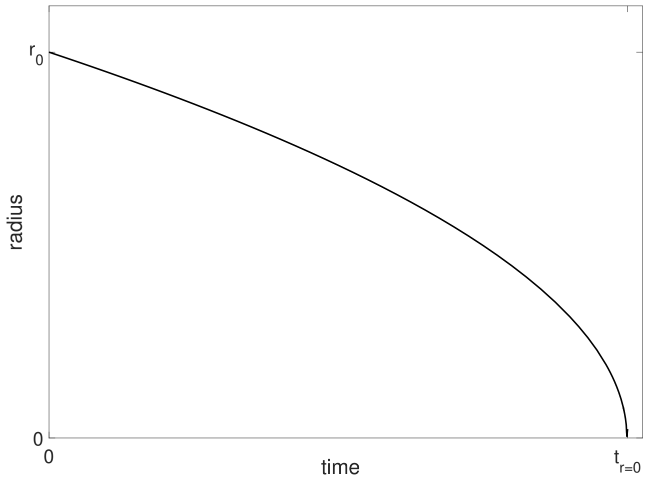


Figure 3.9: Radius evolution during the sphere shrinkage with a constant velocity due to curvature

simulations. It can also be known, but the shape of the grains can be not round, so the driven force is different for the different parts of grain and the local curvature is required. Hence for the CA simulations local curvature k is used:

$$v = M\gamma k \tag{3.43}$$

The curvature k can be calculated and the most common method which is based on Kremeyer's work [115] has already been used in several works in the past (see Lan et al. [34], Zheng et al. [113], Yang et al. [116], Hallberg [42], [41].)

$$k = \frac{A}{a}\frac{Kink - N_i}{N+1}, \tag{3.44}$$

where A is a constant parameter, a is the cell size, N_i is the number of cells in the neighbourhood of $N+1$ cells belonging to the grain, and $Kink$ is a number of cells within the neighbourhood belonging to the grain for a flat interface.

3.3 Algorithm testing for simulation of single grain shrinkage at different driving forces

The common problem of using CA methods for the simulation of the microstructure evolution is a requirement of the time scaling based on the real physics [1]. The present method uses the real time increment Δt, and hence the real distance Δr which the grain boundary passed for this time Δt. Consequently the results of the simulations do not require any further scaling. The simplified algorithm is shown in Fig 3.10, more detailed code is shown in Appendix 3. The algorithm was tested on the structures with different dimensions: 1D line segment, 2D disk and 3D sphere with radius of 25 cells, which are shown in Fig. 3.11.

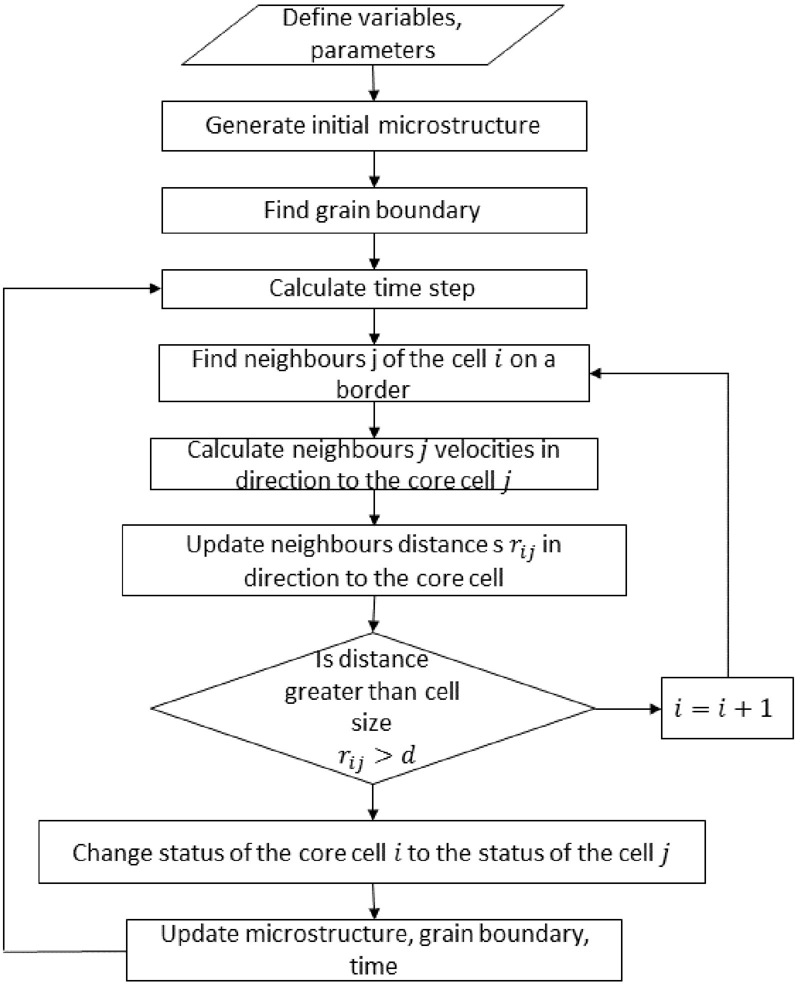

Define variables, parameters
Generate initial microstructure
Find grain boundary
Calculate time step
Find neighbours j of the cell i on a border
Calculate neighbours j velocities in direction to the core cell j
Update neighbours distance s r_{ij} in direction to the core cell
Is distance greater than cell size $r_{ij} > d$
$i = i + 1$
Change status of the core cell i to the status of the cell j
Update microstructure, grain boundary, time

Figure 3.10: The simplified algorithm for CA simulations.

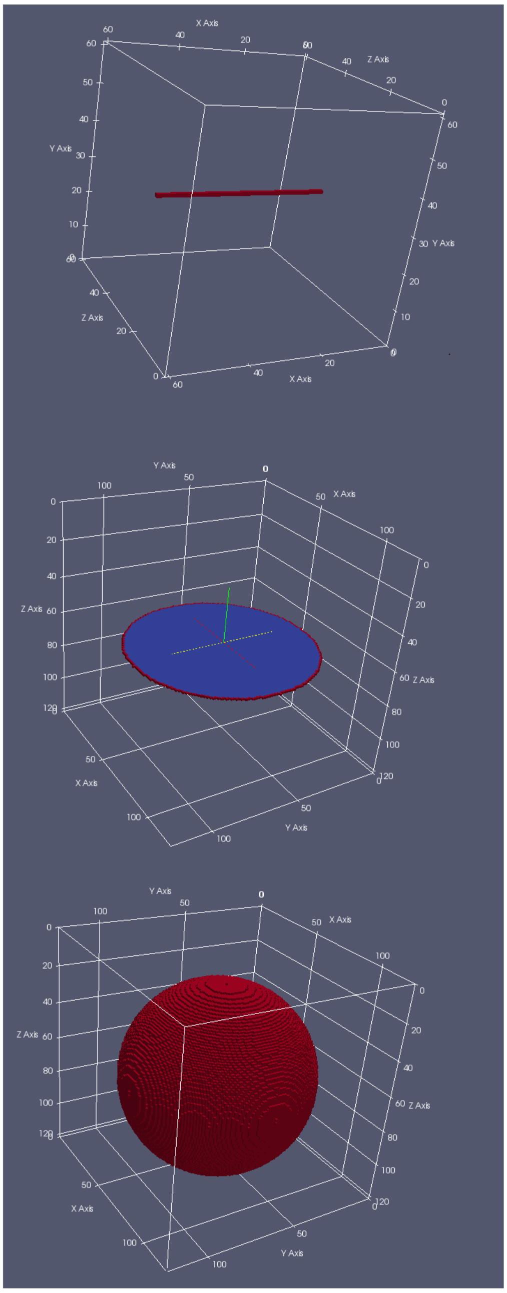


Figure 3.11: Initial structures: line, disk and sphere with radius of 50 cells. The boundary cells are shown by red color. Visualization made in Paraview.

3.3.1 Simulation of a single grain shrinkage driven by constant dislocation density pressure

Similarly to the previous section, we will simulate the grain shrinkage due to three different contributions of the driving forces: 1) the dislocation density difference; 2) evolving dislocation density difference; 3) the curvature.

Assume that the structure consists of cells with a dislocation density ρ_1 and surrounded by cells with $\rho_2 = 0$. The driving pressure will then be $p_D = \tau[\rho] = \tau(\rho_1 - \rho_2) = \tau\rho_1$.

3.3.1.1 Dislocation density difference jump in 1D. Effect of the cells size and time increment

The cells are located on a single line and each cell has two neighbours: on the left and on the right. If a neighbouring cell is moving towards the central position due to the difference in the dislocation density, the state of the central cell changes once the passed distance becomes larger than the cell size.

The change of the radius with time, analytical and simulated, at different time increments and cell sizes are shown in Fig. 3.12. If we choose a time increment as $\Delta t = d_{cell}/v_D^{max}$, then the driving force p_D will move the grain boundary and the segment length will decrease with a constant velocity $v_D = Mp_D$ by one cell from both ends after each iteration. The simulation result perfectly agrees with the theory as can be seen in Fig. 3.12(a). This Δt is a minimal time increment and a larger value will lead to the wrong results, while the smaller time increment will not influence on the kinetics, but the line "radius-time" will have a stepped form (see Fig. 3.12b). This form becomes less obvious with a finer cell structure (see Fig. 3.12c).

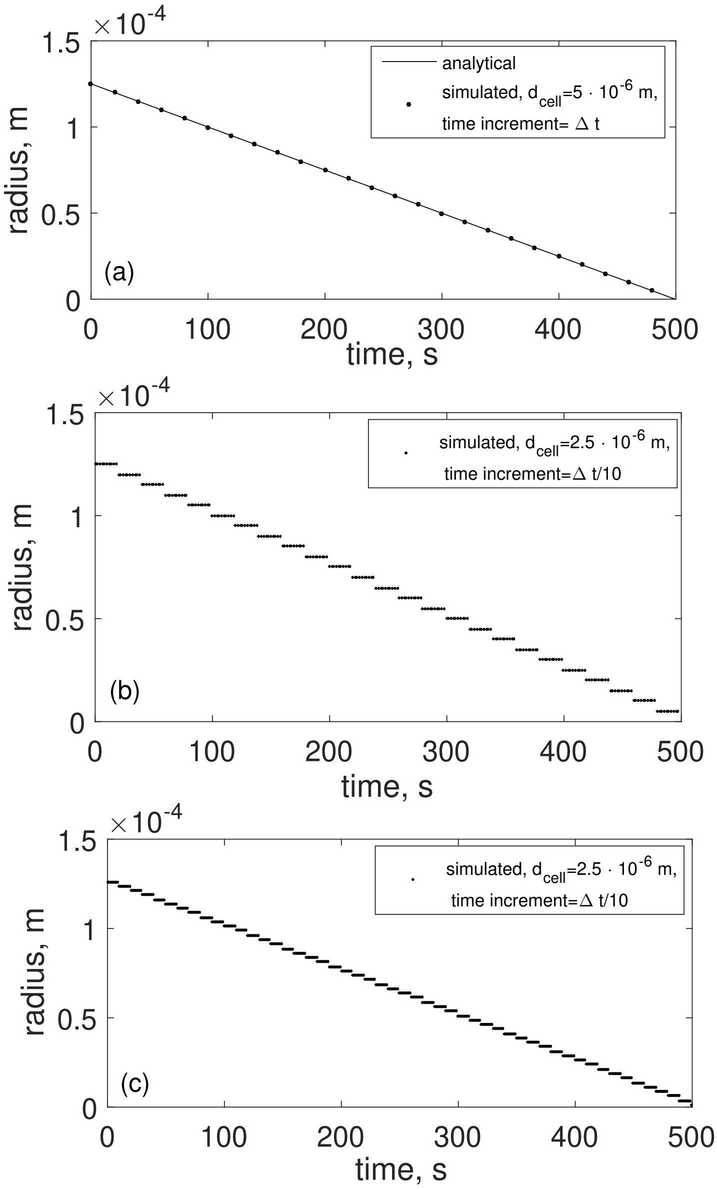


Figure 3.12: The change of the radius a 1D segment with time: a) analytical and simulated with cell size $d_{cell} = 5 \cdot 10^{-6}$ m. b) simulated with the same size cell but with time increment $\Delta t/10$. c) simulated with time increment $\Delta t/10$ and the cell size $d_{cell} = 2.5 \cdot 10^{-6}$ m.

3.3.1.2 The 2D - Dislocation density difference jump

The kinetics in 1D simulations can be predicted very precisely, because the geometry of cells grid does not influence accuracy. However, when the structure becomes 2D or 3D the results lead to a small difference from analytical solution, because of the discreteness of the model.

An investigation into what happens with similar simulations as above for the 2D disk has been made. For simplicity, the Von Neumann neighbourhood will be used: the cell and 4 nearest neighbours.

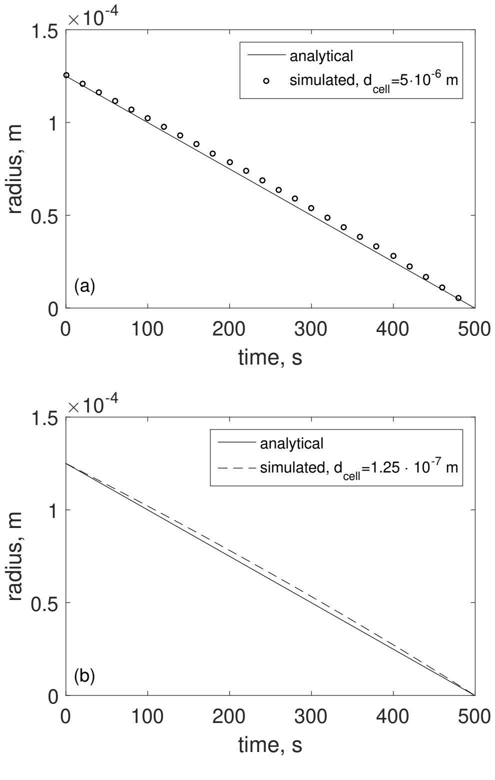


Figure 3.13: The change of the radius of a 2D disk with time calculated analytically compared with simulated with cells size a) $d_{cell} = 5 \cdot 10^{-6}$ m and b) $d_{cell} = 1.25 \cdot 10^{-7}$ m.

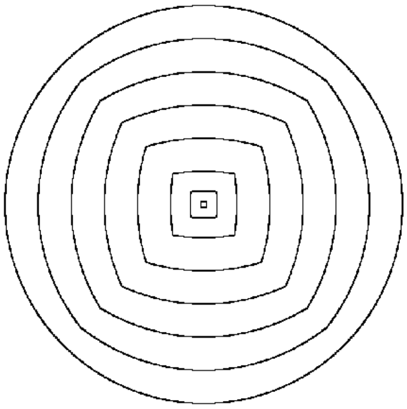


Figure 3.14: Shrinkage of a 2D disk with initial radius of 300 cells after 0, 50, 100, 150, 200, 250, 280, 296 time steps.

Effect of grid anisotropy: The results of simulations with different cell sizes are shown in Fig. 3.13. As can be seen, the results of the 2D simulations do not ideally lie on the analytical line. There is a discrepancy, which can be explained by the anisotropic effects of the grid (see Chapter 5 for details). The initially round shape takes the form of a square (see Fig. 3.14). The similar effect was observed in others works for example by Lan et al. [34]. The decrease of the cell size does not help to rid of this effect. In Fig. 3.13b the cell size was 40 times smaller, which increased the time of calculations, but did not improve results.

Accuracy: As 2D and therefore 3D simulations will lead to some deviations from the theory, we need to estimated the accuracy of simulations. The error was estimated using the area under curves as follows:

$$error = \frac{Area_{sim} - Area_{an}}{A_{an}} \tag{3.45}$$

The errors calculated by this method are 0.0411 and 0.0367 for $d_{cell} =$

$5 \cdot 10^{-6}$ m and $d_{cell} = 1.25 \cdot 10^{-7}$ m respectively. It is also worth noting that the analytical and simulated initial radii are different due to discreteness: the initial analytical radius is $r_0 = 1.25 \cdot 10^{-4}$ m, simulated are $r_0 \approx 1.254 \cdot 10^{-4}$ m and $r_0 \approx 1.25 \cdot 10^{-4}$ m at $d_{cell} = 5 \cdot 10^{-6}$ m and $d_{cell} = 1.25 \cdot 10^{-7}$ m respectively. The plot of errors with the decrease of the cells size or increase of number per initial radius n_{cells} is shown in Fig.3.15. As can be seen from this graph, the limit of error as the cell size approaches zero equals $\lim_{d_{cell} \to 0} error(d_{cell}) = 0.037$.

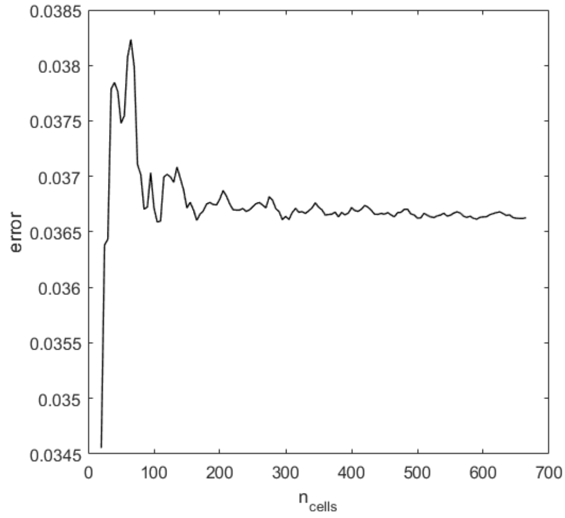


Figure 3.15: The errors of the simulations of the 2D disk shrinkage due to the constant difference in the dislocation density at the different number of cells per initial radius of the grain n_{cells}.

3.3.1.3 Dislocation density difference jump in 3D

The anisotropic effect of the grid observed at 2D is intensified in the 3D simulations. The kinetics of shrinkage of a 2D disk and 3D sphere with the same cell size $d_{cell} = 1.25 \cdot 10^{-6}$ m compared with analytical line are shown in

Fig. 3.16. As can be seen, the simulation in 2D results give the result as being closer to the analytical solution, than in 3D. The error for 2D is 0.037 and for 3D is 0.0646.

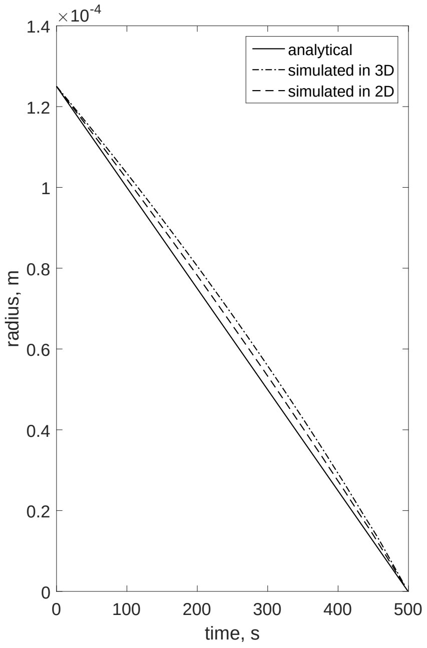


Figure 3.16: The change of the radius of a single grain shrinking with a constant velocity calculated analytically and simulated in 2D and 3D at $d_{cell} = 1.25 \cdot 10^{-6}$ m.

3.3.2 Evolving dislocation density. Time increment for CA

The CA algorithm described in the previous chapter has been adopted for the case of evolution of dislocation density with change of strain. In this case, the driving force changes with time and velocity has its peak at the almost beginning (see Fig.3.8) and decreases with time. Therefore, the time increment can be chosen as $t_{inc} = \frac{d_{cell}}{v_{max}}$, where v_{max} is either the highest velocity which can happen, particularly in the considering case it is a velocity at $t = t_{peak}$ and should be calculated, or v_{max} is the maximum velocity at the current time step. Let us, for future notation, to call first definition fixed t_{inc} and second as updated at each iteration, or shortly updated.

For the case of the updated time increment, the real maximum velocity has been found at each iteration. For the case of the fixed time increment $\Delta t = \frac{d_{cell}}{v_{max}}$, where $v_{max} = M\tau[\rho]_{max}$, $[\rho]_{max} = [\rho](t = t_{max})$ corresponds to the peak of the dislocation density difference (see Fig. 3.8b). In order to find t_{max} the following derivative test has been used:

$$\frac{d[\rho]}{dt} = 0, \tag{3.46}$$

where $[\rho]$ is described by Eq. (3.30):

$$\begin{aligned} &\frac{d}{dt}\left(\frac{1}{k_2^2}\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^d_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2 - \right.\\ &\left. -\frac{1}{k_2^2}\left(e^{-\frac{1}{2}k_2\dot{\varepsilon}(t-t^r_{ini})}(-\sqrt{\rho_{ini}}k_2 + k_1) - k_1\right)^2\right) = 0, \end{aligned} \tag{3.47}$$

and the solution is follows:

$$t_{max} = t^r_{ini} - \frac{2}{k_2\dot{\varepsilon}}\ln\left(-\frac{k_1}{e^{\frac{1}{2}k_2\dot{\varepsilon}(t^d_{ini}-t^r_{ini})}(\sqrt{\rho_{ini}}k_2 - k_1) + \sqrt{\rho_{ini}}k_2 - k_1}\right), \tag{3.48}$$

3.3.2.1 Results of 1D CA simulations

Fig. 3.17 shows the change of the radius $r(t)$ with time simulated by CA with different cell sizes (or number of cells per segment). The graph shows that the decrease of the cell size leads to a smoother solution. The same fact has been observed in the previous subsection (3.3.1.1). However, for the fixed time increment even the small cell size such $d_{cell} = 2.5 \times 10^{-7}$ m does not agree well with an analytical solution (see Fig. 3.17a). The updated t_{inc} Fig. 3.17b shows that simulated solution converges with analytical with bigger number of cells per the initial radius.

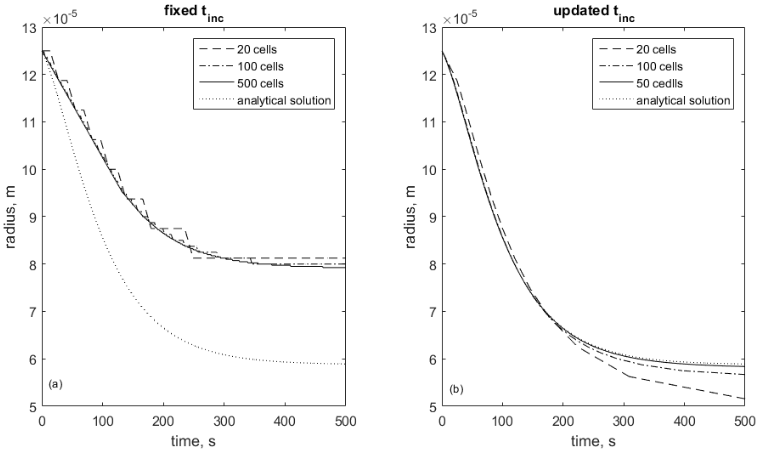


Figure 3.17: The change of the radius of a 1D segment with time at the different cell sizes (or number of cells per initial radius): $d_{cell} = 6.25 \times 10^{-6}$ m (or 20 cells), $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells), $d_{cell} = 2.5 \times 10^{-7}$ m (or 500 cells), for fixed time increment (left) and updated at each iteration (right) compared with analytical solution.

Effect of cell sizes: The error as the percentage discrepancy from analytical solution has been calculated as follows:

$$error = \frac{\Delta r}{r_{an}} = \frac{r_{sim} - r_{an}}{r_{an}}, \tag{3.49}$$

where r_{sim} is the radius of the grain at the end of the simulation and r_{an} is the analytical value at the corresponding time. The simulations were carried out until time $t = 800$ s. The errors calculated as above with the change of the cell sizes for the fixed and the updated time increments are shown in Fig. 3.18.

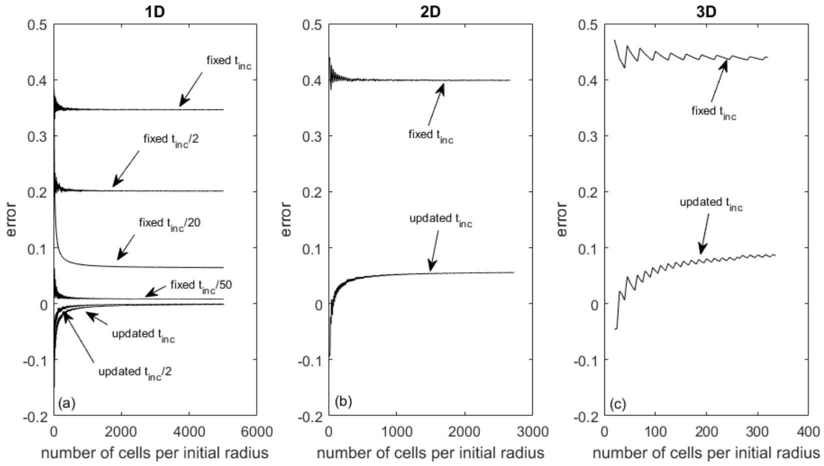


Figure 3.18: The errors of the simulations of a single grain shrinking driven by the evolving dislocation density simulated in 1D (a), 2D (b) and 3D (c) cases at different cell sizes (or number of cells per initial radius) for fixed time increment and updated at each iteration (right).

Effect of time increment: The simulations performed with the fixed and updated time increment t_{inc} reduced by number $n = 1, 2, 3, 4, 10, 50, 100$ are shown in Fig. 3.19. As has been mentioned above, simulations with fixed time increment does not agree well with analytical solution well, Fig. 3.17. However, the reduction of the time increment by integer n lead to convergence of simulation, Fig. 3.19a. Fig. 3.19b shows the similar set of the simulations with the updated time increment. The updated time increment shows good prediction already at the $n = 1$. Further reduction of the time increment shows only slight improvement.

These facts are also shown in Fig. 3.20. As can be seen in this figure, for both cases, fixed and updated t_{inc}, a decrease in the time increment by n significantly reduces error. However, for the updated time increment t_{inc}, the error approaches zero faster than for the fixed time increment. Fig. 3.18a

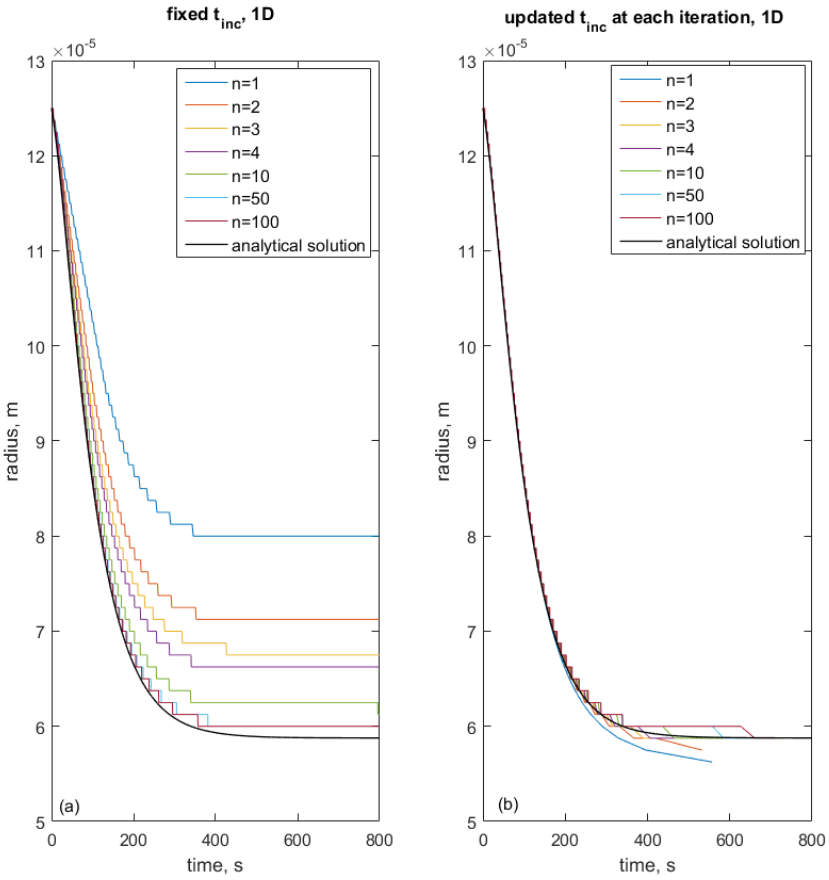


Figure 3.19: The change of the radius of a single grain during shrinkage driven by the evolving dislocation density simulated in 1D segment using the different time increments $t_{inc} = \frac{\Delta t}{n} = \frac{d_{cell}}{n \cdot v_{max}}$ for the fixed (left) and updated (right) at each iteration v_{max}. The cell size is $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells).

compares errors with different time increments with an increase of the number of cells per initial radius n_{cells}. It shows that the limit of the error as a function of n_{cells} decreases with the decrease of time increment t_{inc}.

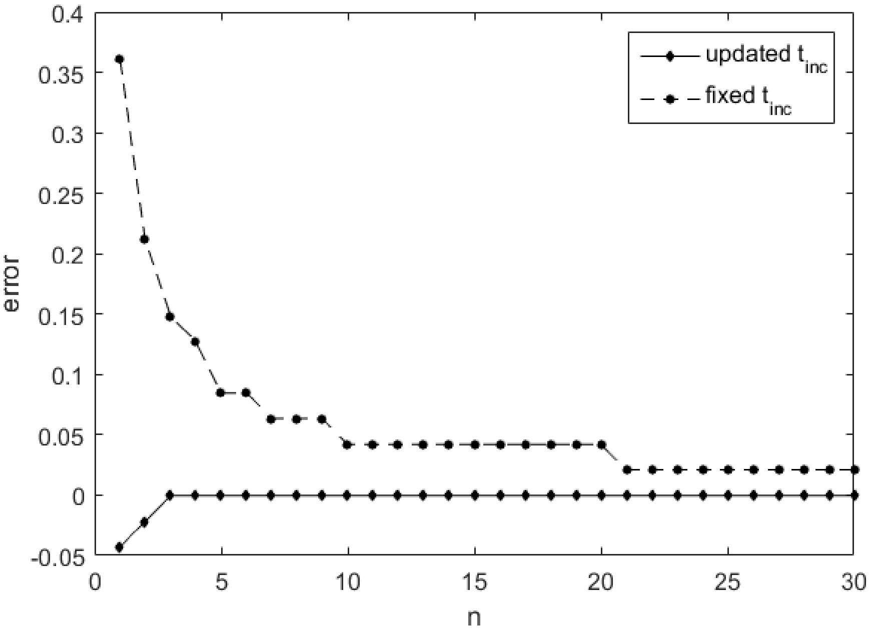


Figure 3.20: The errors of the simulations at the different time increments $t_{inc} = \frac{\Delta t}{n} = \frac{d_{cell}}{n \cdot v_{max}}$ for the fixed v_{max} and updated at each iteration. The cell size is $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells).

3.3.2.2 Results of 2D and 3D CA simulations

The same series of simulations as for 1D has been implemented for 2D disk. The change of the radius $r(t)$ with time is compared with 1D simulations for the fixed and the updated time increments in Fig. 3.21. Simulations have been performed using the cell size $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells). In the case of the fixed t_{inc} the simulations a 2D have bigger discrepancy with analytical solution than for 1D. Fig. 3.18b shows that the errors in 2D are shifted up in comparison with 1D. The error approaches value $error = 0.055$ with the decrease of the cell size. The errors for 3D are also greater than in 2D. Therefore, similarly to the results in 3.3.1.1, the grid anisotropy in 2D and

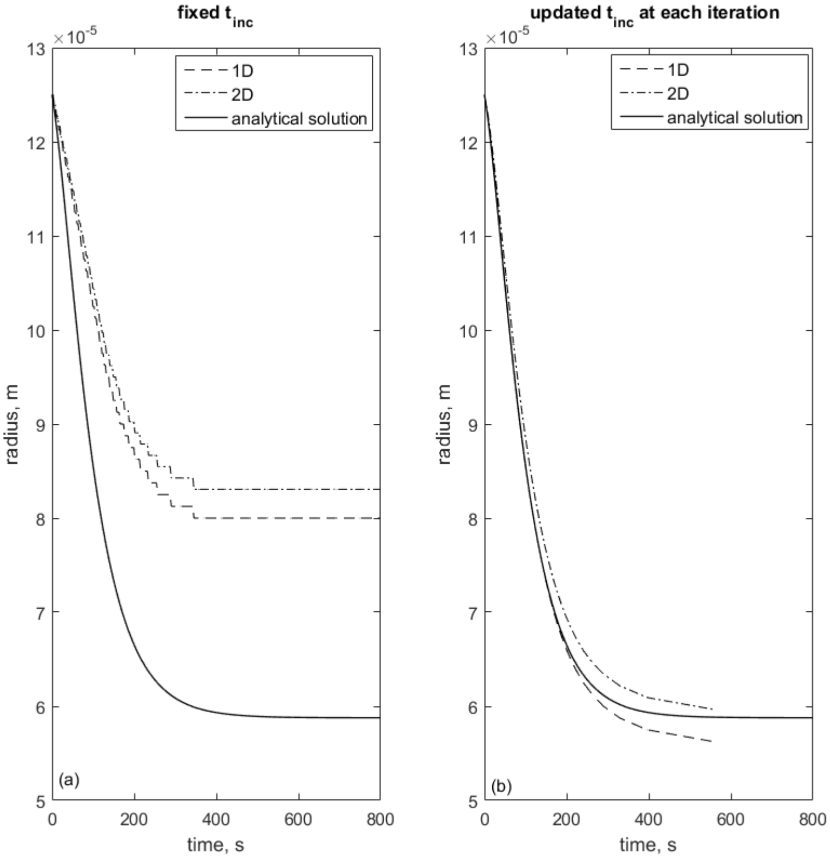


Figure 3.21: The change of the radius of a single grain during shrinkage driven by the evolving dislocation density simulated in 1D and 2D with different types of the time increment: fixed (left) and updated (right) compared with analytical solution. The cell size is $d_{cell} = 1.25 \times 10^{-6}$ m

3D simulations caused deviations from analytical solution.

3.3.3 Simulation of a single grain shrinkage driven by curvature

The grain boundary is moving with a velocity $v_c = M\gamma k$, where M is the grain boundary mobility, γ is the grain boundary energy and k is the curvature.

The curvature k can be calculated by various of methods, as described in the literature. The most straightforward is the counting cell method which was suggested by Kremeyer [115] and has successfully been applied to CA simulations by Janssens [1], Lan [34], Hallberg [41] and others.

A kink template (see Fig. 3.22a) is mapped over the cell. The template assigns "1" to all positions which belonging to the same grain as the center one and zero belonging to others. Furthermore, the curvature is calculated by the following formula [34]:

$$k = \frac{A}{a}\frac{Kink - N_i}{N+1}, \tag{3.50}$$

where N_i is the number of cells within the neighbourhood belonging to the grain i, i.e. the same grain as the central cell, $Kink$ is the number of cells within a disk template belonging to the grain i in case of flat interface ($k = 0$), A is the coefficient, a is the cell size and N is the number of neighbours within neighbourhood. In some articles, it has been found wrong interpretation of parameter A, for example [113], where A was called "topological coefficient", which is in fact a scaling parameter. The parameter A in most papers (Zheng 2008 [106], Zheng et al. 2012, [113]) was chosen equal 1.28, the earliest [34], but no derivation has been shown. It needs to be derived by calibration, as has been done by Han et al. [37] for example. Han ([37] used following formula [117], [118]) for the curvature which has been applied for disk templates with a bigger radius:

$$k \simeq \frac{3\pi}{b}\left(\frac{A}{A_{tot}} - \frac{1}{2}\right), \tag{3.51}$$

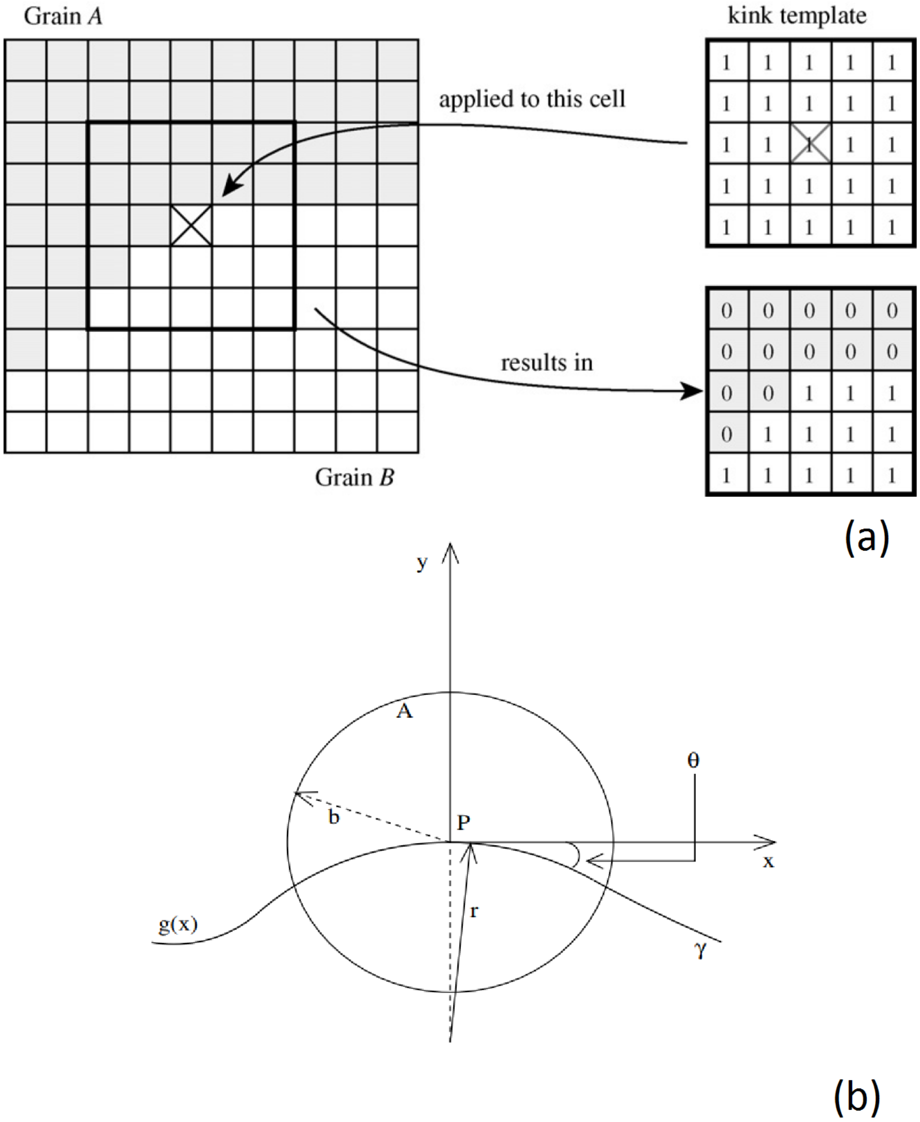
where b is the radius of a disk template, A_{tot} is the total area of the discrete template, A is the area outside the line (Fig. 3.22b).

However, in this particular article the parameter A has been calibrated at one particular initial radius and has not been checked against grain shrinkage with the different radii. The same practice has been implemented by Hallberg [41], [42].

After the curvature has been calculated, the central cell should change its state if the curvature is positive $k > 0$ [1]. The main difference between this approach and a technique used in the present research is that now the curvature of the neighbours is calculated instead of the central cell. The algorithm scheme is shown in Fig. 3.10. This modification helps to avoid uncertainty in the triple junctions (see Appendix A).

3.3.3.1 Curvature driving force in 2D - Effect of the cell size and time increment

The driving force caused by curvature cannot be applied in 1D dimensions, so the simulations for the curvature are performed only for the disk and sphere. Let us consider Eq.(3.50) for the 2D case: the parameter N defines used neighbourhood, a is a cell size, N_i is defined by cells allocation, so the two parameters left $Kink$ and A can be considered as free which can be varied. In fact, A will not change the form of the curve, but it can be considered as a scaling parameter (see Fig.3.23), and $Kink$ corresponds for the shape of curve. One can obtain the fitted simulation model to the analytical by tuning parameters $Kink$ and A.



1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

0	0	0	0	0
0	0	0	0	0
0	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Figure 3.22: A scheme of a curvature calculation: a) a disk template 5x5 template applied to the cell [1], b) curvature depends linearly on the area above the curve γ [117].

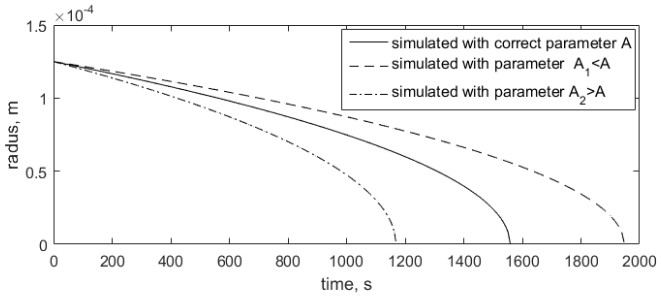


Figure 3.23: Simulation of a single grain shrinkage driven by curvature with different parameters A. The kinetics is slower than analytical at the parameter $A_1 < A$ and faster at the parameter $A_2 > A$.

The commonly used values for the parameters in Eq.(3.50) are [34]: $Kink = 15$, $A = 1.28$ and $N = 24$ (in the case of a Moore neighbourhood in 2D with the first- and second-nearest neighbours). However, considering the fact that the algorithm has been modified, the new parameters A and $Kink$ should be re-estimated. In order to see how different values of $Kink$ influence results, the simulations have been run with different values of $Kink$, the curves were scaled to the analytical solution and the shapes have been compared. The results of the simulations for $d_{cell} = 2.5 \cdot 10^{-6}$ m are shown in Fig. 3.24. The difference is not visible to the naked eye, hence further the calculated error will be used for comparison.

The critical condition is that once the parameters A and $Kink$ have been established, the same needs to be used in the following chapter for the case of the coupled forces. Ideally, the same parameters should be used at the different time increments (smaller than minimal) and different cell sizes.

Effect of time increment: As has been discussed previously, there are two ways in which to define the time increment within the algorithm: 1) to use the minimal time increment: $t_{inc} = \frac{d_{cell}}{v_{max}}$, where $v_{max} = M\gamma k_{max}$, the maximum

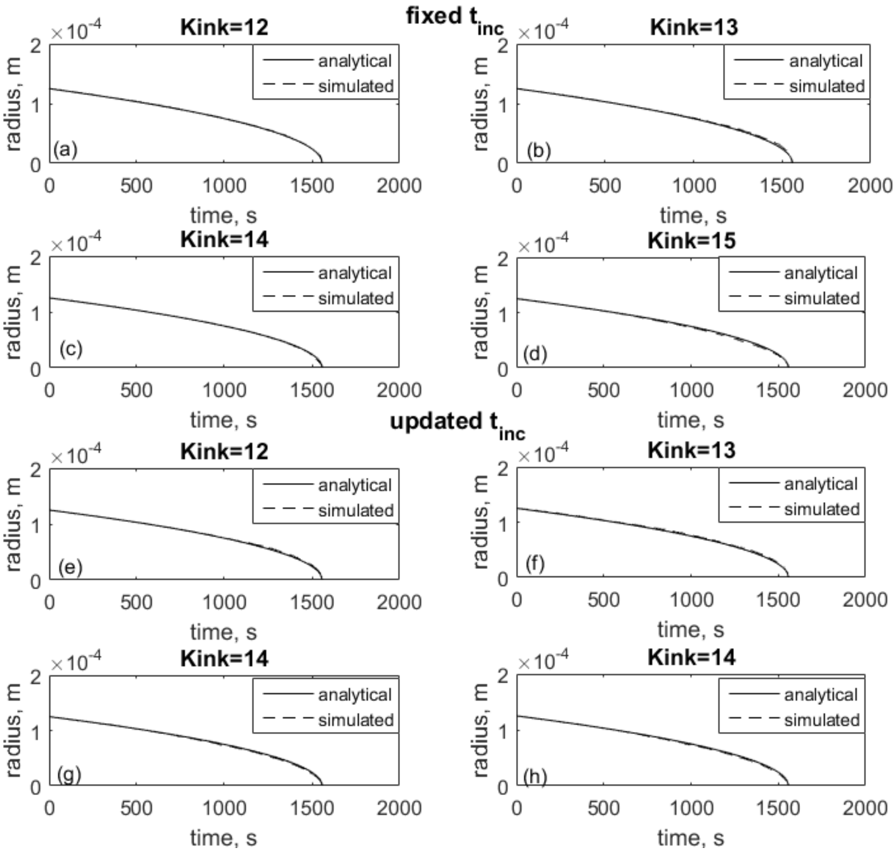


Figure 3.24: The change of the radius of a single grain during shrinkage driven by curvature, simulated at the different values of $Kink$ and scaled to the analytical solution, for the fixed (a)-(d) and updated (e)-(h) t_{inc}. The cell size is $d_{cell} = 1.25 \cdot 10^{-6}$ m or 100 cells.

curvature k_{max} is happening at $N_i = 0$ (see Fig. 3.25); 2) to find the real maximum velocity on the domain area and calculate the time increment at each iteration, that for example has been realized by Hallberg [42].

However, for both methods, no analysis has been shown in the literature as to what would happen if the time increments were to be reduced by integer number n: $t_{inc} = \frac{d_{cell}}{n \cdot v_{max}}$. For the cell size $d_{cell} = 5 \cdot 10^{-6}$ m, $Kink = 12$ a series of simulations has been done, for the fixed time increment and updated

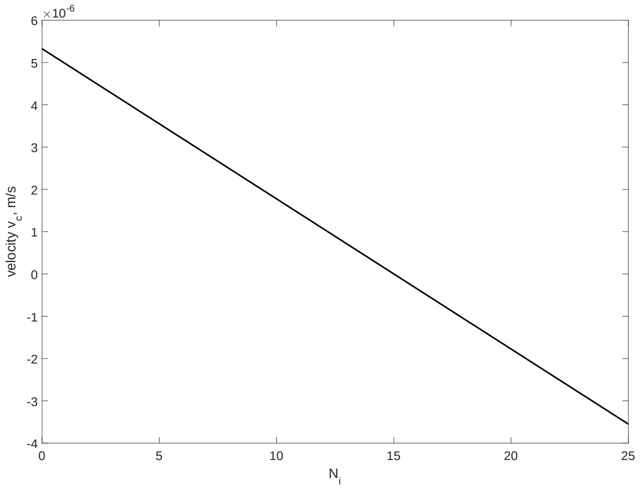


Figure 3.25: The grain boundary velocity of an individual cell driven by curvature with the change of the number N_i according to Eq. (3.50).

at each iteration at different values n. The results are shown in Fig. 3.26. In order to make the starting point of all cases equal zero the values of A are shifted by first point $A_i - A_1$. As can be seen from this figure, the parameter A initially decreases with n and then continues to fluctuate. In the case of the fixed time increment t_{inc} the $Kink = 15$ has the highest amplitude of fluctuations, but at the same time the smallest slope that means A is more consistent. The usage of the fixed time increment also lead to the more consistent A for others values of $Kink$ in comparison with the updated time increment. The amplitude of fluctuations in the case of the updated time increment is smaller than for the fixed time increment.

The corresponding errors are shown in Fig. 3.27. As can be seen in this figure, the errors do not decrease with n, so there is no need to use small time increment, as it will not improve accuracy of simulations.

The fact that the proposed algorithm in [42] does not lead to the same

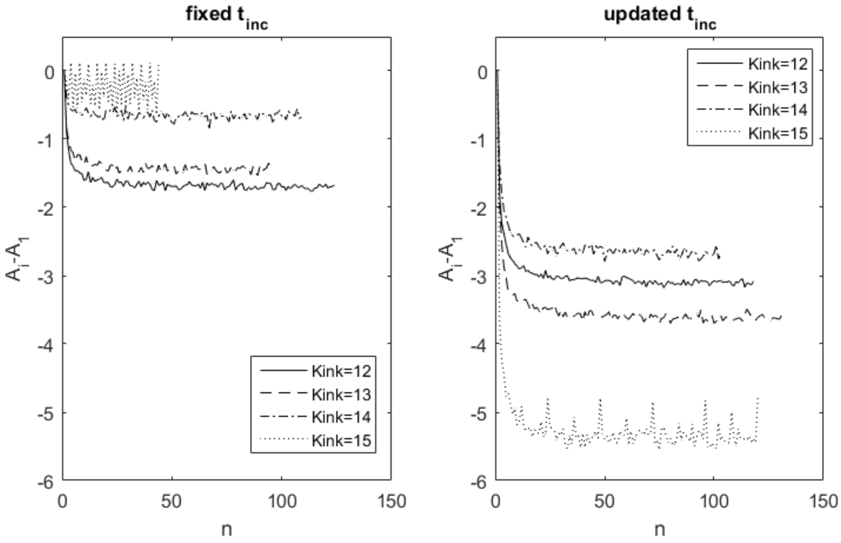


Figure 3.26: The change of the parameter A shifted by A_1(at $n = 1$) with the change of the time increment $\Delta t = \frac{d_{cell}}{n \cdot v_{max}}$ for the fixed (left) and updated (right) time increment. The cell size is $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells).

results with the different n ($t_{inc} = \frac{d_{cell}}{n \cdot v_{max}}$) can be explained as follows: in the case of the dislocation density, the maximum velocity is constant, hence the change of the time increment does not influence the results, flat segments appear (see Fig. 3.12), but the shrinkage time does not shift. In the case of the curvature driven shrinkage, there are no such flat segments, so at each iteration, regardless of how small is the iteration, the cell structure changes. This is happening because the maximum velocity (and therefore the time increment $t_{inc} = \frac{d_{cell}}{n \cdot v_{max}}$) is not constant. The velocity $v_c = M\gamma k$ depends on the number N_i Eq. (3.50) and the maximum curvature occurs at minimal $N_i = 0$ (see Fig. 3.25). However, the real minimal N_i in the simulations changes with every iteration and oscillates between 9 and 12, (see Fig. 3.28).

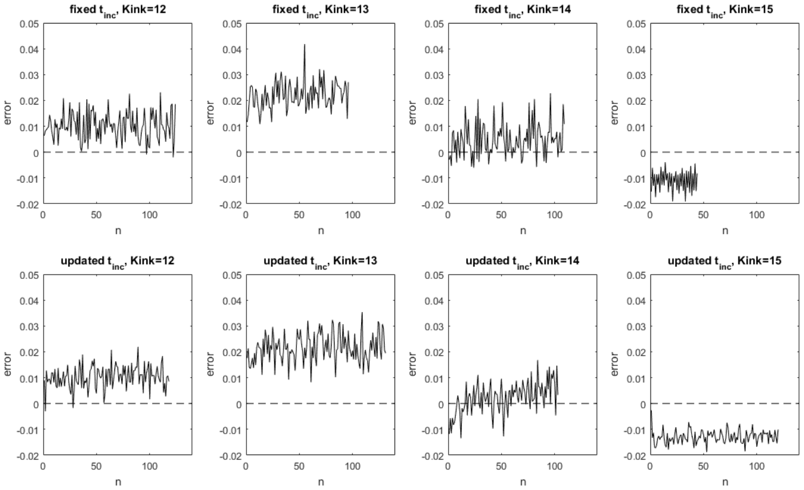


Figure 3.27: The errors of the scaled simulations performed with different $Kink$ and time increments $\Delta t = \frac{d_{cell}}{n \cdot cell v_{max}}$, fixed (top) and updated time increments at each iteration (bottom). The cell size is $d_{cell} = 1.25 \times 10^{-6}$ m (or 100 cells).

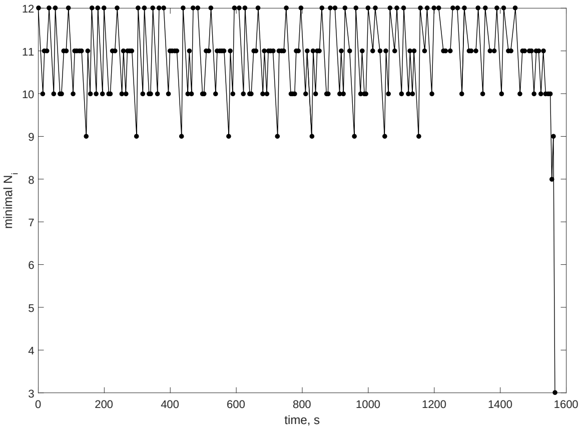


Figure 3.28: An example of the change of the minimal N_i with time.

Effect of cell sizes: In contrast to the simulations of the dislocation density jump (section 3.3.1.1), there is a significant dependence of the parameter A and $Kink$ on the cell size in the case of the curvature driven grain growth simulations. In multigrain simulations grains are changing in size, so the number of cells per grain are changing as well. Therefore, it is important to have a model for the curvature driven growth which works consistently at the different numbers of cells per grain (or the cell size for the single grain problem).

In order to check how the cell size influences the value of the parameter A and errors, a series of simulations has been implemented with the initial grain radius is $r_0 = 1.25 \cdot 10^{-4}$ m and different values of $Kink = 12..15$. The errors in the simulations calculated by Eq. (3.45) are shown in Fig. 3.29. As can be seen from this figure all values of $Kink$ shows big oscillations of error at the big cell sizes. With a decrease of cell size the error stabilizes and the best error convergence has been observed at $Kink = 14$ and fixed t_{inc}.

Another graph Fig. 3.30 shows changes in A with the cell size for simulations with different values of $Kink$. A should be consistent, as grains change size in multigrain simulations and consist of different number of cells. As can be seen from interpolated lines shown in Fig. 3.30, the smallest slopes are in the case of fixed time increment with $Kink = 12$ (the slope equals to -7×10^{-5}). The value of A at $Kink = 15$ increases with the cell size, while at others value of $Kink$ it tends to remain constant.

This information can be interpreted as follows: the model will work consistently in the certain range of cell sizes for values of $Kink = 12, 13$ or 14, but not for 15. Any change in the cell size will require to recalculate A which will make the multigrain calculations inaccurate. The preferable value for $Kink$ is 14, because of the best error convergence. Hence, these results also put

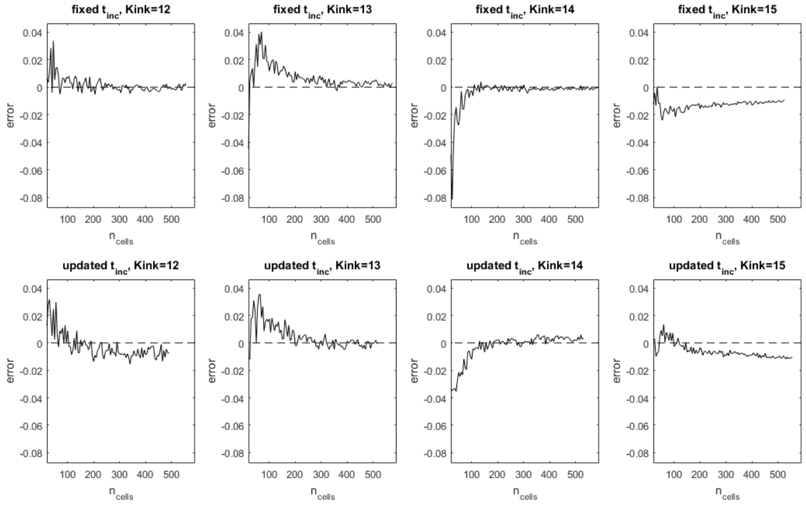


Figure 3.29: The errors of the scaled simulations performed with the different $Kink$ and types of CAS: fixed (top) and updated t_{inc}(bottom) with the change of the cell sizes.

restrictions on the minimal space resolution of the model.

3.3.3.2 Curvature driven force in 3D

In 3D, the parameters for Eq. (3.50) will be: $N = 124$, and possible values for $Kink$ are bigger than for 2D. To find the optimal series of simulations similar for 2D are required. However, the simulations of shrinkage driven by curvature require more total iterations than that by the dislocation density jump, and the simulations in 3D are more time consuming. Therefore, this analysis has not been performed and is advised for future work.

3.3.4 Number of iterations

The number of iterations is of particular interest. Although the results become more accurate, the bigger number of the calculation steps requires greater

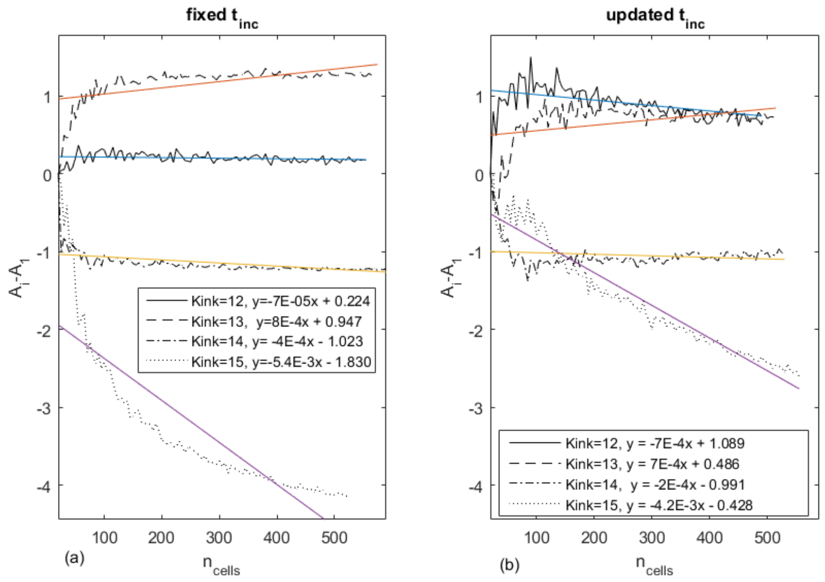


Figure 3.30: The change of the parameter A_i shifted by A_1 calculated at the different values of $Kink$ and types of CAS: fixed (a) and updated (b), with the change of the cell sizes. The cell size for A_1 is $d_{cell} = 6.25 \times 10^{-6}$ m or 20 cells

computation time. As can be seen from this graph, the number of iterations n grows non-linearly with the decrease of the cell size. The quickest growth is observed in the simulations with the updated time increment at $Kink = 12$ and $Kink = 13$, the slowest is happening in the case of the fixed CAS and $Kink = 15$.

3.4 The probabilistic version of developed CA technique

The probabilistic version of a developed CA technique based on Monte Carlo step instead of calculating distances has been developed. If the random num-

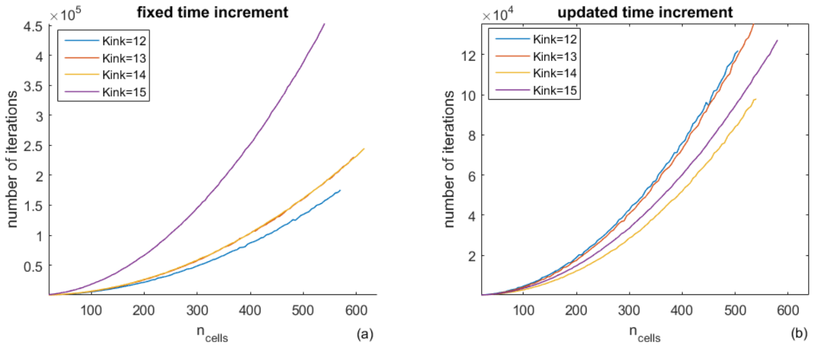


Figure 3.31: The growth of the number of iterations for the simulations of a single grain shrinkage driven by curvature at the different values of $Kink$ and types of CAS: fixed (a) and updated(b) with the change the cell sizes.

ber is less than the velocity of the moving the neighbour related to the maximum velocity $rand < \frac{v_{neighb}}{v_{max}}$ then the cell changes its state to the neighbours state. The deterministic CA rule implies that cell changes state to the state of neighbour with distance variable greater than cell size. The probabilistic CA rule implies that the cell changes its state to neighbours state with probability equal to the value $\frac{v_{neighb}}{v_{max}}$.

3.4.1 Simulation of a single grain shrinkage driven by curvature using developed PCA algorithm

The grain shrinkage with the fixed CAS is shown in Fig. 3.32. The disk shrinks only at values of $Kink = 5..15$. Fig. 3.33 demonstrates the same simulations by the deterministic CA using the updated t_{inc}. As can be seen, the disk with initial cell size $r_0 = 20$ cells shrinks with any value of $Kink = 1..15$.

The probabilistic CA with the same initial radius, $r_0 = 20$ cells, leads to the different results of simulation with the same parameters. Fig. 3.34 shows 10 simulations for the probabilistic CA simulations using the fixed time increment

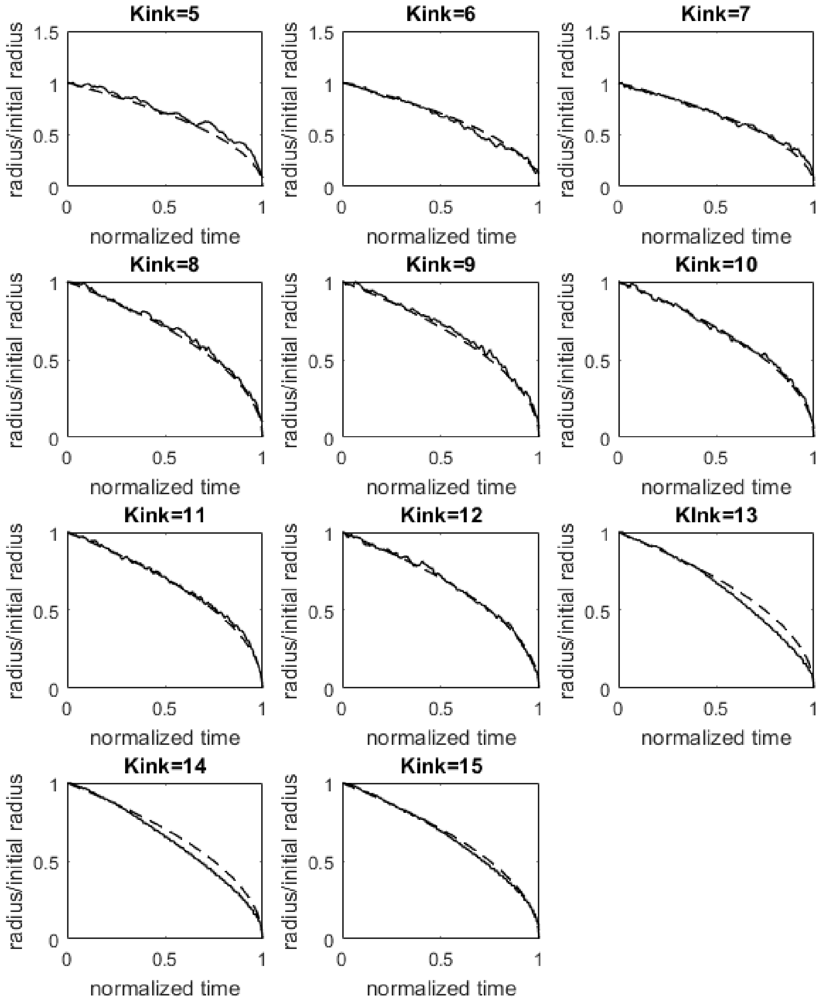


Figure 3.32: The change of the radius of a single grain during curvature driven shrinkage simulated with DCA at the values of $Kink = 5..15$ and scaled to the analytical solution, for the fixed t_{inc}.

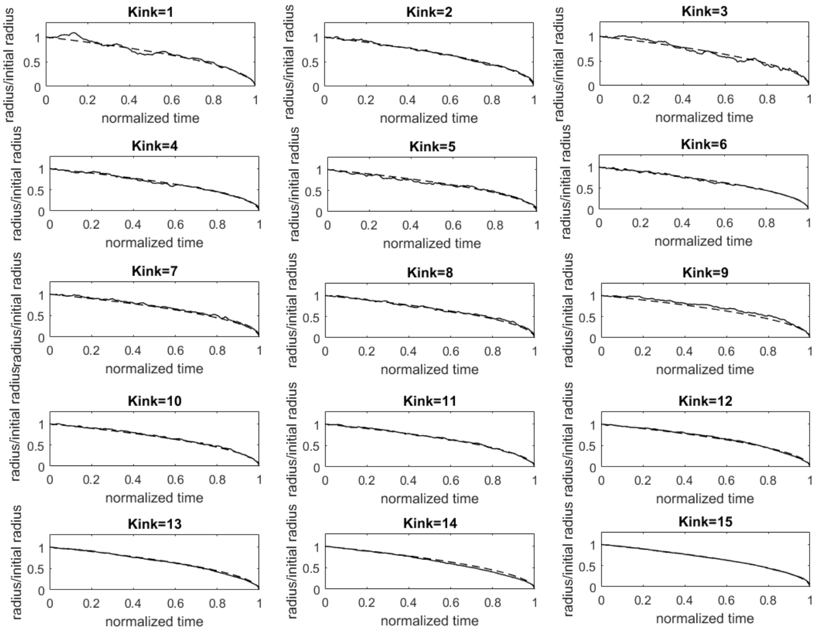


Figure 3.33: The change of the radius of a single grain during curvature driven shrinkage simulated with DCA at the values of $Kink = 5..15$ and scaled to the analytical solution, for the updated t_{inc}.

and Fig. 3.35, Fig. 3.36 – updated.

The probabilistic CA with values of $Kink = 12..15$ has been studied similarly to the deterministic CA described in the previous sections. The mean parameter A (see Fig. 3.37) and errors (see Fig. 3.38 were estimated by running 10 simulations with the same parameters. The behaviour of the parameter A and errors with an increase of the number of cells per initial radius is similar to the case of the deterministic CA.

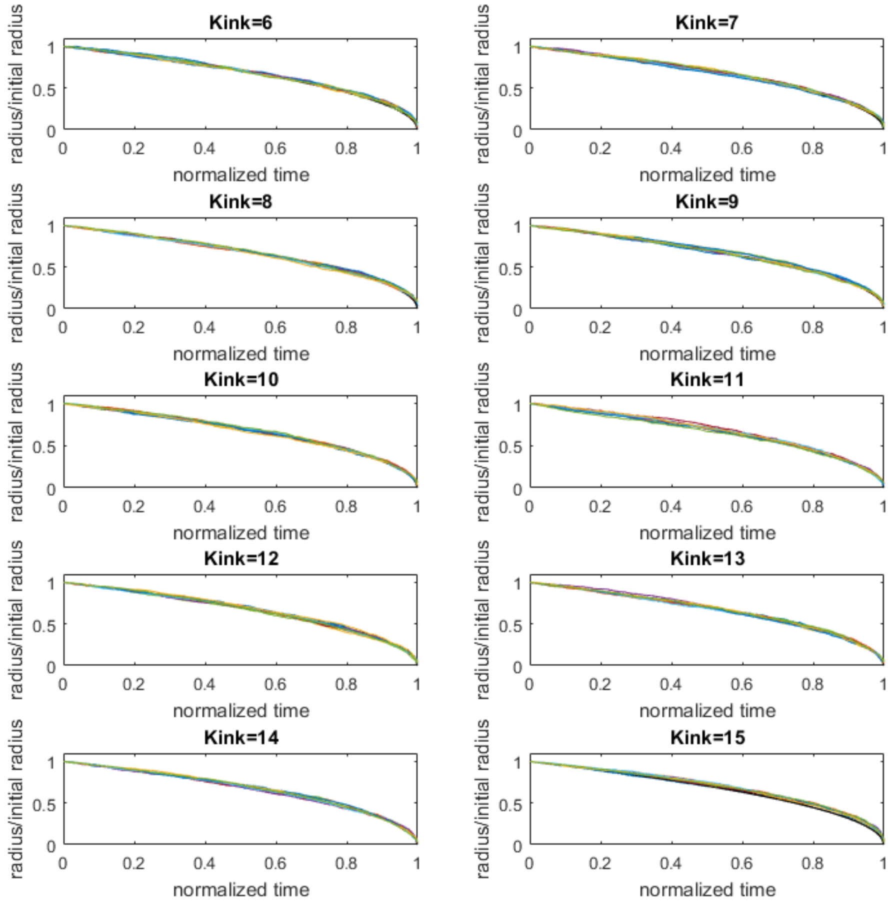


Figure 3.34: The change of the radius of a single grain curvature during driven shrinkage simulated with PCA at the values of $Kink = 6..15$ and scaled to the analytical solution, for the fixed t_{inc}.

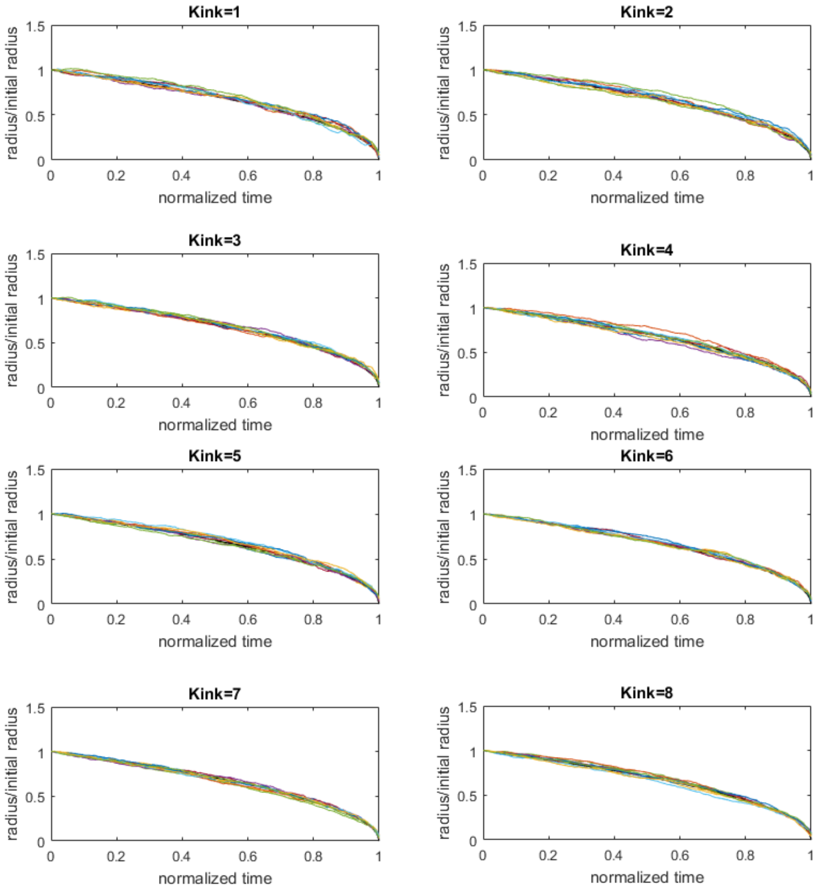


Figure 3.35: The change of the radius of a single grain during curvature driven shrinkage simulated with PCA at the values of $Kink = 1..8$ and scaled to the analytical solution, for the updated t_{inc}.

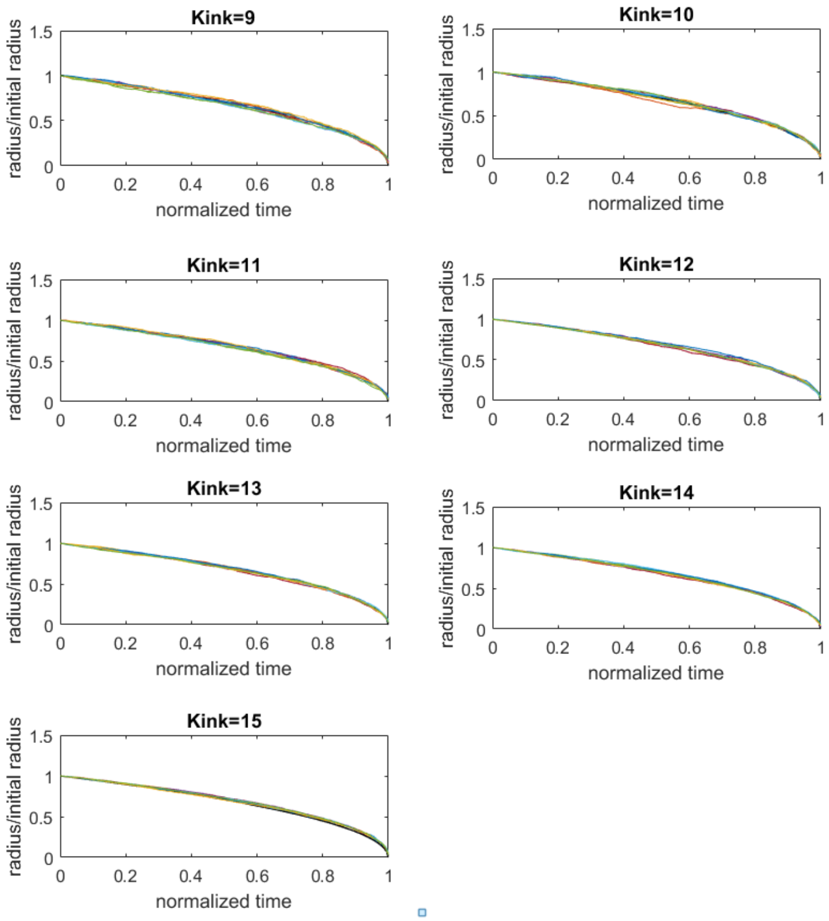


Figure 3.36: The change of the radius of a single grain during curvature driven shrinkage simulated with PCA at the values of $Kink = 9..15$ and scaled to the analytical solution, for the updated t_{inc}.

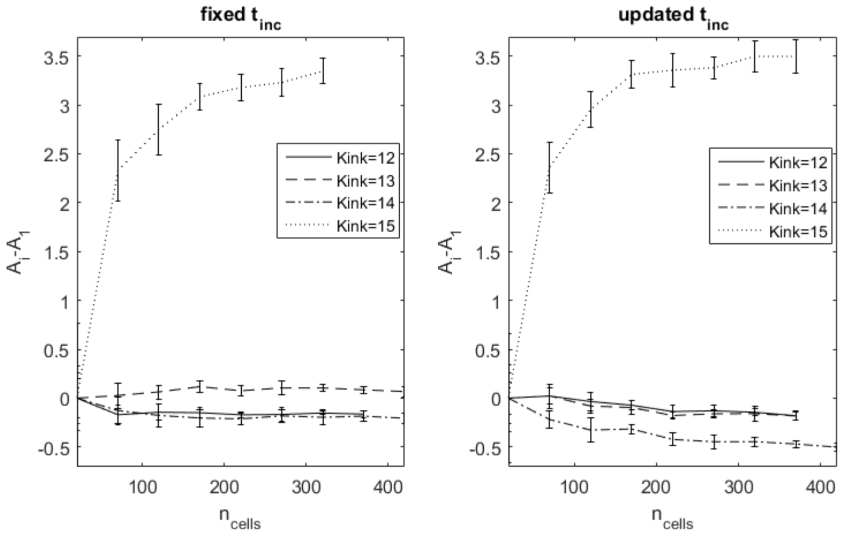


Figure 3.37: The mean parameters A_i shifted by A_1 over 10 different simulations calculated at the different values of $Kink$ and CAS with the change of the cell sizes. Bars show standard deviation. The cell size for A_1 is $d_{cell} = 6.25 \times 10^{-6}$ m, $r_0 = 20$ cells

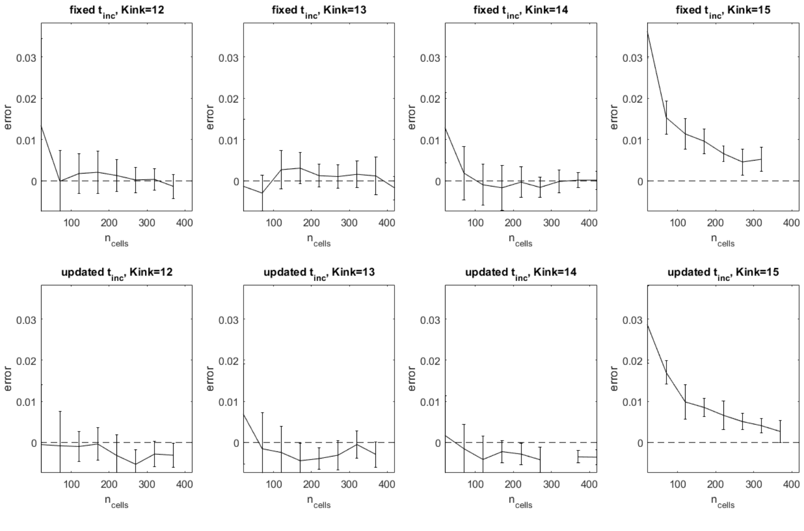


Figure 3.38: The mean errors of the scaled simulations performed with the different $Kink$ and types of CAS with the change of the cell sizes. Bars show standard deviation.

3.5 Summary

Single grain simulations allows to validate CA model with theory, where a dynamics of the grain growth is known. Once algorithm has been validated with analytical solution, it can be used for multigrain stricture. The main difference between single grain and miltigrain simulations is the presence of triple junctions in the last one. A newly developed CA technique computes boundary movements in the triple junctions inherently. Explicit CA simulation is a powerful instrument for simulation of a complex systems with large number of components, such as grains during recrystallization and grain growth. The knowledge of driving forces which is a considered as a theory does not allow to predict the evolution of structure. However validated on a single grain CA approach solves governing equations discretely, which makes possible to take into account local parameters such as impurities, texture, etc.

Three different ordinary differential equations describing the shrinkage of a single grain, have been solved by the proposed real time CA. For the dislocation density jump (first and second cases) the most accurate are simulations in 1D, because calculations in the higher dimensions increase the discrepancy from the analytical due to the grid anisotropy. For evolving dislocation density jump and curvature driven shrinkage two different approaches of choosing the time increment have been considered, fixed and updated at each iteration, which both have different advantages and disadvantages. It has been shown that CA with the updated time increment proved to be more accurate for the case of the evolving the dislocation density jump. At the same time, the reduction of the fixed time increment significantly improves accuracy.

The third equation described the curvature driven shrinkage, where the variable CA parameters are the scale parameter A, and $Kink$. Simulations have been carried out for the values of $Kink = 12, 13, 14, 15$ using CA with the fixed and updated time increment. It has been shown that the decrease of time increment by integer n does not influence the average error of the scaled simulations, but changes parameter A. The values of $Kink$ influence the parameter A and the most consistent A at different cell sizes is $Kink = 12$ at the fixed time increment.

For all three equations, the influence of the cell size has been also studied. It has been shown that finer initial structure will have more accurate results, than the coarse structure. The decrease of time increment increases convergence (not true for curvature).

Chapter 4

Modelling of a Single Grain - The Coupled Driving Forces

The differential equation considered in the previous chapter describes grain boundary movement under different driving forces, dislocation density jump and curvature, separately. However, in reality, these two forces are simultaneous and concurrent. During dynamic recrystallization the newly formed nuclei with a low dislocation density are growing into a deformed matrix with a high dislocation density, but at the same time is influenced by the curvature driven force, which tends to shrink it.

There are not many works devoted to study of coupled forces: Lan [34], Han [37], Zheng et al. [106], Zheng et al. [113] studied only curvature driving grain shrinkage. The coupled driven forces have been considered by Kugler [39], but in his paper the technique to calculate curvature has not been specified. Ding [14] also simulated the grain growth driven by coupled forces, but their methods were based on the calculation of the radii of the grains instead of curvature. The same method has been found in works Xiao et al. [119], Yazdipour et al. [120]. Hallberg with co-workers [42], [41] described CA for the simulation of the single grain shrinkage due to evolving dislocation density and curvature driven forces. However, the calibration has been made with one particular radius, and no investigation on the consistency of its parameters at grains with the different radii.

Therefore, in the following chapter, CA will be applied to solve the combined differential equations, where forces are acting simultaneously.

4.1 Analytical solutions

4.1.1 Single grain shrinkage due to constant dislocation density jump and curvature

Combining two contributions, curvature and dislocation density jump driving forces, into the driving pressure, we propose to derive the grain boundary velocity, which consists of two parts:

$$v = v_C + v_D = M(p_C + p_D) \tag{4.1}$$

The kinetics of the grain radius can be written as follows:

$$\frac{dr}{dt} = M\left(\tau[\rho] - 2\frac{\gamma}{r}\right) \tag{4.2}$$

This is a 1st-order non-linear differential equation with initial conditions, $r(0) = r_0$. Dividing both sides by $M(\tau[\rho] - \frac{2\gamma}{r})$ yields:

$$\frac{\frac{dr}{dt}}{M(\tau[\rho] - \frac{2\gamma}{r})} = 1, \tag{4.3}$$

Integrating both sides with respect to t leads to:

$$\int \frac{\frac{dr}{dt}}{M(\tau[\rho] - \frac{2\gamma}{r})} dt = \int 1 dt, \tag{4.4}$$

For the left side integral, substitute $u = r(t)$ and $du = \frac{dr}{dt} dt$;

$$\frac{1}{M} \int \frac{1}{(\tau[\rho] - \frac{2\gamma}{u})} du = \int dt, \tag{4.5}$$

$$\frac{1}{M\tau[\rho]} \int \frac{u}{\left(u - \frac{2\gamma}{\tau[\rho]}\right)} du = \int dt, \tag{4.6}$$

$$\frac{1}{M\tau[\rho]} \int \frac{u + u - \frac{2\gamma}{\tau[\rho]} - u + \frac{2\gamma}{\tau[\rho]}}{\left(u - \frac{2\gamma}{\tau[\rho]}\right)} du = \int dt, \tag{4.7}$$

$$\frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \int \frac{1}{u - \frac{2\gamma}{\tau[\rho]}} du + \int du \right) = \int dt, \tag{4.8}$$

$$\frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \ln \left(u - \frac{2\gamma}{\tau[\rho]} \right) + u \right) = t + C, \tag{4.9}$$

And going back to r:

$$\frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \ln \left(r - \frac{2\gamma}{\tau[\rho]} \right) + r \right) = t + C, \tag{4.10}$$

where C is the constant and can be found from initial condition $r(0) = r_0$:

$$C = \frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \ln \left(r_0 - \frac{2\gamma}{\tau[\rho]} \right) + r_0 \right), \tag{4.11}$$

$$\frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \ln \left(r - \frac{2\gamma}{\tau[\rho]} \right) + r \right) = t + \frac{1}{M\tau[\rho]} \left(\frac{2\gamma}{\tau[\rho]} \ln \left(r_0 - \frac{2\gamma}{\tau[\rho]} \right) + r_0 \right), \tag{4.12}$$

$$\frac{2\gamma}{\tau[\rho]}\ln\left(r-\frac{2\gamma}{\tau[\rho]}\right)+r=M\tau[\rho]t+\frac{2\gamma}{\tau[\rho]}\ln\left(r_0-\frac{2\gamma}{\tau[\rho]}\right)+r_0, \tag{4.13}$$

$$\frac{2\gamma}{\tau[\rho]}\left(\ln\left(r-\frac{2\gamma}{\tau[\rho]}\right)-\ln\left(r_0-\frac{2\gamma}{\tau[\rho]}\right)\right)+r-r_0=M\tau[\rho]t, \tag{4.14}$$

Which leads to the following expression:

$$\frac{2\gamma}{\tau[\rho]}\ln\frac{r-\frac{2\gamma}{\tau[\rho]}}{r_0-\frac{2\gamma}{\tau[\rho]}}+(r-r_0)=M\tau[\rho]t, \tag{4.15}$$

Fig. 4.1 shows the kinetics of the radius obtained via solving Eq. (3.15), Eq. (3.42) and Eq. (4.15). As can bee seen, the sum of two forces leads to more rapid shrinkage than in the case of each force separately. The parameters used in the simulations are represented in Table 4.1.

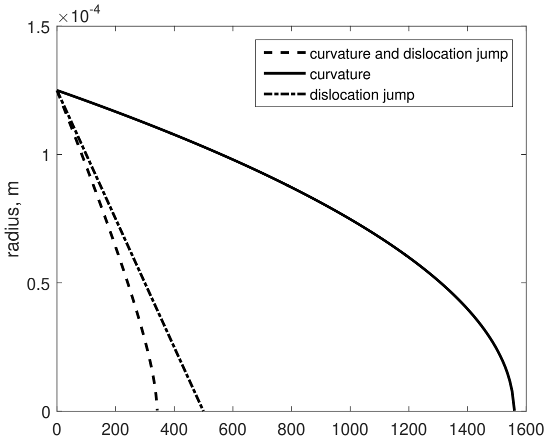


Figure 4.1: The change of the radius of a single grain during shrinkage driven by three different forces: constant dislocation density jump, curvature and combination of them acting simultaneously. The corresponding times of the full shrinkage are ~ 342 s, ~ 500 s and ~ 1559 s.

Parameter	Value	Description
r_0	125 μm	Initial radius
b	$2.56 \cdot 10^{-10}$ m	Magnitude of Burgers vector
k	$1.38 \cdot 10^{-23}$ JK^{-1}	Boltzmann constant
R	8.31 Jmol^{-1}K^{-1}	Gas constant
T	$750K$	Temperature
Q_b	1) $104 \cdot 10^3$ J/mol [14] 2) $110 \cdot 10^3$ J/mol [39]	Boundary-diffusion activation energy
γ	0.625 Jm^2	Grain boundary energy
δD_{ob}	$5 \cdot 10^{-15}$ [14], $7.5 \cdot 10^{-15}$ [39] m^{3}s^{-1}	boundary self-diffusion multiplied by grain boundary thickness
μ_0	$4.21 \times 10^9 \ \frac{N}{m^2}$	Shear modulus at 300 K
$\frac{T_M}{\mu_0} \frac{d\mu}{dT}$	-0.54	Temperature dependence of modulus
T_m	1356 K	Melting temperature
α	0.5	Dislocation interaction coefficient
$[\rho]$	$5.51 \cdot 10^{13} m^{-2}$	Dislocation density difference inside and outside of grain

Table 4.1: Single grain parameters.

4.1.2 Single grain shrinkage due to evolving dislocation density jump and curvature

The combination of the two driving forces, curvature and constant dislocation density jump, has been discussed in the previous subsection (see section 4.1.1). It is proposed to find the analytical solution taking into account the time dependence of the dislocation density. The equation to solve is as follows:

$$\begin{aligned} \frac{dr}{dt} = -\frac{M\tau}{k_2^2} \Bigg(\left(e^{-\frac{1}{2} k_2 \dot{\varepsilon} (t - t_{ini}^d)} (-\sqrt{\rho_{ini}} k_2 + k_1) - k_1 \right)^2 - \\ - \left(e^{-\frac{1}{2} k_2 \dot{\varepsilon} (t - t_{ini}^r)} (-\sqrt{\rho_{ini}} k_2 + k_1) - k_1 \right)^2 \Bigg) - \frac{2M\gamma}{r} \end{aligned} \tag{4.16}$$

The numerical solution of this equation compared with the case of the constant dislocation density jump is shown in Fig. 4.2.

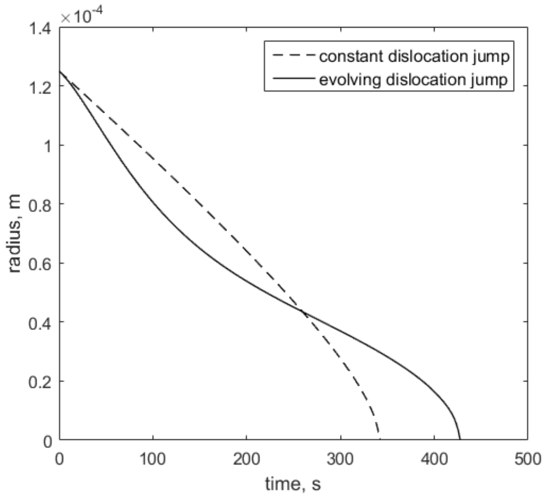


Figure 4.2: The change of the radius of a single grain during shrinkage driven by two different forces acting simultaneously, curvature and dislocation density difference, constant and evolving with time. The corresponding times of the full shrinkage are ~ 342 s and ~ 427.5 respectively.

4.2 Algorithm testing for simulation of a single grain shrinkage driven by different driving forces

The combination of the two driving forces, curvature and constant dislocation density difference jump, acting simultaneously will lead to faster shrinkage than at forces acting separately as shown in Fig. 4.1. A series of simulations has been implemented, for the different cell sizes and values of $Kink$ for a fixed and an updated time increment. The parameter A was taken from the results in the previous chapter for the corresponding set of parameters: cell size, $Kink$ and type of time increment.

4.2.1 Simulation of a single grain shrinkage simultaneously driven by the constant dislocation density jump and curvature using a developed algorithm

The full time of a grain shrinkage for the analytical solution Eq. (4.15) is $t_{an} \approx 342$ s (see Fig. 4.1). The difference between the simulated time and analytical solution with the change of the cell sizes is shown in Fig. 4.3. The deviation of the simulated results from the analytical solution have been estimated as a relation t_{sim}/t_{an} and plotted against cell sizes. The ideal relation is 1 and marked by a black dashed line. As can be seen from the graph, the closest to the analytical solution simulation has been performed by CA with the updated t_{inc} at $Kink = 12$. However, the general tendency is the decrease of the characteristic t_{sim}/t_{an} with the decrease of the cell size. The updated time increment in comparison with the fixed give slightly better performance.

If the relation t_{sim}/t_{an} would be equal 1, then it could also cause significant deviations from the analytical solution, because of the wrong shape. Fig. 4.4 shows how the shape of the scaled solutions differs from the analytical curve. The simulations were performed at the cell size $d_{cell} = 1.25 \times 10^{-6}$ m or 100 cells.

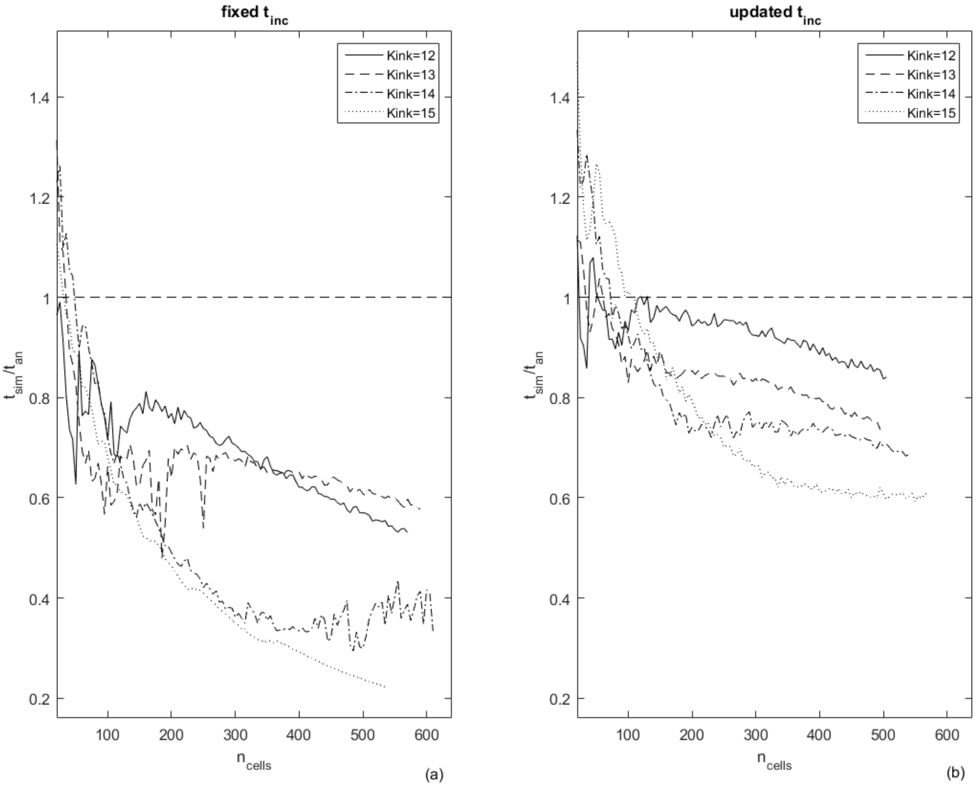


Figure 4.3: The change of the deviations of the simulations of a single grain during shrinkage, driven by curvature combined with a constant dislocation density difference, from the theory calculated for different values of $Kink$ and types of CAS: fixed (a) and updated t_{inc} (b) with the decrease of the cell size. The deviation has been estimated as a relation of the simulated time of shrinkage to the analytical t_{sim}/t_{an}.

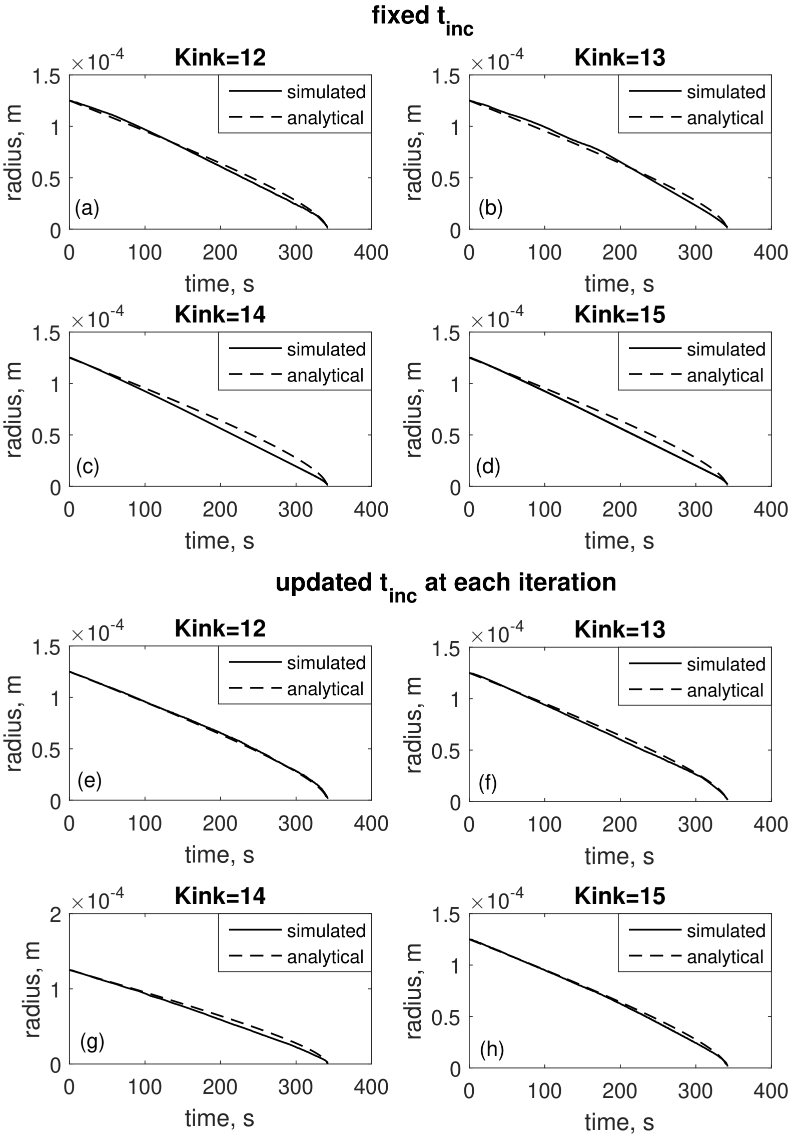


Figure 4.4: The change of the radius of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump simulated at the different values of $Kink$ and scaled to the analytical solution, for the fixed (a)-(d) and updated (e)-(h) t_{inc}. The cell size is $d_{cell} = 1.25 \cdot 10^{-6}$ m or 100 cells.

As can be seen from this figure $Kink = 14$ and $Kink = 15$ at the fixed t_{inc} lead to a significant deviation of the shape from the analytical curve. Whereas $Kink = 12$ at the updated t_{inc} shows almost ideal prediction of shape. The corresponding errors of the scaled simulations at the different cell sizes are shown in Fig. 4.5.

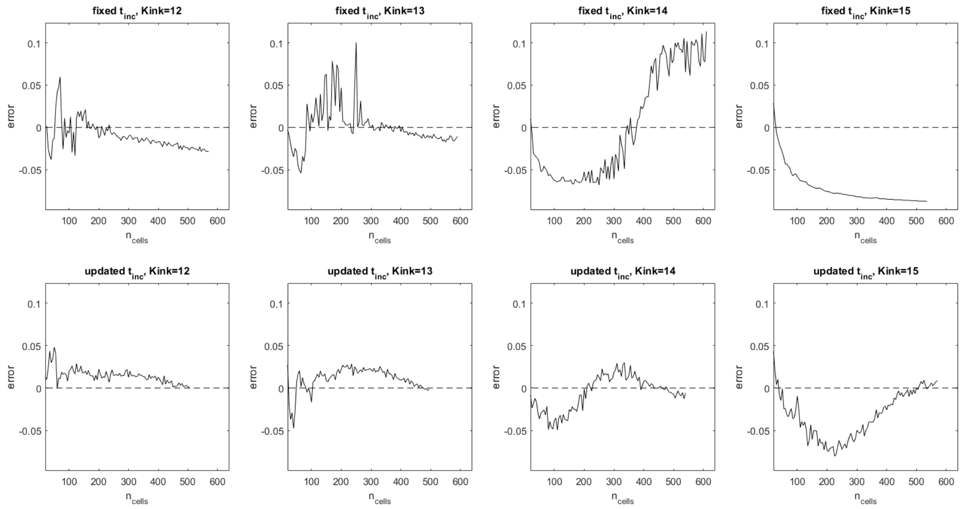


Figure 4.5: The errors of the scaled simulations of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated (bottom) t_{inc} with the decrease of the cell sizes.

The simulations performed by CA with the updated t_{inc} at $Kink = 12$ have minimal levels of error which decrease with the number of cell per initial radius. These two contributions, minimal deviation of shape and time, give a better total prediction, which can be seen in Fig. 4.6, where the minimal level of error has been observed also for the case of the updated t_{inc} at $Kink = 12$. The degradation of the performance characteristic t_{sim}/t_{an} (see Fig. 4.3) has a significant contribution to the increase of the error.

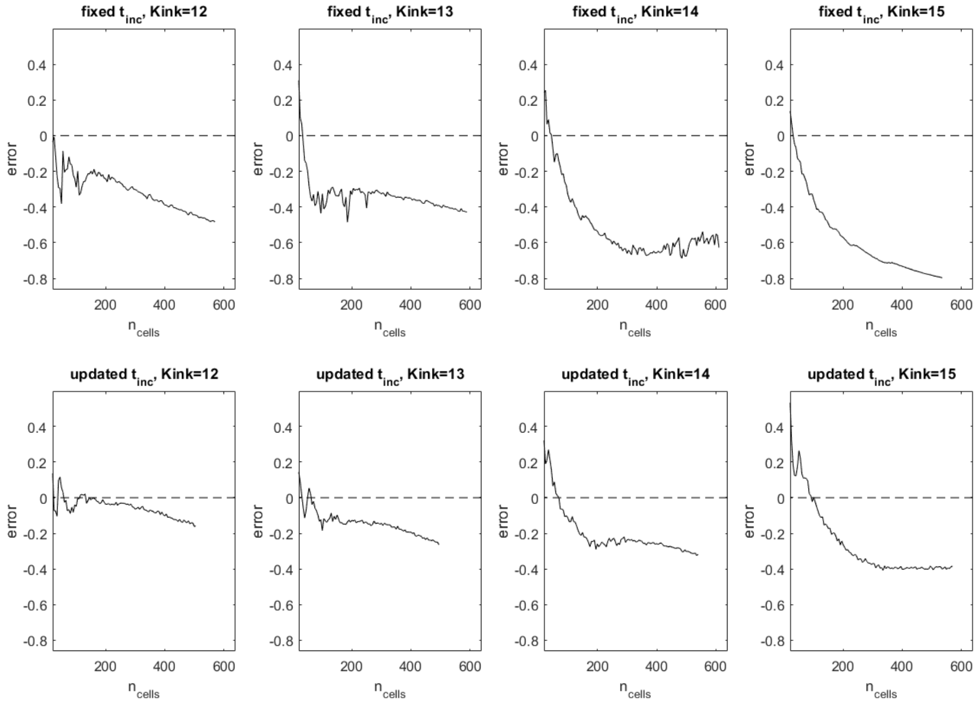


Figure 4.6: The errors of the non-scaled simulations of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated t_{inc} (bottom), with the decrease of the cell size.

The examples of simulations performed at the cell size $d_{cell} = 4.3860 \times 10^{-7}$ m (or 285 cells) compared with the analytical solutions are shown in Fig. 4.7. Although the CA simulation with the updated t_{inc} and $Kink = 12$ is not ideally close to the analytical solution, its advantage as compared with others values of $Kink$ is apparent.

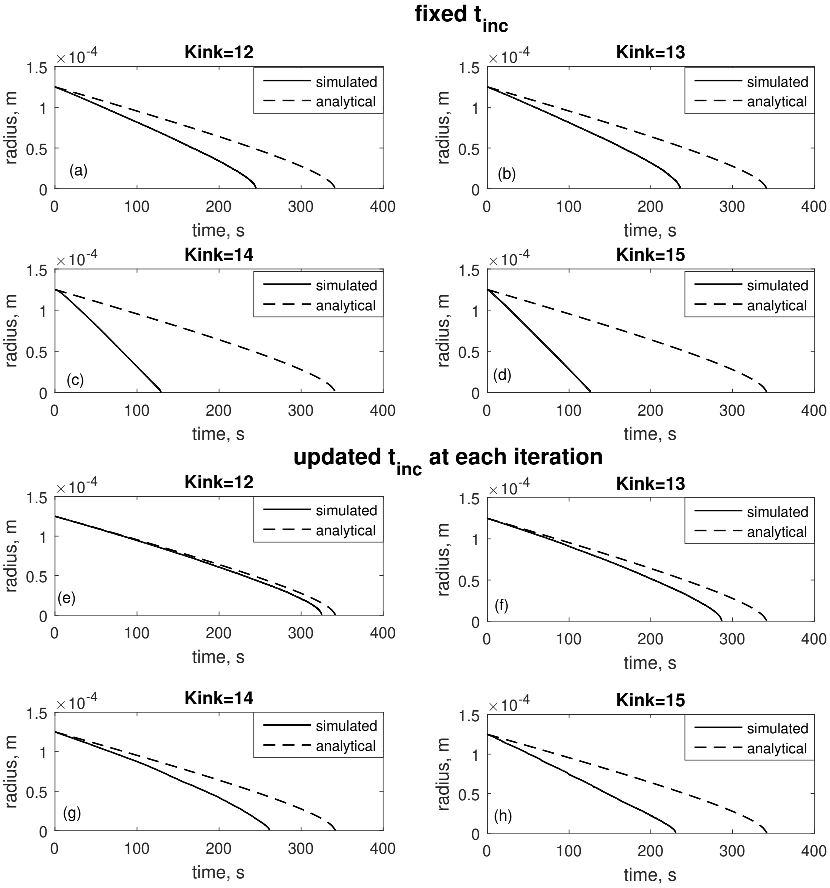


Figure 4.7: The change of the radius of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump simulated at the different values of $Kink$ and types of CAS: fixed (a)-(d) updated (e)-(h) t_{inc}, compared with the analytical solution. The cell size is $d_{cell} = 4.3860 \times 10^{-7}$ m or 285 cells.

4.2.2 Simulation of a single grain shrinkage simultaneously driven by the evolving dislocation density jump and curvature using a developed algorithm

Another case of the coupled driving forces studied in the present research is the combination of the evolving dislocation density and curvature driven forces acting simultaneously. In this case the shape of curve is more complex (see Fig. 4.2). By the analogy with the previous subsection, a series of CA simulations has been carried out and the values t_{sim}/t_{an}, errors of the scaled and non scaled simulation curves have been found at the different cell sizes, time increment t_{inc} and values of $Kink$.

The relation t_{sim}/t_{an} is shown in Fig. 4.8. As can be seen from this figure, the CA with the updated t_{inc} and $Kink = 12$ leads to the more accurate prediction of time of the grain shrinkage. This result agrees with the result for the constant dislocation jump in the previous subsection.

The difference in shape of the simulated curves compared with the analytical solution at the different values of $Kink$ and t_{inc} is shown in Fig.4.9. The cell size used in the simulations is $d_{cell} = 4.717 \times 10^{-7}$ m or 265 cells. The values $Kink = [14, 15]$ at the fixed t_{inc} lead to almost straight lines, whereas $Kink = 12$ at the both, fixed and updated, t_{inc} predict the shape. To investigate how prediction of the shape improves with the decrease of the cell size, the errors of the scaled simulations were plotted against the number of cells per initial radius n_{cells} (see Fig.4.10). As can be seen from this graph, the mean error for $Kink = 12$ is less than for others. However, the error increases with the decrease of cell size. The errors of the non-scaled simulations are shown in Fig.4.11. The errors increase with the decrease of the cell size for all cases, and lie in the range greater than for the constant dislocation jump. An

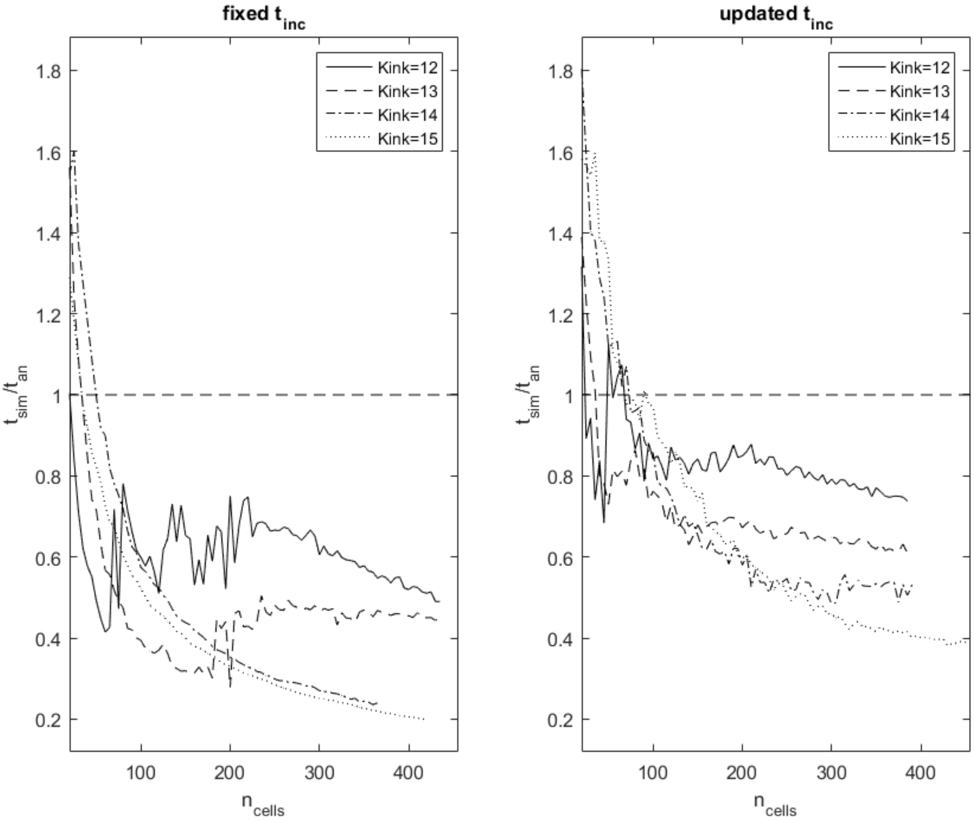


Figure 4.8: The change of the deviations of the simulations of a single grain during shrinkage driven by curvature combined with a evolving dislocation density difference, from the theory calculated for different values of $Kink$ and types of CAS: fixed (a) and updated t_{inc} (b) with the decrease of the cell size. The deviation has been estimated as a relation of simulated time of shrinkage to the analytical t_{sim}/t_{an}

example of simulations is shown in Fig.4.12, where the significant differences between the simulations and the analytical solutions can be seen.

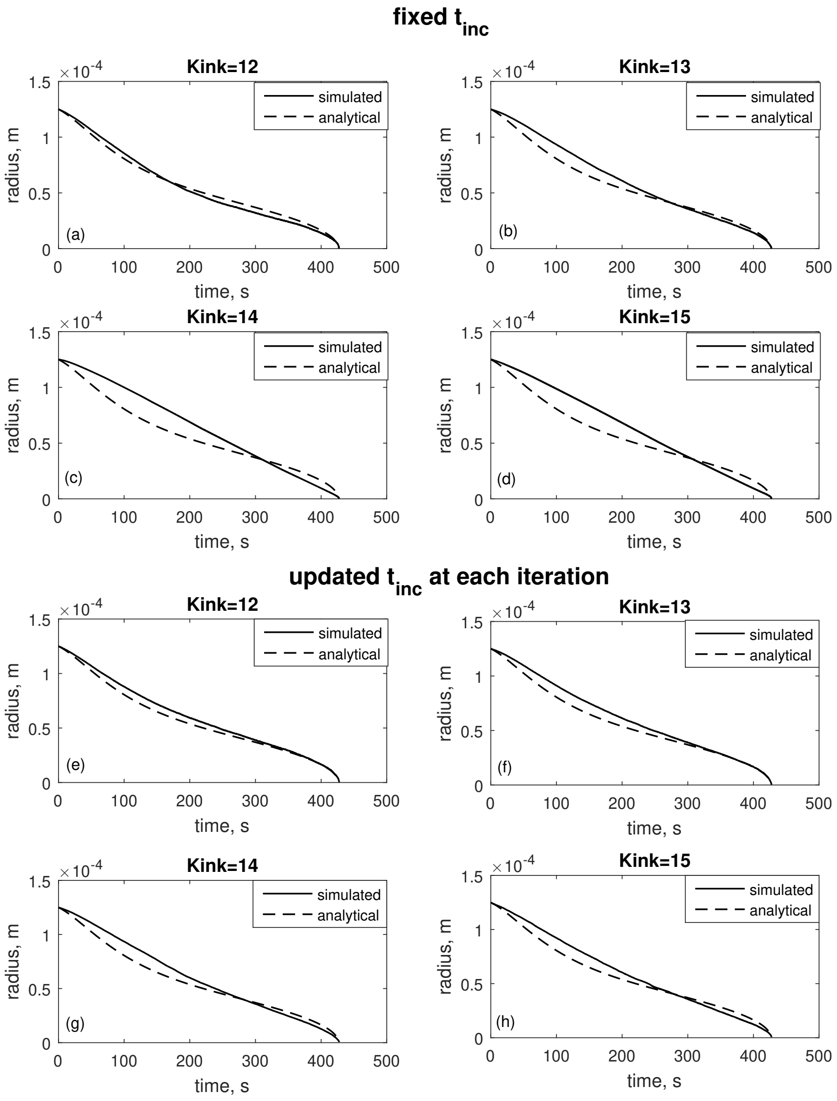


Figure 4.9: The change of the radius of a single grain during shrinkage, driven by curvature combined with an evolving dislocation density difference, simulated at the different values of $Kink$ types of CAS: fixed (a)-(d) and updated (e)-(h) t_{inc}, and scaled to the analytical solution. The cell size is $d_{cell} = 4.717 \times 10^{-7}$ m or 265 cells.

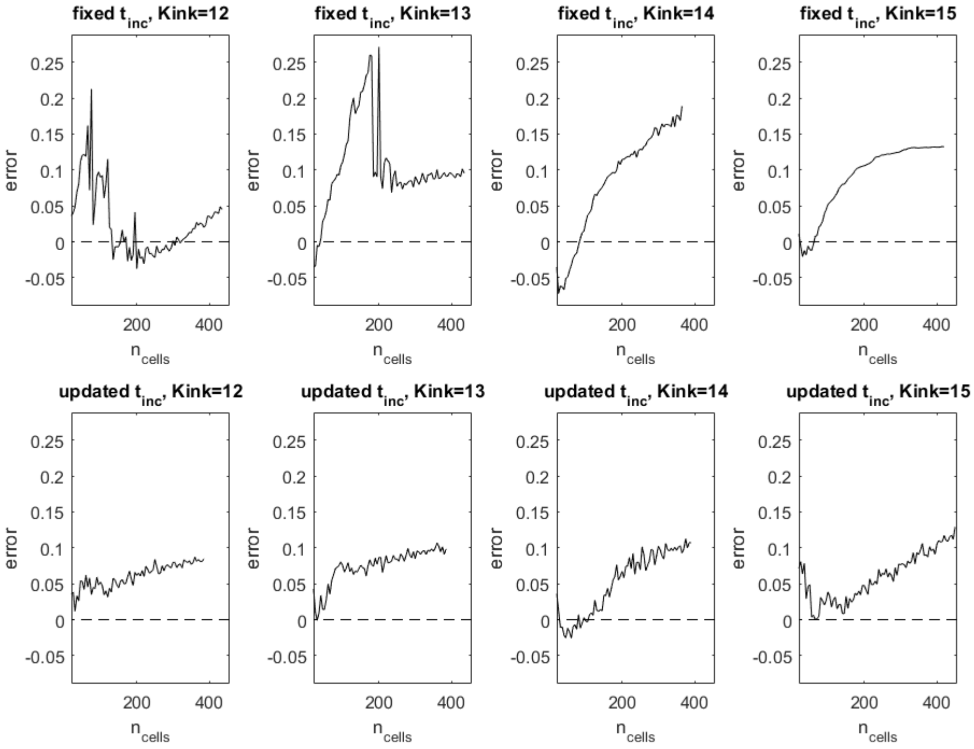


Figure 4.10: The errors of the scaled simulations of a single grain during shrinkage, driven by curvature combined with an evolving dislocation density difference, performed with the different values of $Kink$ and types of CAS: fixed (top) and updated (bottom) t_{inc} with the decrease of the cell sizes.

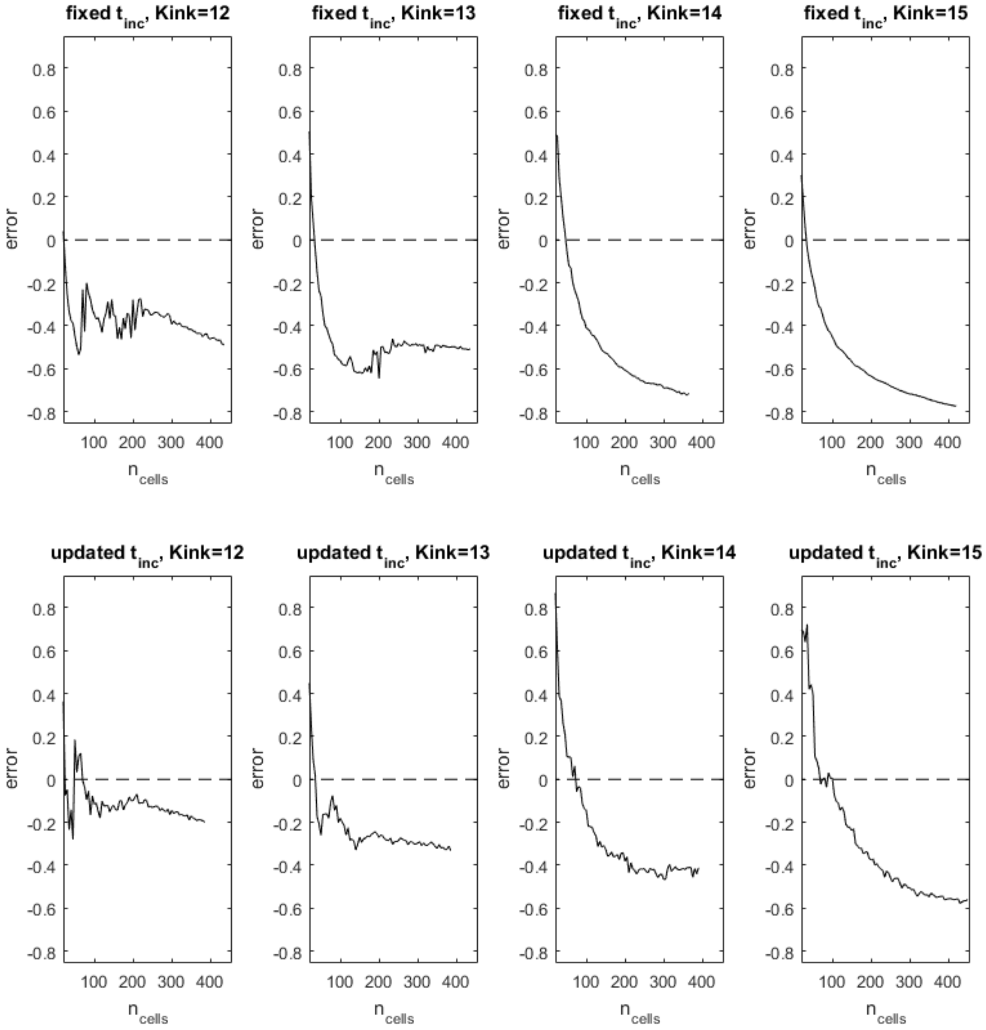


Figure 4.11: The errors of the non-scaled simulations of a single grain during shrinkage driven by curvature combined with an evolving dislocation density difference, performed with the different values of $Kink$ and types of CAS: fixed (top) and updated t_{inc} (bottom), with the decrease of the cell size.

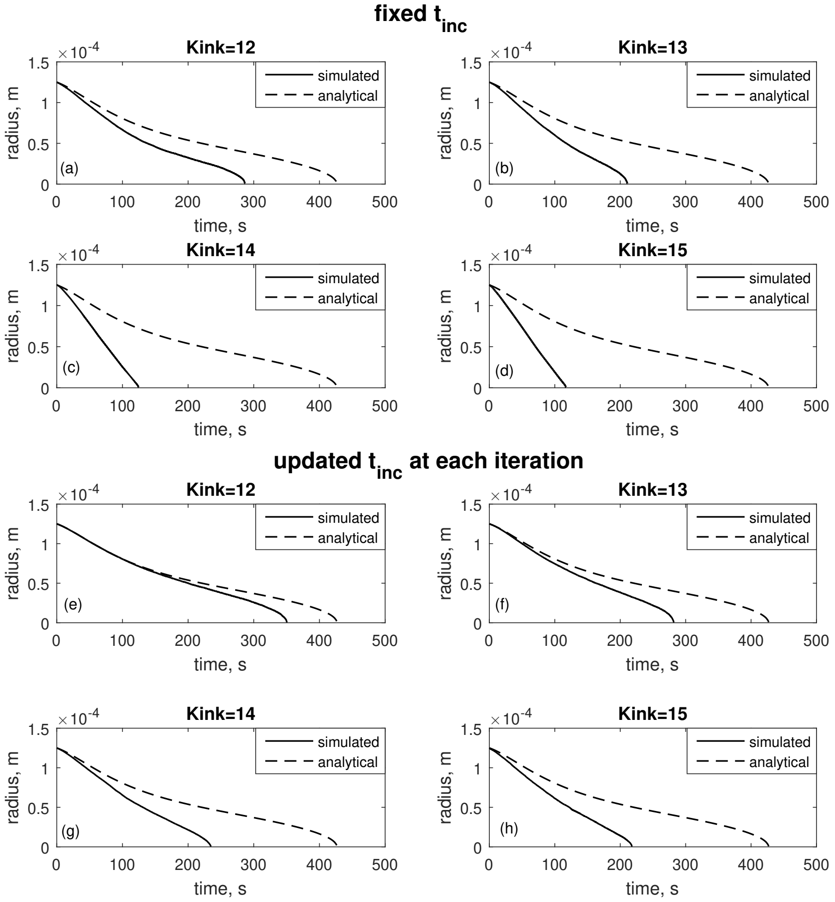


Figure 4.12: The change of the radius of a single grain during shrinkage driven by curvature combined with an evolving dislocation density jump simulated at the different values of $Kink$ and types of CAS: fixed (a)-(d) updated (e)-(h) t_{inc}, compared with the analytical solution. The cell size is $d_{cell} = 4.3860 \times 10^{-7}$ m or 285 cells.

4.3 Probabilistic CA technique testing for simulation of a single grain shrinkage driven by the coupled driving forces

The probabilistic CA technique described in Chapter 3 has been applied for solving coupled differential equation and analysed in the same manner as the deterministic CA method. The parameter A has been obtained during simulation on curvature (see Chapter 3), and the average value has been used for the corresponding cell size in the simulations for the coupled driving forces.

4.3.1 Simulation of a single grain shrinkage simultaneously driven by the constant dislocation density jump and curvature driven forces using developed PCA algorithm

The simulated time of the grain shrinkage related to the analytical solution with the change of the cell sizes is shown in Fig. 4.13. In comparison with the deterministic CA (see Fig. 4.3) one can say that the probabilistic CA allows to obtain the simulation results closer to the analytical than with the deterministic CA, especially at the small cell sizes.

The deviation of the simulated results from the analytical solution has been estimated as a relation t_{sim}/t_{an} and plotted against cell sizes. The bars show standard deviation of the value t/t_{sim}. As can be seen from the graph, at the large cell sizes the accuracy of the prediction estimated via t_{sim}/t_{an} is within 20%. The parameter $Kink = 15$ gives an improvement of the characteristic t_{sim}/t_{an} with the decrease of the cell size and approaches 1, whereas other parameters of $Kink$ fluctuate around the same value for the updated and slowly grow for the fixed time increment. The bars on the graph show

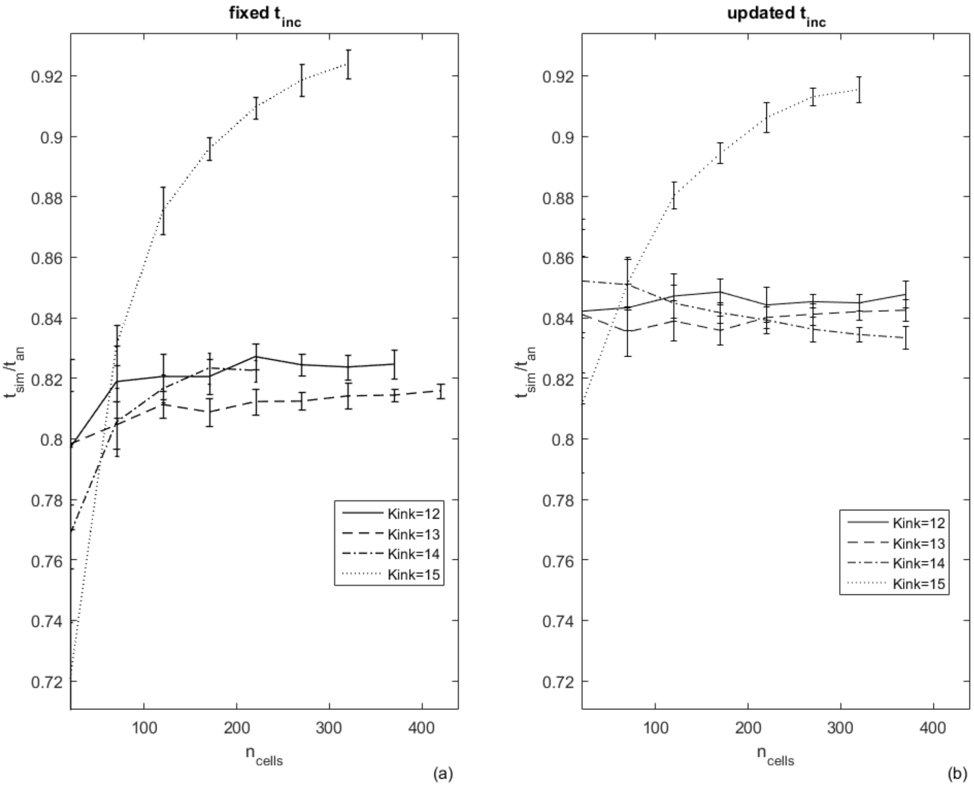


Figure 4.13: The change of the deviations of the PCA simulations of a single grain during shrinkage, driven by curvature combined with a constant dislocation density difference, from the theory calculated for different values of $Kink$ and types of CAS: fixed (a) and updated t_{inc} (b) with the decrease of the cell size. The deviation has been estimated as a relation of simulated time of shrinkage to the analytical t_{sim}/t_{an}.

that the standard deviation does not radically change with the decrease of the cell size.

The errors of the scaled simulations are shown in Fig. 4.14. As can be seen from this graph, the range of errors of the scaled simulations, which reflects the similarity of the shapes of the curves, for PCA is also smaller than for DCA (see Fig. 4.5).

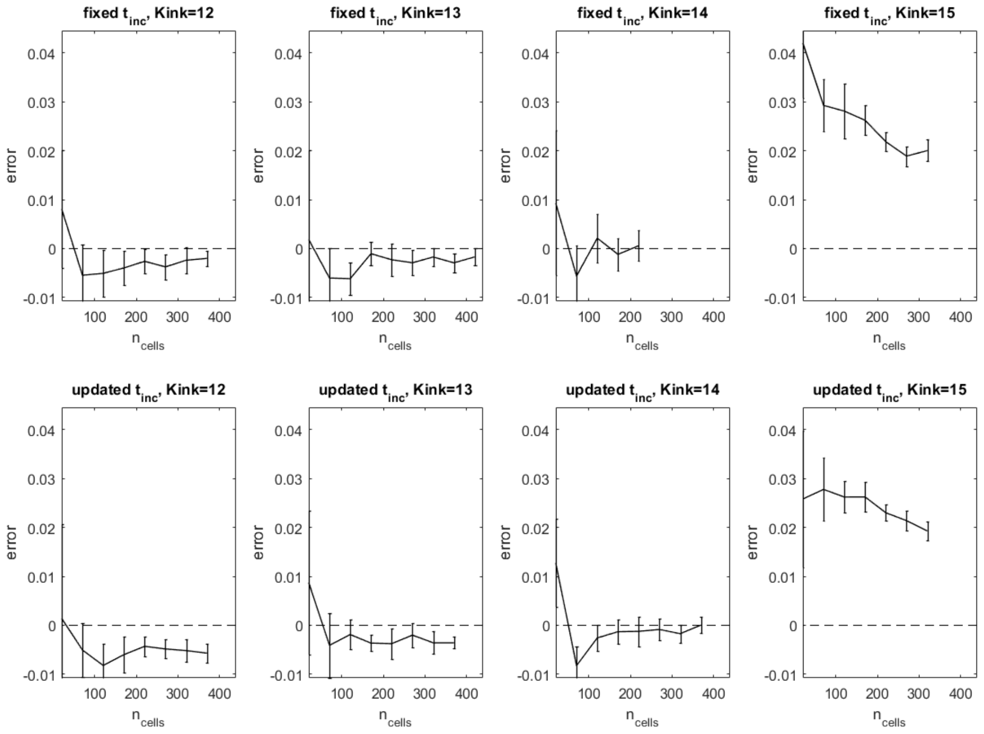


Figure 4.14: The errors of the scaled PCA simulations of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated (bottom) t_{inc} with the decrease of the cell sizes.

The errors for the real simulations (see Fig. 4.15) are all negative and $Kink = 15$ gives errors approaching zero with the decrease of the cell size. This behaviour is related to the improvement of t_{sim}/t_{an}.

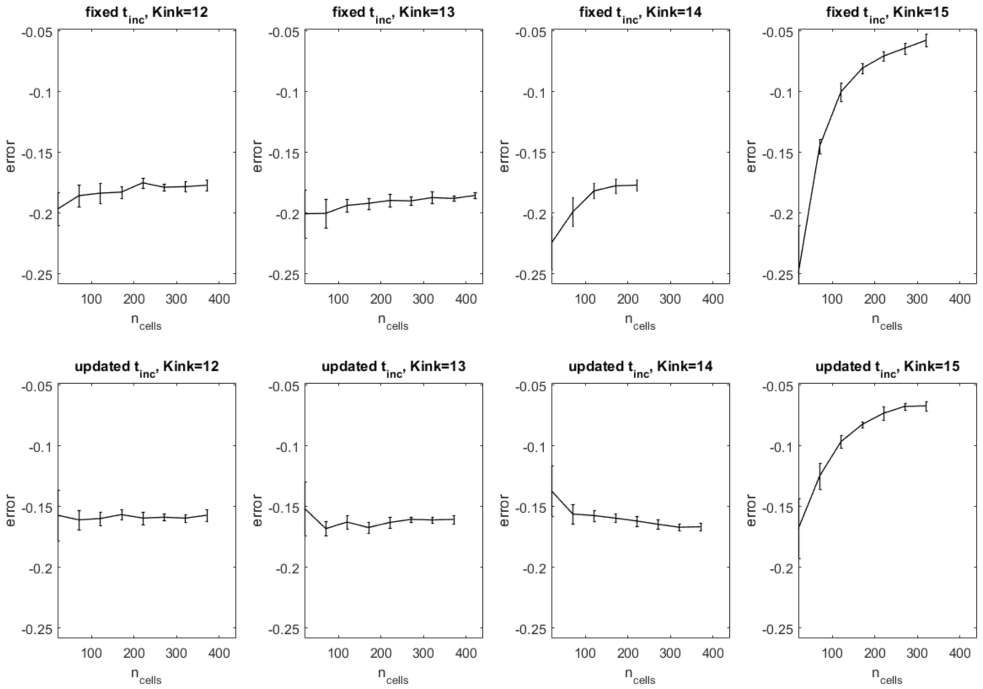


Figure 4.15: The errors of the non-scaled PCA simulations of a single grain during shrinkage driven by curvature combined with a constant dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated t_{inc} (bottom), with the decrease of the cell size.

4.3.2 Simulation of a single grain shrinkage simultaneously driven by the evolving dislocation density jump and curvature driven forces using developed PCA algorithm

By the analogy with the previous section the PCA version of a developed technique has been applied for the 2D simulation of the single grain shrinkage driven by evolving dislocation density jump and curvature simultaneously. Fig. 4.16 shows the characteristic t_{sim}/t_{an} and its behaviour with the cell size is very similar to the previous case. However, the average value of t_{sim}/t_{an} is smaller, which means low accuracy of simulations.

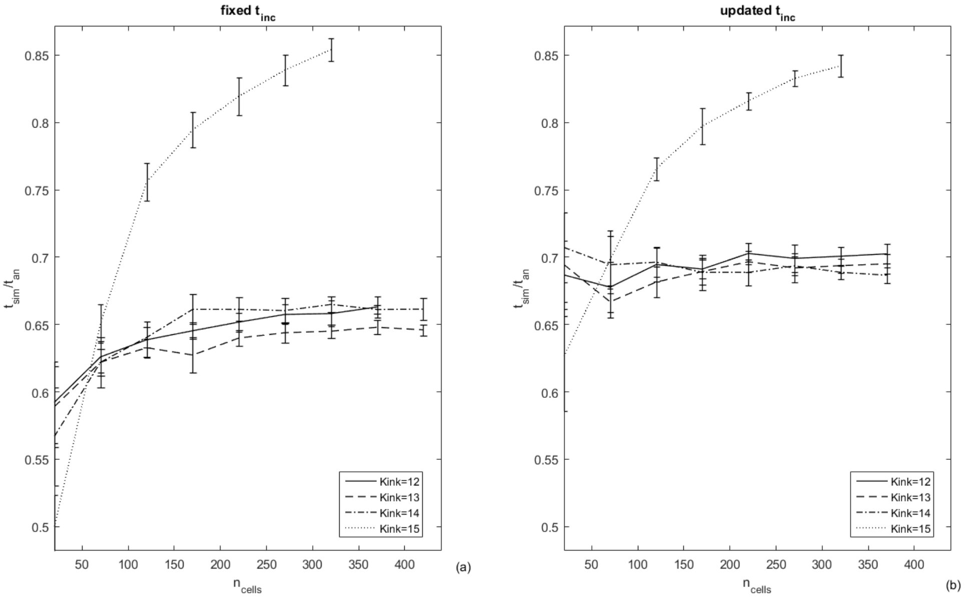


Figure 4.16: The change of the deviations of the PCA simulations of a single grain during shrinkage, driven by curvature combined with an evolving dislocation density difference, from the theory calculated for different values of $Kink$ and types of CAS: fixed (a) and updated t_{inc} (b) with the decrease of the cell size. The deviation has been estimated as a relation of simulated time of shrinkage to the analytical t_{sim}/t_{an}.

The shape of the scaled curves (see Fig.4.17) in comparison with the DCA is better, $Kink = 15$ starts from the bigger errors but then with the decrease of the cell size, errors approach zero, as well as other parameters.

The real errors (see Fig. 4.18) also approach zero with the decrease of the cell size in contrast to the deterministic algorithm.

4.4 Summary

A developed CA has been tested in the case of the coupled driving forces acting simultaneously, where the dislocation density jump first was chosen as

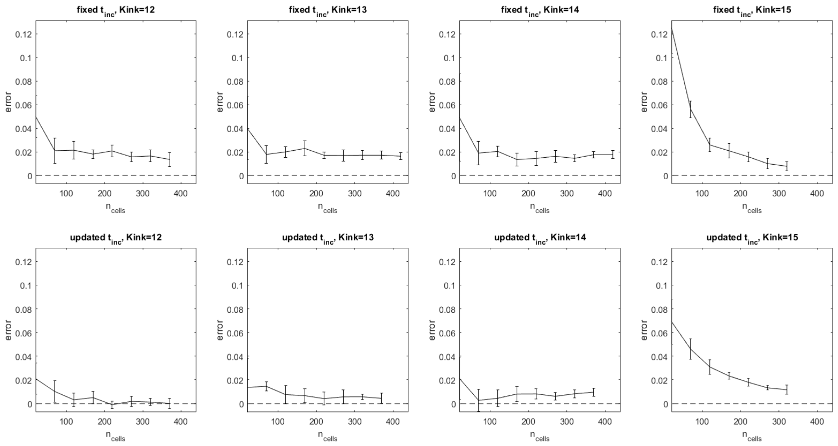


Figure 4.17: The errors of the scaled PCA simulations of single grain during shrinkage driven by curvature combined with an evolving dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated (bottom) t_{inc} with the decrease of the cell sizes.

a constant, and in the next series of experiments as evolving due to hardening and recovery. It has been shown that the values of $Kink$ and the type of time increment influence the shape of curve of the solution. The best variables are $Kink = 12$ at the updated t_{inc}, whereas $Kink = 14, 15$ showed the biggest discrepancy in the shape. The error approaches zero or fluctuates near a constant value in the case of the constant dislocation density jump, whereas for evolving dislocation density jump it grows with the decrease of cell size.

A developed CA solves the system of equations for the evolving dislocation density with the bigger errors. One of the reasons of that is non-symmetrical grid, which caused the high errors in the simulations of evolving dislocation density jump in the previous chapter. However, it showed the ability to solve equation for the constant dislocation density jump with the appropriate choice of the parameters $Kink$ and type of time increment.

Although the advantage of using PCA was not very obvious in Chapter

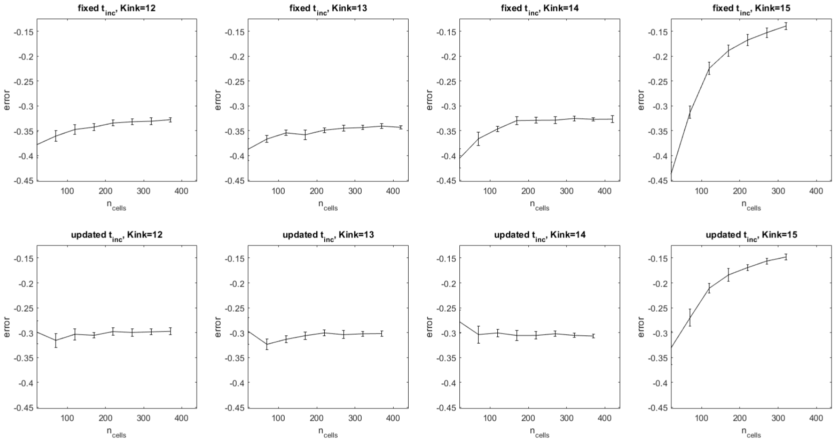


Figure 4.18: The errors of the non-scaled PCA simulations of a single grain during shrinkage driven by curvature combined with an evolving dislocation density jump performed with the different values of $Kink$ and types of CAS: fixed (top) and updated t_{inc} (bottom), with the decrease of the cell size.

3, it showed better results for the simulations of the combined differential equations, especially when the updated CAS and $Kink = 15$ are used. In contrast to the deterministic model, PCA has ability to minimize error with the decrease of the cell size which means that the solver converges. Also, PCA has a wider range of the possible values of $Kink$ than the deterministic CA.

Chapter 5

On Time Symmetry of CA Based Techniques

The Newtonian mechanics (classical mechanics) which describes the motion of the macroscopic objects is deterministic, in other words it is time-symmetric or reversible. The differential equations governing such physical systems can describe the evolution of the system in the future, but also knowing the present state it is possible to say how the system has evolved in the past. Therefore, the shrinkage of the grain is described by the same laws as growth, but with the opposite sign in the driving forces.

Let us consider an example of the dislocation density jump driving force. If there is a deformed grain d with initial radius r_0^d surrounded by recrystallized matrix r, then these grains have dislocation density difference $[\rho] = \rho_r - \rho_d < 0$ which gives the driving force for the grain shrinkage $p_D = \tau[\rho] < 0$ to move the grain boundary with velocity $v_d = Mp_D < 0$ towards the centre. Now, let us consider another situation, where the recrystallized nucleus with a very small radius $r_0^r \ll r_0^d$ is surrounded by deformed matrix d, and the

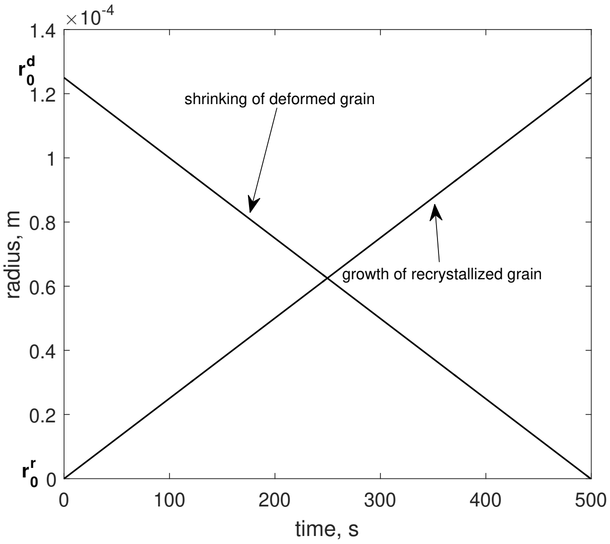


Figure 5.1: The growth and shrinkage of a single grain driven by equal in absolute value and opposite in sign dislocation density difference driving force.

dislocation density is now positive $[\rho] = \rho_d - \rho_r > 0$, so the driving force $p_D = \tau[\rho] > 0$ moves grain boundary away from centre and the velocity is positive $v_r = Mp_D > 0$.

The velocities of the deformed and recrystallized grain are opposite in sign and equal in the absolute values: $|v_r| = |v_d|$, $v_r = -v_d$. The evolution of the radius with time for this two examples is schematically shown in Fig. 5.1. In fact, the considered examples reflects the time-symmetry: the negative sign of $[\rho]$ can be taken to the left side as follows:

$$\frac{dr}{d(-t)} = M\tau[\rho]. \tag{5.1}$$

In the case of the curvature driven force, the equation of the grain shrinkage is as follows:

$$\frac{dr}{dt} = -M\frac{2\gamma}{r}, \tag{5.2}$$

where the driving force is $p_C = -\frac{2\gamma}{r}$. The hypothetical driving force with opposite sign and equal in absolute value $p_C = \frac{2\gamma}{r}$, will make the grain grow instead of shrinkage (see Fig. 5.2), which can also be interpreted as time reversibility, as follows:

$$\frac{dr}{d(-t)} = -M\frac{2\gamma}{r}, \tag{5.3}$$

Therefore, the concept of time symmetry for shrinkage of a single grain, in the sense of the deterministic behaviour, can be expressed via the following expressions:

$$\frac{dr}{dt} = Mp_{shrinkage}, \tag{5.4}$$

$$\frac{dr}{d(-t)} = Mp_{growth}, \tag{5.5}$$

$$|p_{shrinkage}| = -|p_{growth}| \tag{5.6}$$

The CA techniques which are used to solve the time-symmetric differential equations are required to be able to simulate both, the shrinkage and the growth of the grain, for the case of the deterministic CA, so the CA needs to be reversible, which means that each configuration has an unique predecessor. Unfortunately, the discrete methods for simulations rarely show the deterministic properties [121]. The problems and the potential solutions for the suggested CA techniques are discussed in this chapter.

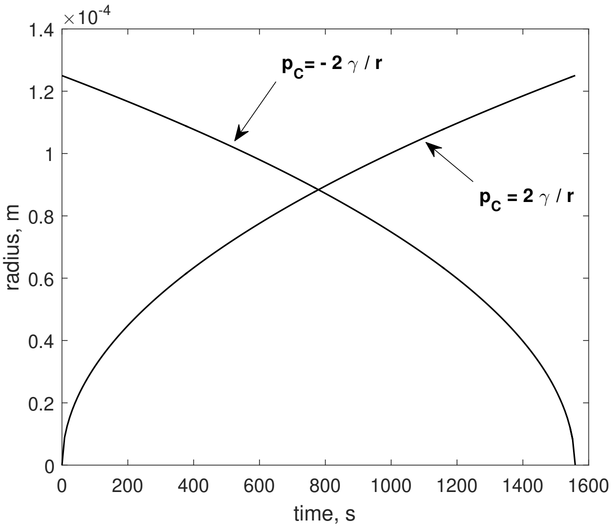


Figure 5.2: The growth and shrinkage of a single grain driven by equal in absolute value and opposite in sign dislocation density difference driven force.

5.1 Time symmetry of the grain growth-shrinkage driven by curvature

The grain boundary velocity of the curvature driven grain growth is described as follows: for the grain growth driven by curvature it can be described as follows:

$$v = Mp = mk\gamma, \tag{5.7}$$

where grain boundary curvature k is theoretically defined as $k = 2/R$ [4] [4] The commonly used rules for calculating curvature are based on the disk-template approach (see Fig. 3.22). This CA technique is well described by

[4] In some articles (see Lan et al. [34], Han et al. [37]) it has been stated that curvature is defined as $k = -1/R$, whereas in other works (see Hallberg [42], [41], Xiao [119], Kigler [39], [1]) it has been defined as $k = -2/R$. In the present research the second version has been used, because it agrees with Gibbs-Thomson relationship [4].

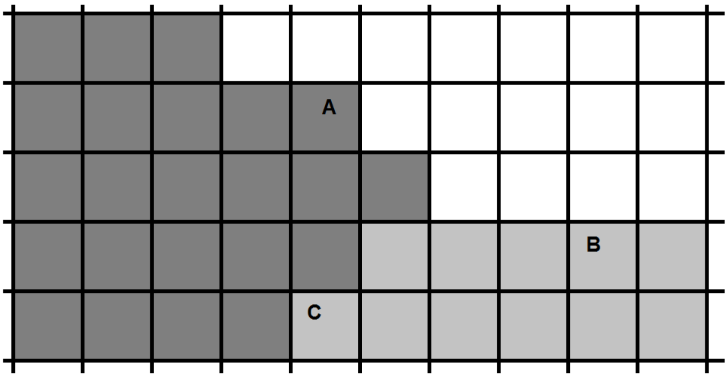


Figure 5.3: Grain boundaries in the triple junction.

Janssens[1]. The curvature is calculated via the following formula:

$$k = 15 - \sum \phi_i \varepsilon_i, \tag{5.8}$$

where i is the location of the cell within neighbourhood, ε_i is the weight of the cell i on the template and is equal to 1 in this case, a phase value $\phi_i = 1$ if the cell i belongs to the same grain as the central cell and $\phi_i = 0$ if the cell i does not belong to the same grain of the central cell, see Fig. 3.22 a. The technique is probabilistic: the cell changes its state with the probability $P = \frac{k}{k_{max}}$. In the case of multigrain structure (see Fig. 5.3) phase values ϕ_i should be extended from the binary 0 and 1 to p-adic or real numbers. More details related to the problem of curvature calculations on the multigrain structure is discussed in Appendix A.

An example of the disk shrinkage with the initial radius of 25 cell simulated by Janssens CA, is shown in Fig. 5.4. As can be seen the choice of neighbourhood does not change the average solution, Moore neighbourhood increases standard deviation.

It is worth noting, that Janssens CA result does not fit analytical solution and requires further time scaling. As can be seen from Fig. 5.4c, the average time of the grain shrinkage is about 250 iterations. The plot of the analytical solution of the equation

$$\frac{dr}{dt} = M\frac{2\gamma}{R}, \tag{5.9}$$

is shown in Fig.5.5, the parameters are as follows: the initial radius was $r_0 = 25$ cells, the mobility $M = 1$ $\frac{m^3}{Ns}$ and grain boundary energy $\gamma = 1$ $\frac{J}{m^2}$. Assuming that the cell size is 1 m, that gives the time of the full shrinkage of the grain equal to $t = 156.25$ s, whereas the average number of iterations required for the grain shrinkage is about 250. Therefore, the iteration in Janssens techniques does not equal to seconds and requires scaling.

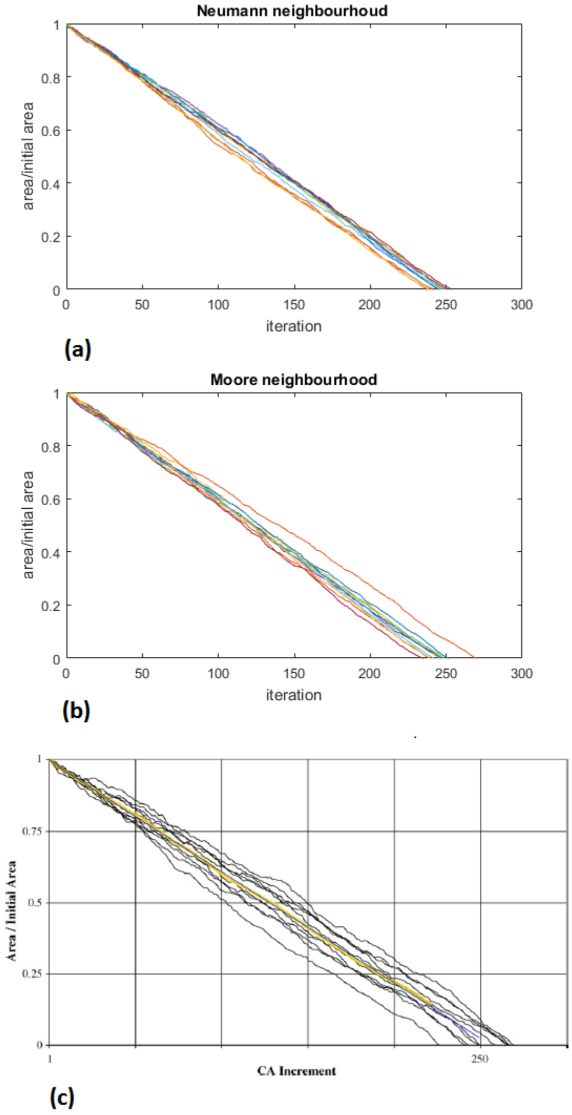


Figure 5.4: 10 different 2D cellular automaton simulations of a single grain with radius of 25 cells shrinkage due to curvature driven force, where the grain boundary was defined using Neumann (a) and Moore (b) neighbourhood, (c) results from original work [1].

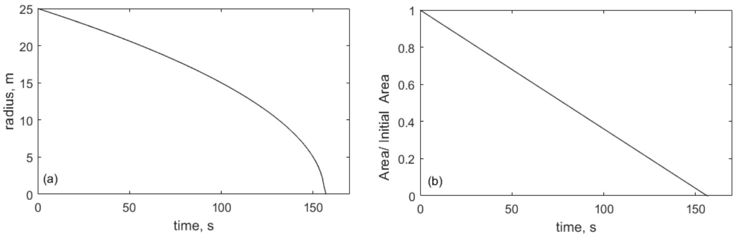


Figure 5.5: The change of the radius and area of a single grain during shrinkage due to curvature. Mobility $M = 1\ \frac{m^3}{Ns}$, grain boundary energy $\gamma = 1\ \frac{J}{m^2}$. The initial radius is $r = 25$ m.

Let us compare Janssen's algorithm with a developed technique in the present work. Fig. 5.6 shows the scaled results for grain shrinkage with two different initial radius $r_0 = 20$ m and $r_0 = 70$, performed by Janssen's algorithm. As can be seen the deviations become smaller with the decrease of the cell size. For comparison, simulations performed by a developed in this work deterministic algorithm are shown in Fig. 5.7 and by the PCA are shown in Fig. 5.8-5.11

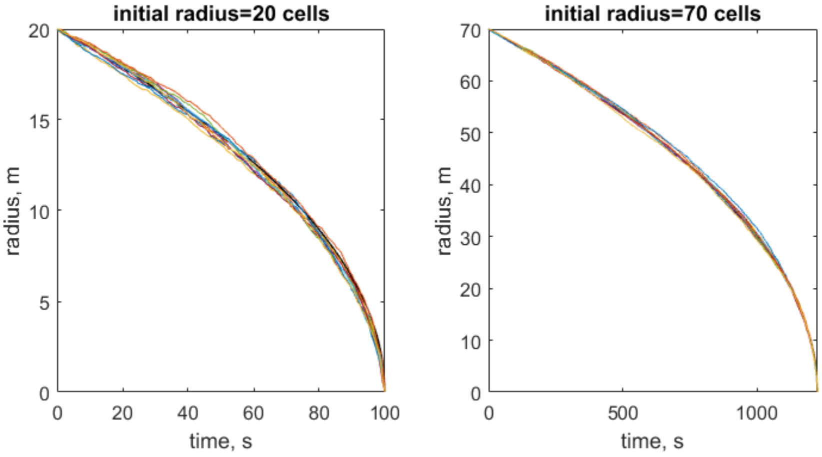


Figure 5.6: 10 2D CA [1] simulations of a single grain shrinkage compared with analytical solution. Initial radii $r_0 = [20, 70]$ cells.

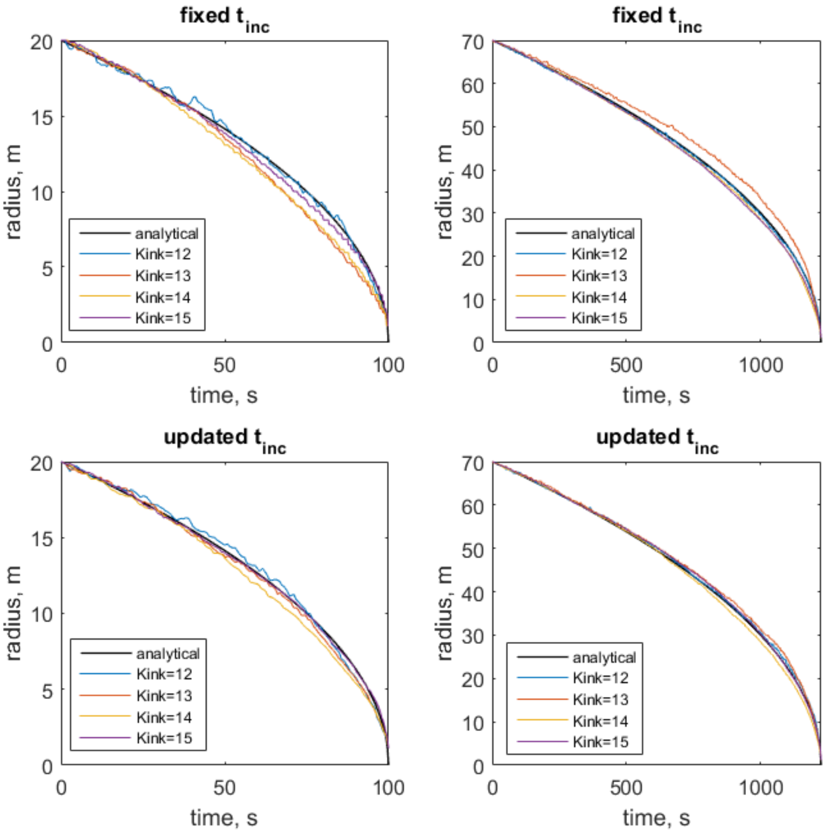


Figure 5.7: 2D deterministic CA simulations of a single grain shrinkage compared with the analytical solution. Initial radii $r_0 = [20, 70]$ cells.

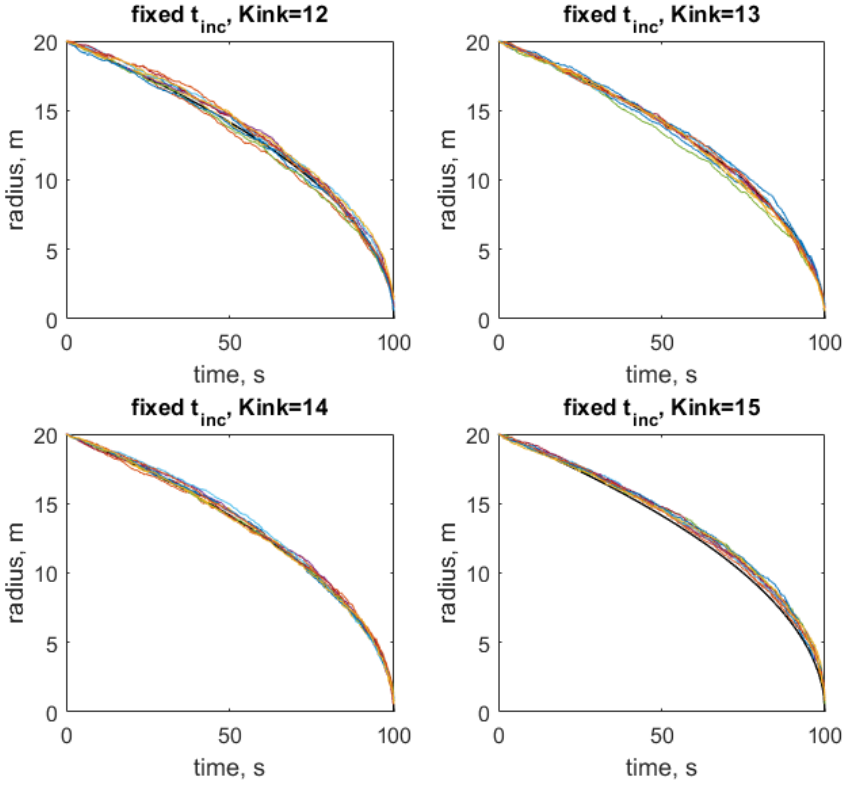


Figure 5.8: 10 2D probabilistic CA simulations with the fixed t_{inc} of a single grain shrinkage compared with the analytical solution. The initial radius $r_0 = 20$ cells.

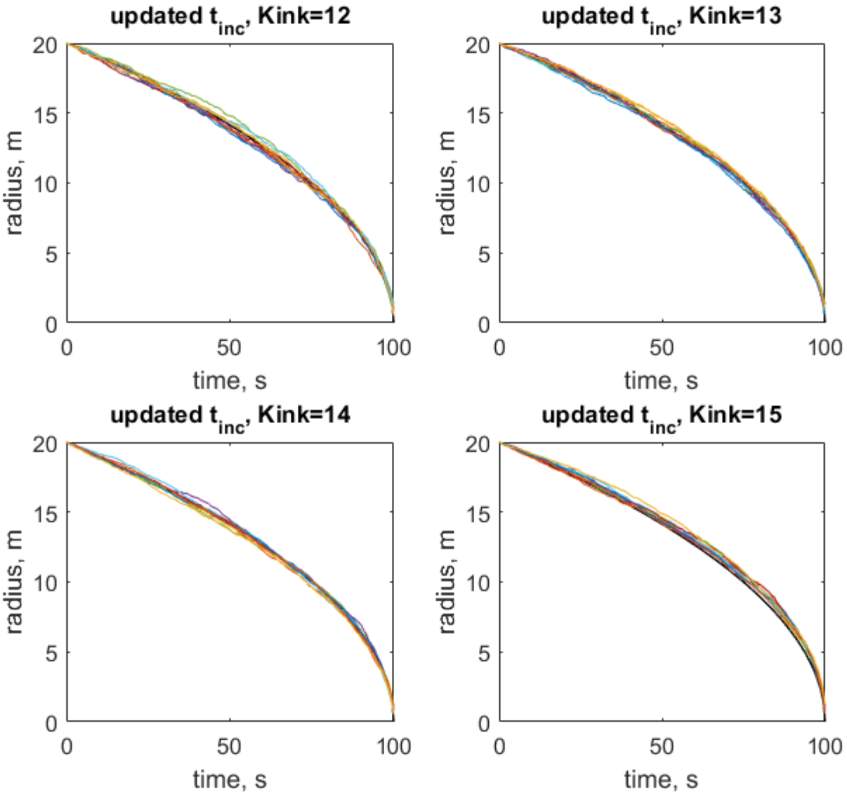


Figure 5.9: 10 2D probabilistic CA simulations with updated t_{inc} of a single grain shrinkage compared with analytical solution. Initial radius $r_0 = 20$ cells.

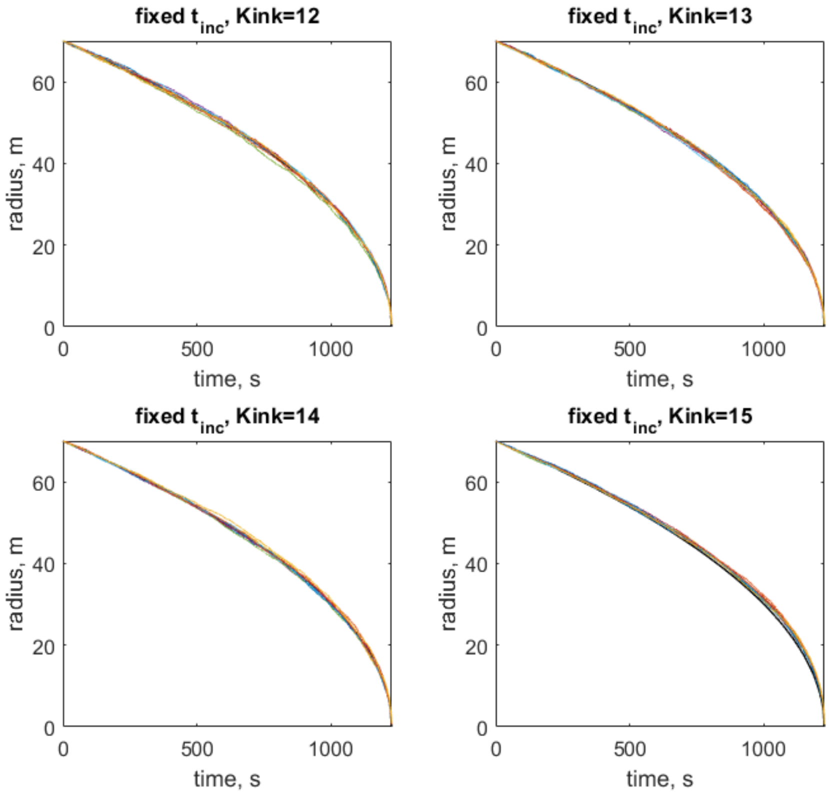


Figure 5.10: 10 2D probabilistic CA simulations with the fixed t_{inc} of a single grain shrinkage compared with the analytical solution. Initial radius $r_0 = 70$ cells.

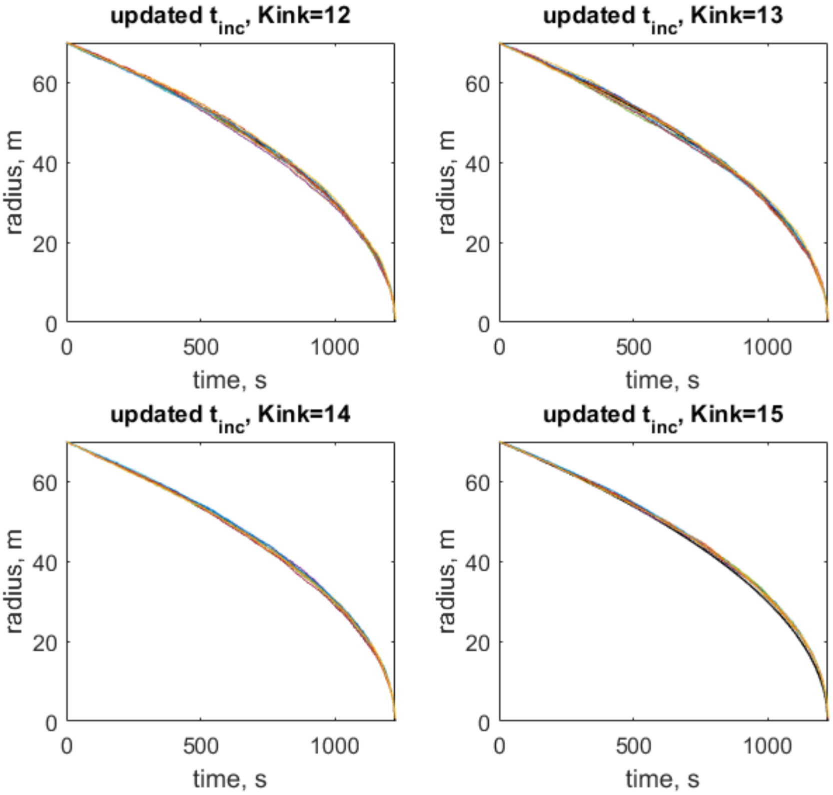


Figure 5.11: 10 2D probabilistic CA simulations with the updated t_{inc} of a single grain shrinkage compared with the analytical solution. Initial radius $r_0 = 70$ cells.

As can be seen from these figures, all methods considered in this work are capable of the simulation the grain shrinkage, with a slightly different but similar accuracy. This approach based on formula Eq. (5.8) for curvature works well for the grain shrinkage, but not for the grain growth. The change of sign in the associated algorithm does not give the desired effect. Fig. 5.12 illustrates the growth and shrinkage of the disk simulated by the same method. As can be seen, the simulations are not reversible.

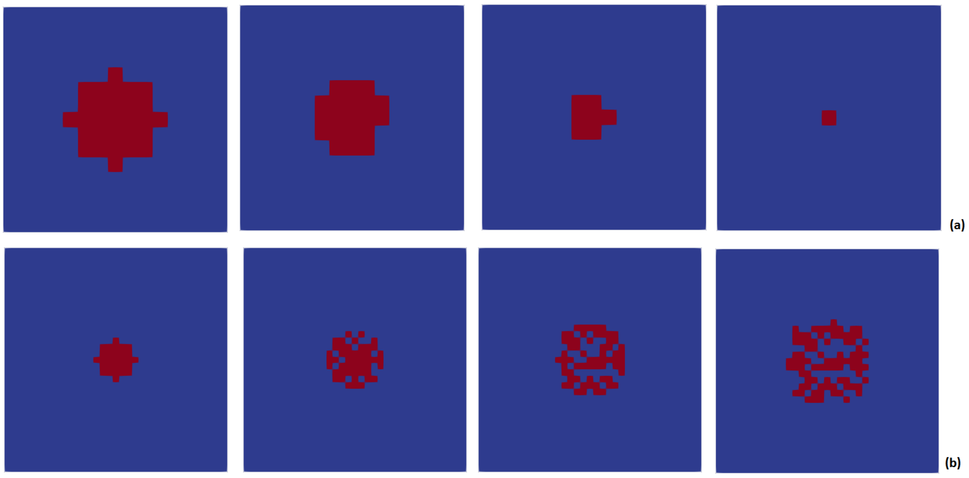


Figure 5.12: a) Grain shrinkage simulated by CA method described in by Janssens[1]; b) Simulation of the grain growth by the same method, where curvature was taken with negative sign.

5.2 Time symmetry of the grain shrinkage-growth driven by dislocation density jump

Fig. 5.13 relates to the simulations by CA in 1D and 2D for the case of linear growth and shrinkage. As can be seen in this figure, 1D is fully reversible, the simulations perfectly predict the analytical solutions, whereas the 2D simulations show the discrepancy from the theory, so in the case of growth simulation errors do not converge. Fig. 5.14 shows a similar comparison for the evolving dislocation density difference. 1D simulations have the same error for growth and for shrinkage, whereas the 2D simulations have a greater error for simulation of growth than for shrinkage.

In the literature, devoted to the study of the grain growth and recrystallization by CA or Monte Carlo methods, it is common to show how the grain shrinks but not how the grain grows, see for examples [1], [98], [101], [34], [27]. Non-reversible curvature calculations and non-symmetry in the calculations of the linear growth-shrinkage (see Fig. 5.13) explain why the study on a single grain shrinkage is usually used instead of the single grain growth.

The 2D CA rules used in the present work are not reversible. Fig. 5.15 illustrates that one configuration can have two different ancestors. Kari (1994) [122], [121] has shown that the reversibility problem is undecidable for the 2D CA with the Neumann neighbourhood. However, in some works, the analytical solution of the grain growth has been compared with the simulations as by Hallberg et al. [42]. They used the hexagonal grid which led to a smaller discrepancy from the theory.

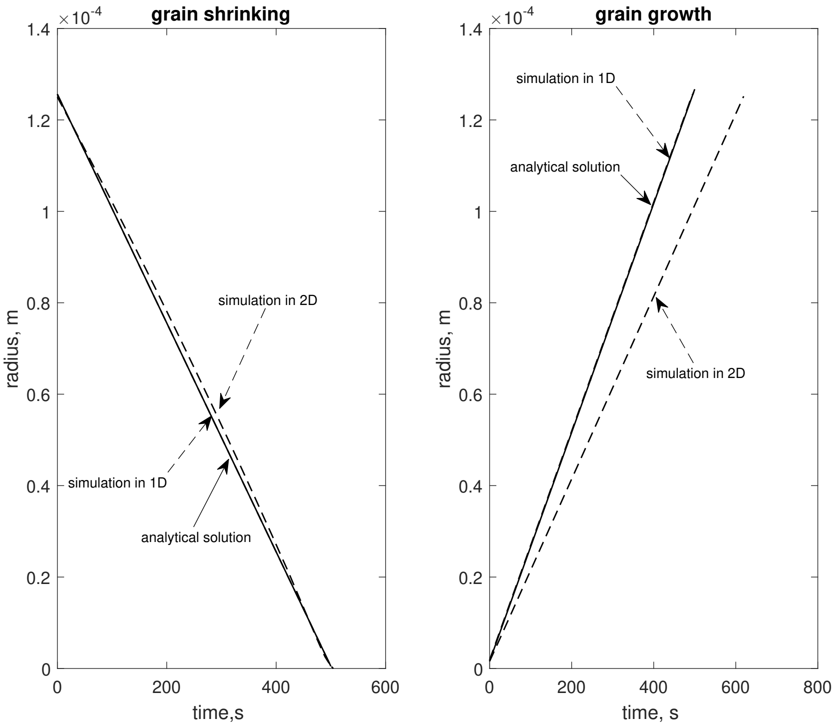


Figure 5.13: The growth and shrinkage of a single grain: analytical solutions compared with simulations (dashed lines) in 1D and 2D. Cell size is $d_{cell} = 1.25 \times 10^{-6}$ m.

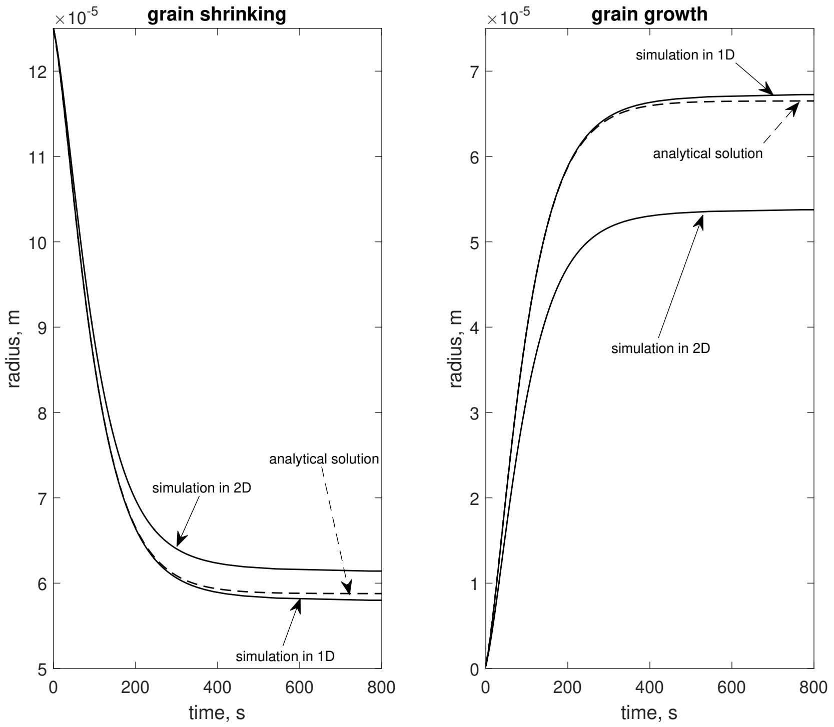


Figure 5.14: The growth and shrinkage of a single grain due to evolving dislocation density difference jump: analytical solutions compared with simulations (dashed lines) in 1D and 2D. Cell size is $d_{cell} = 2.5 \times 10^{-7}$ m.

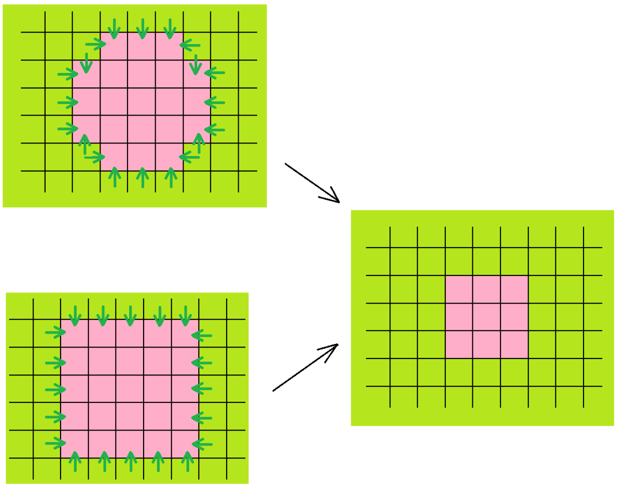


Figure 5.15: For non-reversible CA the same configuration can have two different predecessors.

5.2.1 The nature of errors in 2D simulations of grain growth driven by dislocation density jump

For a deeper understanding of how the error is connected to the grid and neighbourhood, let us look at Fig. 5.16. In the present research, the square grid has been used. The application of CA rules for the grain growth on this non-symmetrical grid will lead to a growth of a square cell in the form of square when the rotated von Neumann and Moore neighbourhoods are used (see Fig. 5.16a). However, the von Neumann neighbourhood, which has been used in the simulation described above, leads to the rhombic grain growth (see Fig. 5.16b).

The error used in the calculations Eq.(5.21) can be rewritten as follows:

$$error = \frac{Area_{rhombus} - Area_{curcle}}{Area_{circe}}, \tag{5.10}$$

where A_{an} is the area of the ideal round grain, and A_{sim} grain in the form of

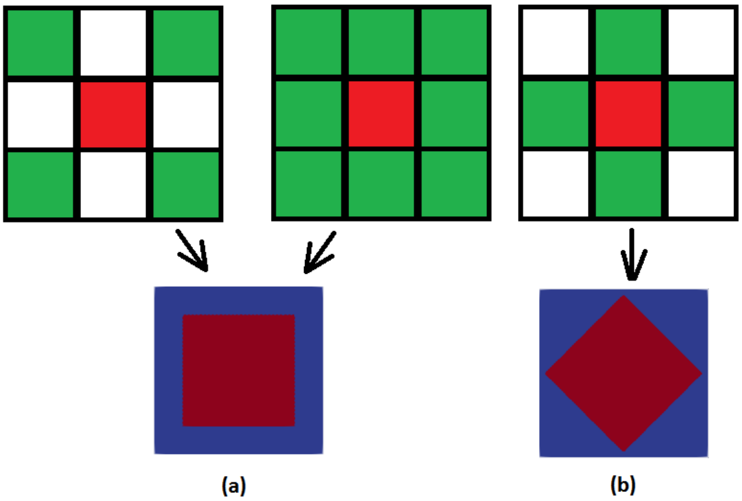


Figure 5.16: a) Rotated von Neumann and Moore neighbourhoods lead to a square grain growth, b) von Neumann leads to a rhombic grain growth.

rhombus obtained in a simulation with Neumann neighbourhood.

Considering the fact that $Area$ represents velocity multiplied by time, this formula will lead to the following:

$$error = \frac{v_{rhombus}t - v_{circle}t}{v_{circe}t}, \tag{5.11}$$

or

$$error = \frac{v_{rhombus} - v_{circle}}{v_{circe}}, \tag{5.12}$$

The grain boundary moves one cell forward per iteration with a velocity $\frac{dr}{dt}$. However, the change of area of the grain for the ideal round growth and the rhombic in the simulations are different. The change of areas $\frac{dS}{dt}$ for the rhombus and the circle can be written as follows:

$$\frac{dS}{dt}_{circle} = S_{t+1} - S_t = \pi r^2 - \pi(r+1)^2 = 2\pi r + \pi, \tag{5.13}$$

$$\frac{dS}{dt}_{rhombus} = 2(r+1)^2 - 2r^2 = 4r + 2, \tag{5.14}$$

For the Moore type neighbourhood, when a cell grows as a square change of area will be as follows:

$$\frac{dS}{dt}_{square} = 4(r+1)^2 - 4r^2 = 8r + 4. \tag{5.15}$$

Reversing back to the velocity variable leads to

$$\frac{dr}{dt}_{circle} = \sqrt{\frac{1}{\pi}\frac{dS}{dt}_{circle}} = \sqrt{\frac{2\pi r + \pi}{\pi}} \tag{5.16}$$

$$\frac{dr}{dt}_{rhombus} = \sqrt{\frac{1}{\pi}\frac{dS}{dt}_{rhombus}} = \sqrt{\frac{4r + 2}{\pi}}, \tag{5.17}$$

$$\frac{dr}{dt}_{square} = \sqrt{\frac{1}{\pi}\frac{dS}{dt}_{square}} = \sqrt{\frac{8r+4}{\pi}}, \tag{5.18}$$

Substituting these variables in Eq. 5.12 leads to the following errors for the grain rhombic and the square growth:

$$error_{rhombus} = \frac{v_{rhombus}}{v_{circle}} - 1 = \frac{\sqrt{4r+2}}{\sqrt{2\pi r+\pi}} - 1 = \sqrt{\frac{2}{\pi}} - 1 \approx -0.2 \tag{5.19}$$

$$error_{square} = \frac{v_{square}}{v_{circle}} - 1 = \frac{\sqrt{8r+4}}{\sqrt{2\pi r+\pi}} - 1 = \frac{2}{\sqrt{\pi}} - 1 \approx 0.128 \tag{5.20}$$

As can be seen from Fig. 5.17, these values agrees with the errors of the corresponding simulations of the grain growth. The error approaches the value of 0.128 for simulations with the rotated Neumann (Eq. 5.20) and -0.2 in simulations with the Neumann neighbourhood (Eq.5.19) with the decrease of the cell size .

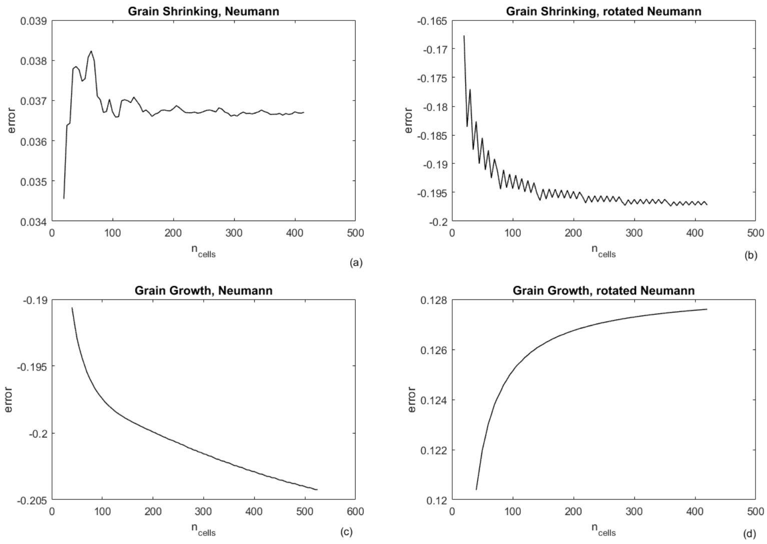


Figure 5.17: Errors of the grain growth and shrinkage if 2D disk with a constant velocity simulated using Neumann and rotated Neumann.

5.2.2 Adjusted velocity

The following straightforward method first came to mind to eliminate the errors caused by the square grid. As has been shown in Fig. 3.15 for the 2D simulations of the grain shrinkage, the error approaches $error_{d_{cell} \to 0} = 0.037$.

Considering the fact that the formula used for calculating the error was based on comparing the areas:

$$error = \frac{Area_{sim} - Area_{an}}{Area_{an}}, \tag{5.21}$$

one can conclude that the error reflects deviation in the velocities. Therefore, adjusting the velocity in the model via the multiplication by 1.037: $v_{adj} = 1.037\,v$ should shift the value of the error and make it $\lim_{d_{cell} \to 0} error(d_{cell}) \approx 0$.

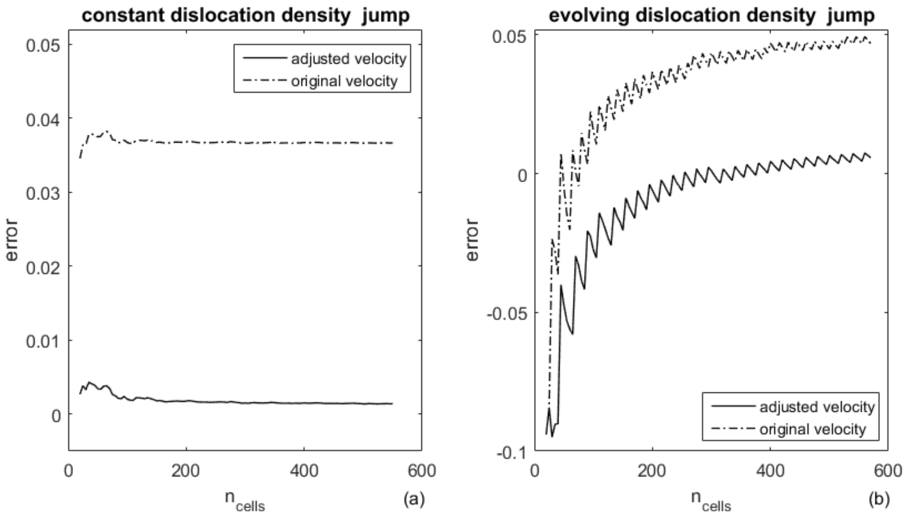


Figure 5.18: Errors of simulations of the grain shrinkage due to constant (a) and evolving (b) dislocation jump with decrease of the cell size, calculated using adjusted velocity and compared with original algorithm.

The results of the implementation of this procedure are shown in Fig. 5.18. Fig. 5.18a shows an improvement for the constant dislocation density jump, as was expected. The simulations for the evolving dislocation density jump with the same adjusted parameter has been carried-out and results are shown in Fig. 5.18b. As can be seen from the graph, the suggested adjustment of the velocity has also reduced the errors for the evolving dislocation density. It is worth noting that the updated CAS has been used, because the fixed CAS caused errors even in 1D, hence the errors on 2D for the fixed CAS are caused not only by the square grid, but also by the time increment.

5.2.3 Corrected Moore-type neighbourhood

As has been shown earlier, the error obtained during simulations for the grain shrinkage with a constant velocity is directly connected with the grid and the choice of neighbourhood. However, it is possible to obtain the different shapes of grains, not only the rhombus and the square, by using a combination of these Neumann and rotated Von Neumann neighbourhoods. Janssens first introduced the corrected Moore-type neighbourhood [108]. This technique is not commonly used, and how the usage of this type of neighbourhood can reduce the errors induced by the grid has never been studied before. In the present chapter, the corrected Moore-type neighbourhood has been applied to attempt to reduce the errors caused by the square grid.

The corrected Moore-type neighbourhood is a hybrid form of the von Neumann neighbourhood and the rotated von Neumann neighbourhood. At each iteration, one of these two neighbourhoods is used with the probabilities P_1 and P_2 respectively. The shape of the linearly growing grain can be obtained almost a being round. It is worth noting that this method is different from the Margolus neighbourhood [123], which is required switch of the sampled subzones of neighbourhood. This technique will be described later.

5.2.3.1 Application of the corrected Moore-type neighbourhood for a newly developed CA technique for grain growth simulation

The equation for fraction for i neighbour at iteration n is written as follows:

$$f_i^n = f_i^{n-1} + v_i \Delta t_n, \tag{5.22}$$

where v_i is the velocity of neighbour i in the direction to the cell, Δt_n is the time increment at the iteration n.

The application of the corrected Moore-type neighbourhood modifies this equation as follows:

$$f_i^n = f_i^{n-1} + s_i v_i \Delta t_n, \tag{5.23}$$

$$f_j^n = f_j^{n-1} + s_j v_j \Delta t_n, \tag{5.24}$$

where i and j are cells belonging to the Neumann and the rotated Von Neumann neighbourhoods respectively and s_i, s_j are defined as follows:

$$s_i = \begin{cases} 1, & \text{if } rand < P_1. \\ 0, & \text{otherwise.} \end{cases} \tag{5.25}$$

$$s_j = \begin{cases} 1, & \text{if } rand < P_2. \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

where P_1, P_2 are the probabilities that neighbours connected by faces – P_1 and edges – P_2 are active. This method can reduce the corrected Moore-type neighbourhood to the Neumann neighbourhood using $P_1 = 1$, $P_2 = 0$, to the Moore neighbourhood $P_1 = 1$, $P_2 = 1$, and to the rotated Von Neumann: $P_1 = 0$, $P_2 = 1$.

Fig. 5.19a shows the results of the shapes obtained during grain growth simulations with three different pair of values P_1, P_2. As can be seen from the figure, pairs $(P_1\ , P_2) = (0, 1)$ and $(P_1\ , P_2)(1, 1)$, corresponding to the rotated Von Neumann and Moore neighbourhood respectively, lead to the quadratic shape that resembles the initial square grain. The Von Neumann neighbourhood with the values $(P_1\ , P_2) = (0, 1)$ leads to the rhombic shape. The simulated kinetics for these three pairs of (P_1, P_2) are compared with the analytical solution and are shown in Fig. 5.19b. As can be seen from this graph, the different combinations of P_1 and P_2 lead to the different kinetics of the grain boundary movement. Let us assume that there is a pair $(P_1\,,P_2)$ that will lead to a spherical shape and the closest simulated kinetics the analytical solution.

A series of simulations of the grain growth initially was implemented with the various combinations $(P_1\,,P_2)$. The initial radius $r_0 = 1$ cell, the final radius is 40 cells, the cell size $d_{cell} = 3.1250\times10^6 m$. The values of probabilities are $P_1 = [0.01 : 0.05 : 0.96]$, $P_2 = [0.01 : 0.05 : 0.96]$. To estimate how far the simulation lies from the analytical solution the errors were estimated by calculating the deviations of the area under the lines from the analytical, as has been described in Chapter 3. The surface plot of the absolute errors against $(P_1\,,P_2)$ is shown in Fig. 5.20a. Fig. 5.20b shows the $X - Y$ view. As can be seen from this figure, the surface represents a gutter and there is no distinguishable minimum error. Also, the errors at the $P_2 = 1$ are all equal, regardless of the value of P_1.

The minimum value in the calculated data set is equal 2.0331×10^{-4} and corresponds to the pair $(P_1\,,P_2) = (0.86, 0.31)$. For comparison, the errors for the known types of neighbourhood estimated by this method are following $error_{(P_1\,,P_2)=(1,0)} = -0.1885$ (Neumann), $error_{(P_1\,,P_2)=(0,1),(1,1)} = 0.1244$ (rotated Neumann, Moore). The shape obtained in the simulations with the values $(P_1\,,P_2) = (0.86, 0.31)$ is shown in Fig.5.21a. As can be seen from this figure, it is possible to obtain the shape of grain close to the circle using the corrected Moore neighbourhood. It also helps to reduce the deviation of the simulated kinetics from the theory as can be seen from 5.21 b.

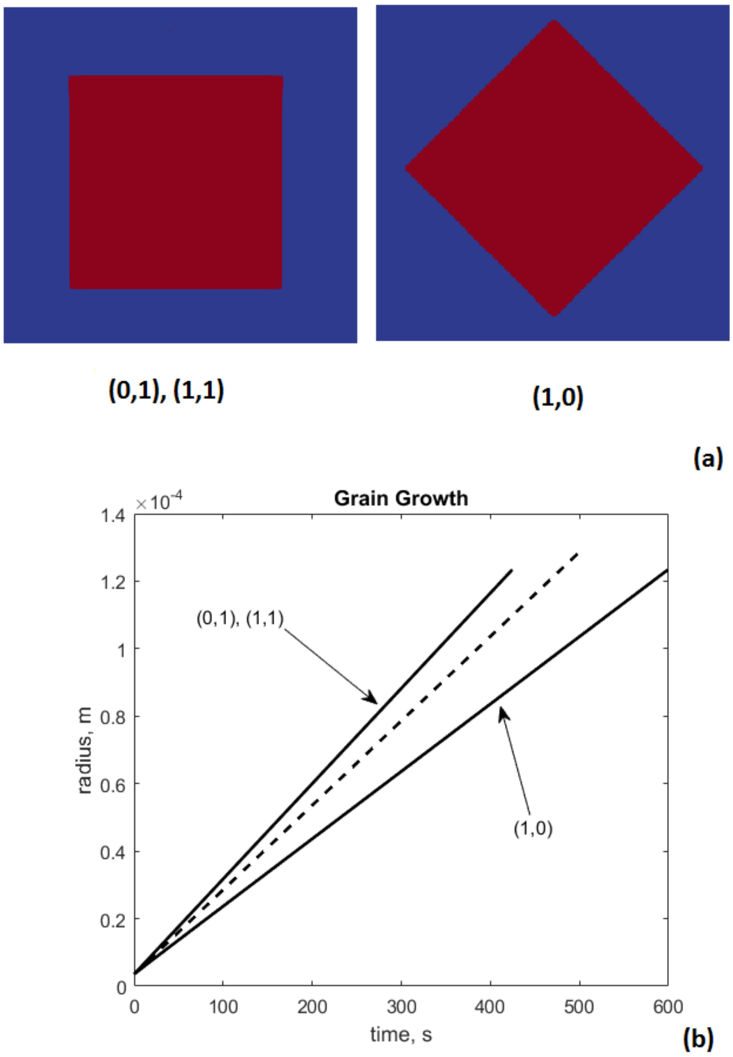


Figure 5.19: The shapes of grain obtained during simulations with different values of probabilities P_1 and P_2 (a) and the corresponding simulated kinetics of the grain growth compared with the theory (b). Simulations performed on the initial grain with radius $r_0 = 1$ cell and final radius $r = 40$ cells. The cell size is $d_{cell} = 3.125 \times 10^{-6}$ m.

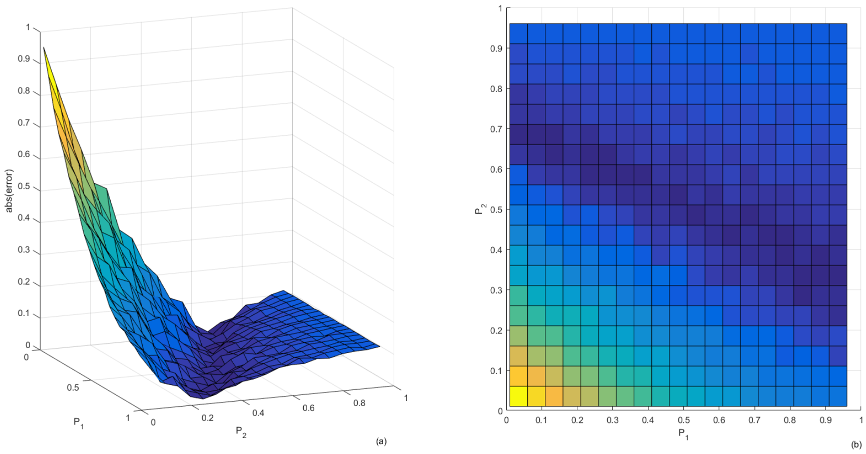


Figure 5.20: Surface plots of errors for simulations of a single grain growth at the different values of probabilities P_1 and P_2. The cell size is $d_{cell} = 3.125 \times 10^{-6}$ m or final radius $r = 40$ cells.

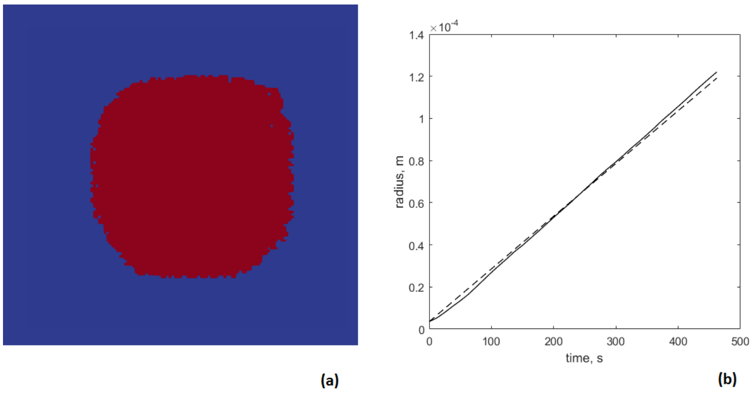


Figure 5.21: The shape of the grain obtained during simulations with values $(P_1\ , P_2) = (0.86, 0.31)$ (a) and the corresponding simulated kinetics of the grain growth (solid line) in comparison with theory (dashed line).

5.2.3.2 The corrected Moore-type neighbourhood applied for the in a newly developed CA technique for grain shrinkage simulation

The 2D CA simulations for dislocations with a constant velocity that is used with Neumann neighbourhood showed insignificant deviations for the grain shrinkage in comparison with the grain growth (see Fig. 5.13). The figure also shows that the maximum deviation of the simulated radius from the analytical occurs at the half time of shrinkage. However, these deviations become important in the simulations that involve the evolving dislocation density (see Fig. 5.14, Fig. 4.11). As has been discussed in Chapter 3, such deviations are caused by the square grid (see Fig. 3.14). To see how the corrected Moore-type neighbourhood can improve the simulations of the grain shrinkage, a procedure similar to the one described above has been implemented. It is worth noting, that the principle of the time symmetry in CA simulations assumes that the parameters (P_1, P_2) should be the same as in the grain growth. This means, that if one perfect pair has been found for the grain growth, it should work for the grain shrinkage, otherwise additional improvements of the technique maybe required.

The surface plots for grain shrinkage with the different probabilities (P_1, P_2) are shown in Fig. 5.22. As can be seen there is an area where the errors become very large, so this plot has been redrawn without visualization of any errors greater than 1 (see Fig. 5.23). As can bee seen from this figure, the error surface plot also includes a gutter, but it is shifted from the one for the grain growth (see Fig. 5.20b), which means that even the application of the corrected Moore will not lead to the fully symmetrical simulations.

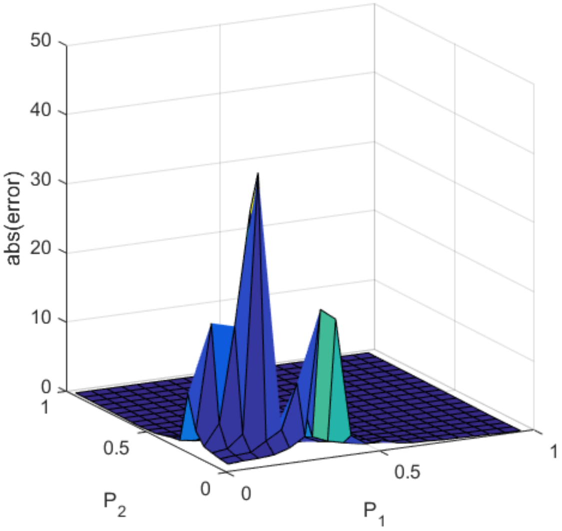


Figure 5.22: Surface plots of errors for simulations of a single grain shrinking at the different values of probabilities P_1 and P_2. The cell size is $d_{cell} = 3.125 \times 10^{-6}$ m or final radius $r = 40$ cells.

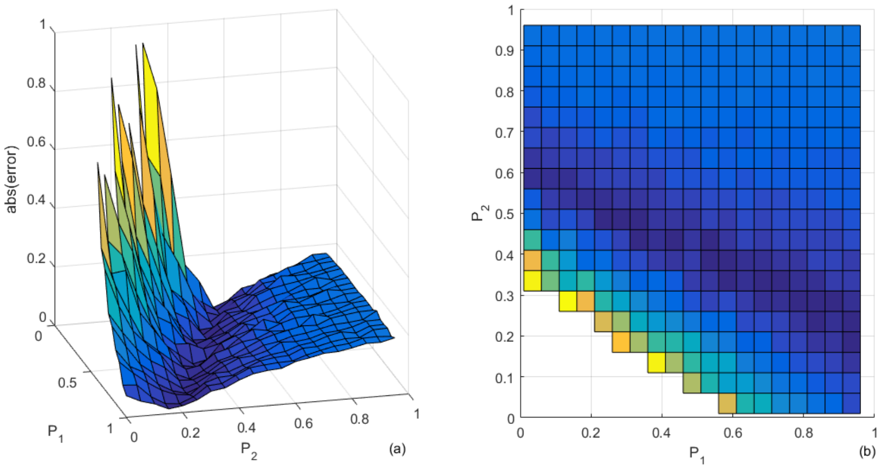


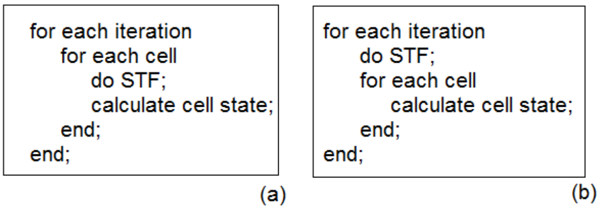
Figure 5.23: Surface plots of errors for simulations of a single grain shrinkage at the different values of probabilities P_1 and P_2. The cell size is $d_{cell} = 3.125 \times 10^{-6}$ m or final radius $r = 40$ cells.

5.2.3.3 Other Moore-type neighbourhood variations

The short summary of the variations the Moore-type neighbourhood is described in terms P_1, P_2 and possible algorithms. Let us call the calculation of the variables s_i, s_j via Eq. (5.25), Eq. (5.26) as a procedure STF (state transformation function). The implementation of this procedure can be carried-out in two different places in the code: at each time step or at each time step and for each cell (see Fig. 5.24).

Extreme values of P_1, P_2: The extreme values P_1, P_2 correspond to the known neighbourhoods: von Neumann ($P_1 = 1, P_2 = 0$), rotated von Neumann – ($P_1 = 0, P_2 = 1$) and Moore neighbourhoods ($P_1 = 1, P_2 = 1$). The ways in which to calculate s_i, s_j do not matter too much. The shapes of the grains obtained by these neighbourhoods have been shown in Fig. 5.19.

STF calculated at each iteration for each cell with other values of P_1, P_2 with: The way of implementation SFT, shown in Fig. 5.24 a, with the values of P_1, P_2 different from the extreme leads the various shapes of the grain (see example in Fig. 5.21). Fig. 5.25 illustrates the influence of each variable P_1 and P_2. The increase of P_1 leads to a smoother shape. P_2 is responsible for the transition from the rhombic to the square shape. $P_2 = 1$ leads to a square



```
for each iteration
    for each cell
        do STF;
        calculate cell state;
    end;
end;
```

```
for each iteration
    do STF;
    for each cell
        calculate cell state;
    end;
end;
```

Figure 5.24: Different ways to place STF in code: calculate s_i and s_j at each time step and for each cell (a); only at each time step (b).

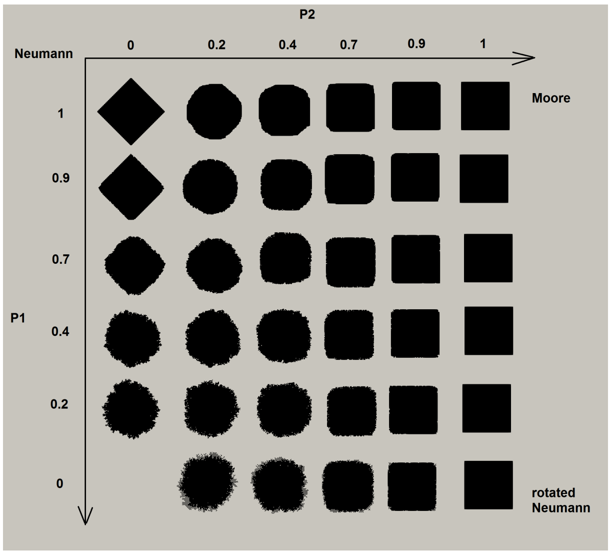


Figure 5.25: Variation of the grain shapes. STF is calculated at the each iteration and for each cell with the different values of P_1, P_2, see procedure in Fig. 5.24 a.

grain with any value of P_1.

STF calculated at each iteration with other values of P_1, P_2: The variables P_1, P_2 are responsible for the transition of non-symmetrical shapes: rhombus → octagon → square. Fig. 5.26 illustrates some forms of the grain which were obtained with different values of P_1 and P_2.

STF calculated at each iteration with other values random P_1, P_2: This method allows the system to choose the variables of P_1 and P_2 randomly. It can be implemented with the different possible ways as well, as shown in Fig.5.27. The code shown in Fig.5.27a leads to octagons, while two other codes allow to obtain various fringed shapes with slightly small differences between (b) and (c) algorithms. Although, this list of variations infinite, we will not investigate this aspect further.

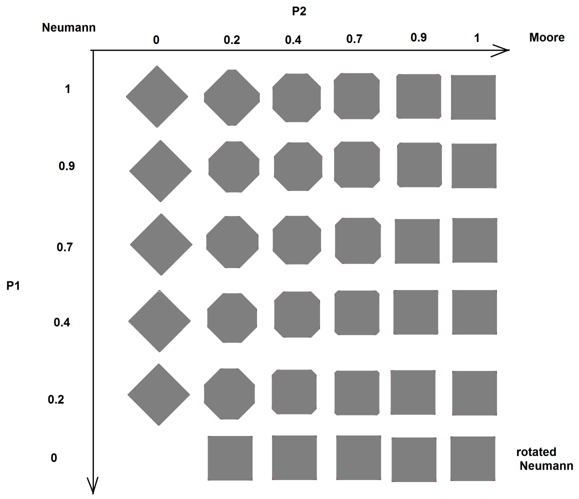
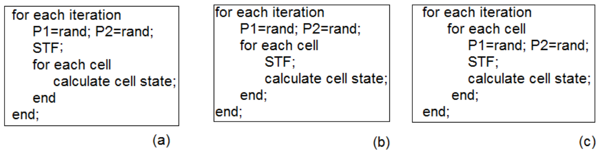


Figure 5.26: Variation of the grain shapes. STF is calculated at the each iteration with the different values of P_1, P_2, see procedure in Fig. 5.24 b.



```
for each iteration
    P1=rand; P2=rand;
    STF;
    for each cell
        calculate cell state;
    end
end;
```

(a)

```
for each iteration
    P1=rand; P2=rand;
    for each cell
        STF;
        calculate cell state;
    end;
end;
```

(b)

```
for each iteration
    for each cell
        P1=rand; P2=rand;
        STF;
        calculate cell state;
    end;
end;
```

(c)

Figure 5.27: P_1, P_2 are defined as random code, variations of the code.

5.2.4 Margolus-type neighbourhood

When investigating symmetry and reversibility it is impossible not to mention the Margolus-type neighbourhood. It is a block cellular automaton, where the grid of cells are separated into 2x2 squares called blocks. They are shifted at each iteration, therefore all 4 cells belong to the different block at each iteration. This type of neighbourhood led to the most interesting results. Although, the algorithm has been developed by the end of the research, the specific preliminary results will be described further.

5.2.4.1 Random Margolus-type neighbourhood

Let us rewrire Eq. (5.23) and Eq. (5.24) for each cell $i = 1..8$ within neighbourhood as follows:

$$\begin{cases} f_1^n = f_1^{n-1} + s_1 v_1 \Delta t_n, \\ ... \\ f_8^n = f_8^{n-1} + s_8 v_8 \Delta t_n, \end{cases} \tag{5.27}$$

where f_i^n is a distance fraction at the time step n of the cell i moving with the velocity v_i. The state transition variable s shows when the cell is active: $s_i = 1$ if active, $s_i = 0$ otherwise. The states transition variables are the functions of the randomly chosen switch numbers $W = 1..4$ (see Fig. 5.28), which can be formulated as follows:

$$\begin{cases} s_1 = w_e, s_2, s_4 = w_f & \text{if W=1} \\ s_3 = w_e, s_2, s_5 = w_f, & \text{if W=2} \\ s_8 = w_e, s_5, s_7 = w_f, & \text{if W=3} \\ s_6 = w_e, s_4, s_7 = w_f, & \text{if W=4} \end{cases} \tag{5.28}$$

As well as for the corrected Moore type neighbourhood, this new type of Margolus includes a few modifications (for example w_e,w_f can be different) and it requires further tests. However, it has been found that a pair $w_e = 1/3$, $w_f = 1/3$ gives almost a round shape of the growing grain without the fringe (see Fig.5.29).

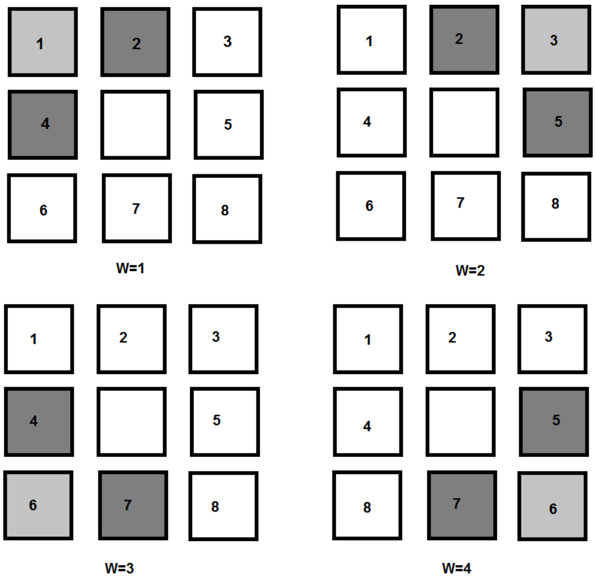


Figure 5.28: The active neighbourhood depending on the W.

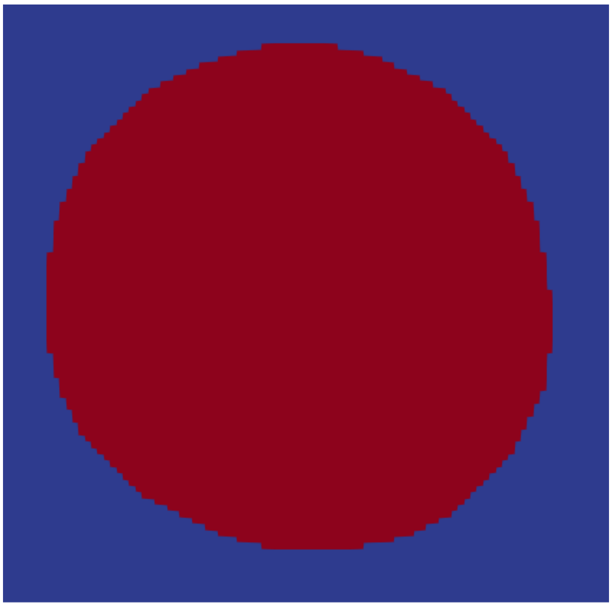


Figure 5.29: The grain obtained with the new random Margolus neighbourhood.

5.3 Summary

The CA techniques used in the present research are time non-reversible. This means that the change of sign in the velocity will not lead to the time reverse process, for example from shrinkage to growth. The non-symmetrical grid is the reason for the non-symmetry in time of the linear growth, whereas the reason for the non-reversible curvature driven growth is the CA construction itself.

However, it has been proven by Kari that the reversibility 2D CA is indecidable. There are few possible ways to eliminate the errors, by designing the neighbourhood that allows the grain to grow in the form of a circle. These methods include the corrected Moore Neighbourhood, but it is possible to obtain a circle at the expense of the smoothness of the boundaries. Another discovered method is a random Margolus-type, which gives promising results and is recommended for future studies.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In the present research, Celluar Automata (CA) was considered as an approach to solve non-linear differential equations describing grain boundary movement. Motivated by the lack of CA models capable of considering triple junctions, a novel deterministic CA technique has been developed which considers the movement of the neighbours instead of only the central cell. For all considered equations the analytical solution of the single grain shrinkage has been compared with the simulations, and the efficiency of a developed CA techniques has been proven. The comparison made via the error calculations, to test their convergence. The method of the error calculation of the errors has been described in detail Chapter 3.

A developed CA technique showed the ability to solve equations that describe the grain boundary motion which is driven by the different forces. Chapter 3 started from the simple equations of the constant velocity caused by dislocation density jump across grain boundary. The effects of the cell size, the dimensions of the calculating area and the time increment have been discussed. It has been shown that the non-symmetrical square grid, which has been used in the present research, decreases the accuracy of the simulated solutions on the 2D simulations and leads to an even greater deviation on the 3D area.

A newly developed CA technique has been constructed to be reflective of time and space. Simulated results do not require further scaling. In the present research two different interpretations of Calculation Automata Steps (CAS) are introduced , which were called "fixed" and "updated". Their influence on a developed CA technique has been compared during the present research. The fixed CAS is a time increment which is calculated at the beginning of simulations and equals to a distance that a cell can pass with the maximum velocity among all possible values that can arise during the simulation. The time increment for the updated CAS is calculated at each iteration in the same manner as for the fixed CAS, but using the maximum velocity among only those that occur in the current step. This means that at each iteration at least one cell will update its state. For the single grain simulation of shrinkage with a constant velocity the choice of this two algorithms does not make a difference in the resulted kinetics. However, it influences the results in the more complex cases, such as the grain boundary movement in the processes of dynamic recovery. In this situation the dislocation density evolves with time and hence the driven force. The fixed CAS has shown significantly greater deviations of the 1D simulations from the theory than the updated CAS. The further decrease of both of these types of time increments as well as the decrease of the cell size led to the increase in accuracy. However, the cell size did not improve the 2D and 3D calculations, because the deviations have been caused by the square grid.

The most challenging task of the present research was the adoption of the common methods for curvature calculations to the new technique. The parameters used in the known formula for curvature such as A and $Kink$ have been tested. The following principle has been stated, that the parameter A, which is a scaling parameter, should be consistent with the cell size and the time increment. The reduction in the time increment significantly slowed down the calculations, so it has been studied briefly. However, this condition is less important than the cell size. Whereas the time increment is a computation parameter, the cell size in fact represents the size of a grain, which is a dynamic parameter and changes during the grain growth simulations. The parameter $Kink$, that represents the flat boundary, has been tested at four different values. The results obtained in the various series of computational experiments are as follows: 1) The consistency of A varies with the change of the value $Kink$ and type of CAS. The best consistency of A has been found using $Kink = 12$ and the fixed CAS. 2) The shape of the simulated curve strongly depends on the parameter $Kink$. The smallest errors were observed for $Kink = 14$ and the fixed CAS. The error approaches zero with the decrease of the cell size, but not in all cases: $Kink = 15$ with the updated CAS showed apparent trend towards an increase of the error with the decrease of the cell size. This is a serious disadvantage of this pair of parameters, considering the stated principle of the consistency of the parameters with the change of the cell size. In the other hand, many simulations are usually conducted on the large resolution in order to save on computational costs. Considering this fact, it is important to have smaller fluctuations at the large cell sizes. Here, the updated CAS combined with $Kink = 15$ gives the best result. Therefore, it is important to consider the average grain sizes for the particular simulations in order to choose the appropriate parameters.

Chapter 4 made an attempt to solve differential equations, that do not even have explicit analytical solutions. The grain shrinkage under two driving forces acting simultaneously has been simulated with a developed CA technique in Chapter 3. It has been shown that a developed technique is capable of solving these equations. However, significant deviations have also been observed, which have been caused by the regular non-symmetry grid. The parameters obtained for the curvature simulations at the particular cell size have been used without any changes for the combined equations simulated at the corresponding cell size. That allowed to investigate how the parameter $Kink$ and the type of CAS influence the accuracy for the coupled driving forces. The main results are as follows:

1) The simulations with $Kink = 12$ and the updated CAS give the closest to the theoretical solution.

2) The shape of the simulated curves differs from the analytical solution within 5% and less and the shape does not become closer to the analytical with the decrease of the cells size.

3) The best time of the grain shrinkage has been observed as being different from the analytical solution within 10% for the updated CAS and $Kink = 12$. However, for almost all cases the tendency to increase errors with the decrease of the cell sizes has been observed.

It is believed that the one of the possible reasons that induce such big errors is the square grid. The usage of the grid with a higher order of symmetry, such as hexagonal, could potentially improve the accuracy.

In contrast to the commonly used probabilistic CA, a developed technique is deterministic. The probabilistic version of the suggested CA has been developed alongside the deterministic approach. The analysis of parameters used in the model has also been carried out similarly to the deterministic case. Although PCA showed big standard deviations, it also improved performance characteristics, such as accuracy of the prediction of time of shrinkage and the shape of curves. Therefore, it is recommended to continue the work with the probabilistic model.

The problems of symmetry were discussed in Chapter 5. It has been shown that the CA technique is non-reversible in time, which is caused by the regular grid and the CA approach for the calculations of the curvature itself. The development of the time-reversible CA for curvature was not the aim of the present research, so it was treated only as a simple observation. An attempt has however been made to build a model that is capable of reductions the effects of the square grid. A newly developed CA has been modified to include the Corrected Moore-type neighbourhood. This is a probabilistic way of choosing the neighbourhood which helps to eliminate the effects of the non-symmetrical grid.

6.2 Future Recommended Work

The results of the present research showed a potential for use of a developed CA technique. Therefore, the following suggestions are recommended for future research:

1. Prior to applying a developed CA technique to the multigrain simulations it recommended to test it on a 3D single grain in order to establish the parameters for curvature calculations.

2. It is recommended to carry out simulations on a hexagonal grid or to use the probabilistic approach of neighbourhood in order to eliminate the influence of the non-symmetrical grid.

3. The suggested probabilistic choice of neighbourhood requires further

testing. It is recommended to choose several pares of values P_1, P_2 which lie on the minimum level of errors plot, then to run the simulations with the chosen parameters to simulate the grain shrinkage due to the evolving dislocation density.

4. It is recommended to test the method of probabilistic choice of neighbourhood on the curvature driven simulations in order to have a better understanding of how the symmetry of grid influences the consistency of time scaling parameter.

5. Developed CA algorithm is ready to use in multigrain simulations as it computes grain motion in triple junctions inherently. Once single grain model is validated with theory it will become a powerful instrument for simulations from first principles, allowing to predict complex system dynamics (different systems, not only grain growth and recrystallization), where the governing equations and main properties of system are known.

It is worth noting that for multigrain the formulas for calculating the misorientation are given in Appendix B, which has also been prepared during the current research.

Appendix A

CA rules

In contrast to the case of the single grain, the multigrain structure contains triple junctions – points where grain boundaries of three different grains meet (see Fig. A.1). The cell at the boundary of the single grain can take only one of two possible states, namely when the curvature is positive $k > 0$, the cell changes its state from grain 2 to grain 1 with the probability k/k_{max} [1]. At the triple junctions, the cell has a neighbourhood of three grains, and when the curvature is positive, the cell should change state from grain 1 to grain 2 or grain 3. Janssens [1] suggested to simply choose those grains which have the bigger fraction in the neighbourhood. However this approach is not effective in the presence of other driving forces, for example when there is dislocation density jump. Hence, in the present work the new algorithm has been developed in order to take into account uncertainty at the triple junctions.

Fig. A.2 shows the example of the grid of cells whose states correspond to the three different grains 0, 1 and 2 and the grain boundaries are marked via the green line. The cells A, B and C lie at the triple junctions. The

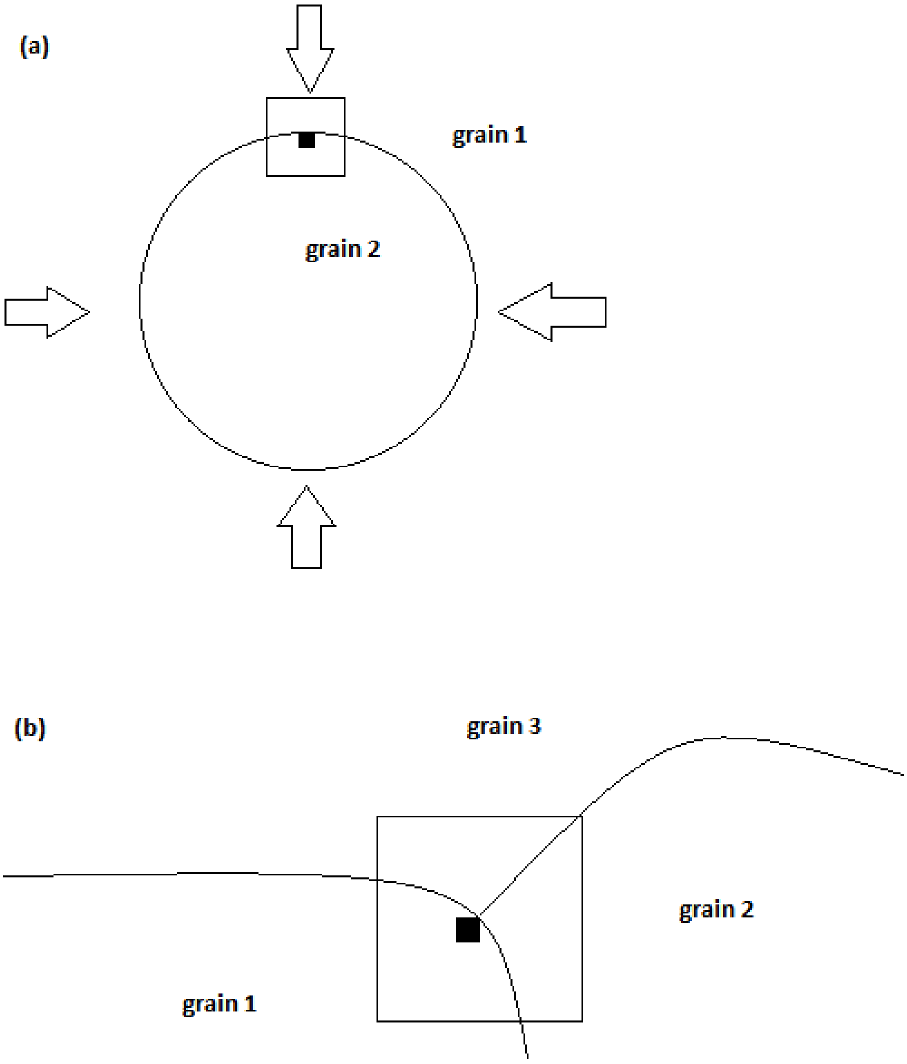


Figure A.1: The cell (black square) is situated on the grain boundary between two grains (a), and at the triple junction (b).

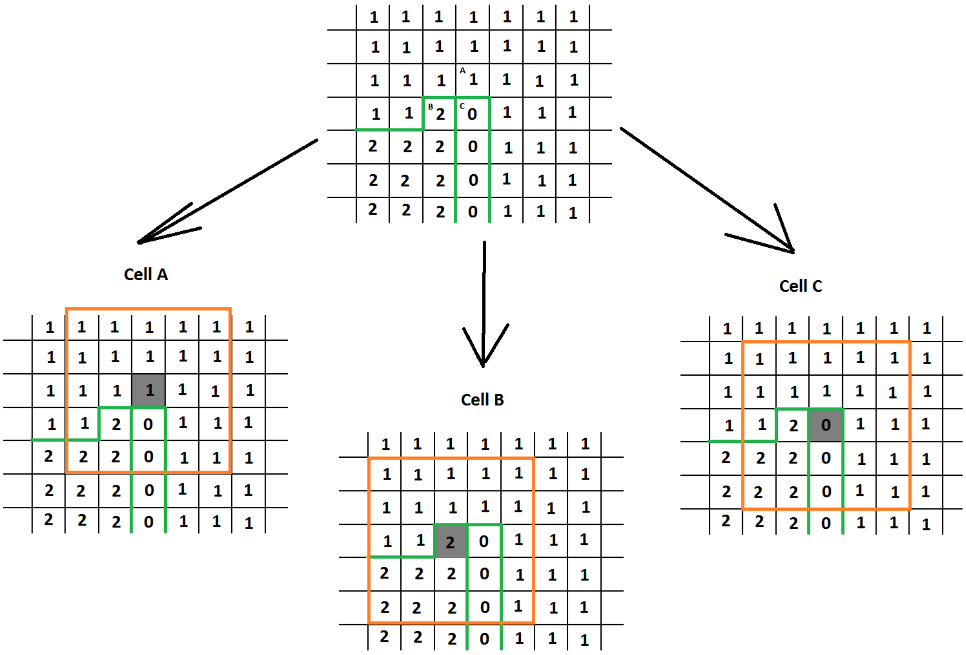
corresponding 5×5 neighbourhoods for each cell A, B and C are shown in Fig. A.2 with orange squares. Cell A has a negative curvature $k_A = Kink - N_1 = 15 - 20 = -5$, hence, according to the old method [1] cell A keeps its state in the next iteration, the curvature of the cell B is $k_B = Kink - N_2 = 15 - 7 = 8$ and for cell C is $k_C = Kink - N_0 = 15 - 3$, and as they are both positive, then cells B and C will change their states in the next iteration.

In order to decide to which state they should change the state a new approach suggests considering in the first place neighbours instead of the central cells. The neighbouring cells are moving towards or away from the central cell with certain velocities Δv and passing distances $\Delta r = \Delta v \Delta t$ by the time increment Δt (which can be fixed or different at each iteration). When the distance becomes greater or equal the cell size, the central cell changes its state to the state of neighbouring cell with the biggest distance. If there are two or more equal distances, then the cell can change state with equal probabilities to one of those states. Similar ideas with a few differences were implemented by Lan et al. [34] for the curvature driven grain growth. The main differences is that Lan's equation included velocities in x and y directions separately and has been applied for the central cell, not the neighbours, as follows:

$$f = \frac{V_x dt}{dx} + \frac{V_y dt}{dy} - \frac{V_x V_y dt^2}{dxdy}, \tag{A.1}$$

However, the method was not described in full details in the paper [34] and as a result it was not possible to duplicate results.

Schematically, Fig. A.3 demonstrates how these approaches are different. On the left side, the velocities $v_{1..4}$ are velocities with which the grey cell is moving towards the corresponding neighbours, and the biggest value shows the direction of the movement for the grey cell, (if the maximum velocity is v_1, then at the next iteration the upper pink cell will become grey). However, this



1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	A 1	1	1	1
1	1	B 2	C 0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1

Cell A

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1

Cell B

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1

Cell C

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1
2	2	2	0	1	1	1

Figure A.2: Example of the triple junction: the grid of cells with states 0, 1 and 2, which represent belonging to the different grains. The cells neighbourhoudes for the cell A, B and C are shown in orange squares.

approach does not take into account the movements of the cells surrounding the upper pink cell and for example, if the yellow cell is moving faster with $u_1 > v_1$, then the decision to change the state of the pink cell into grey would be wrong, because it should become yellow. Therefore, the order of checking cells has a strong influence on the output. Whereas in the newly suggested approach (see Fig. A.3, on the right side) the velocities $u_{1..4}$ show how fast the neighbours are moving towards the cell, thus the cell will change its state to a state of the neighbouring cell with the highest velocity. If there are two or more cells that have maximum velocities, then a grey cell changes its state with equal probability into one of those states. If the velocity is negative, then

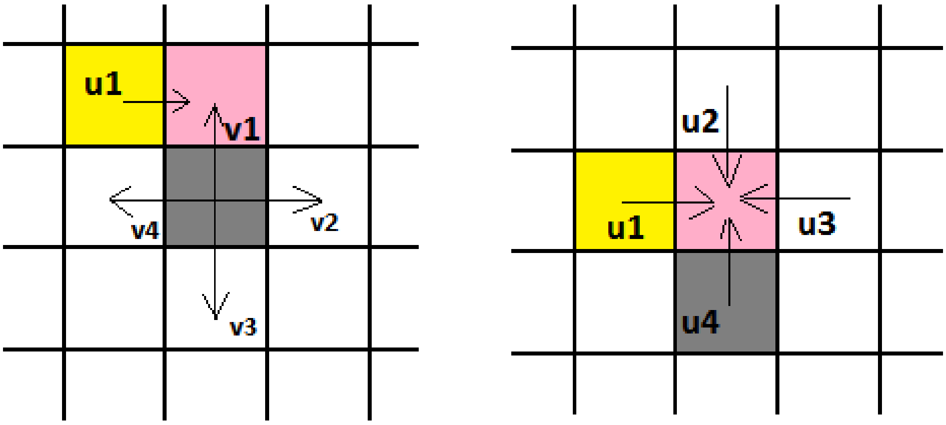


Figure A.3: Old approach considers influence of the central cell to neighbours (left), new approach considers influence of neighbours to the central cell (right).

it is assumed that it is zero and the neighbours are not moving.

Let us consider this approach on the example of the grid of cells shown in Fig. A.2. In this example, for simplicity the velocity simply equals the negative curvature. Hence, for each of 4 neighbours (colourful cells) the curvature has been calculated and the minimal curvature corresponds to the maximum velocity. According to the rules stated above, the cell A would remain its state "1", and the cells B, C would change at their states to the grain "1".

This approach has been adopted for real time simulations. Each cell contains s fraction variables corresponding to the each $i = 1..s$ neighbour ($s = 4$ for 2D Von Neumann neighborhood and $s = 8$ for 2D Moore neighborhood) (see Fig. A.5). The generalized form of equation of fraction for i neighbour at iteration n can be written as follows:

$$f_i^n = f_i^{n-1} + v_i \Delta t_n, \tag{A.2}$$

where v_i is the velocity of neighbour i in the direction to the cell, Δt_n is the

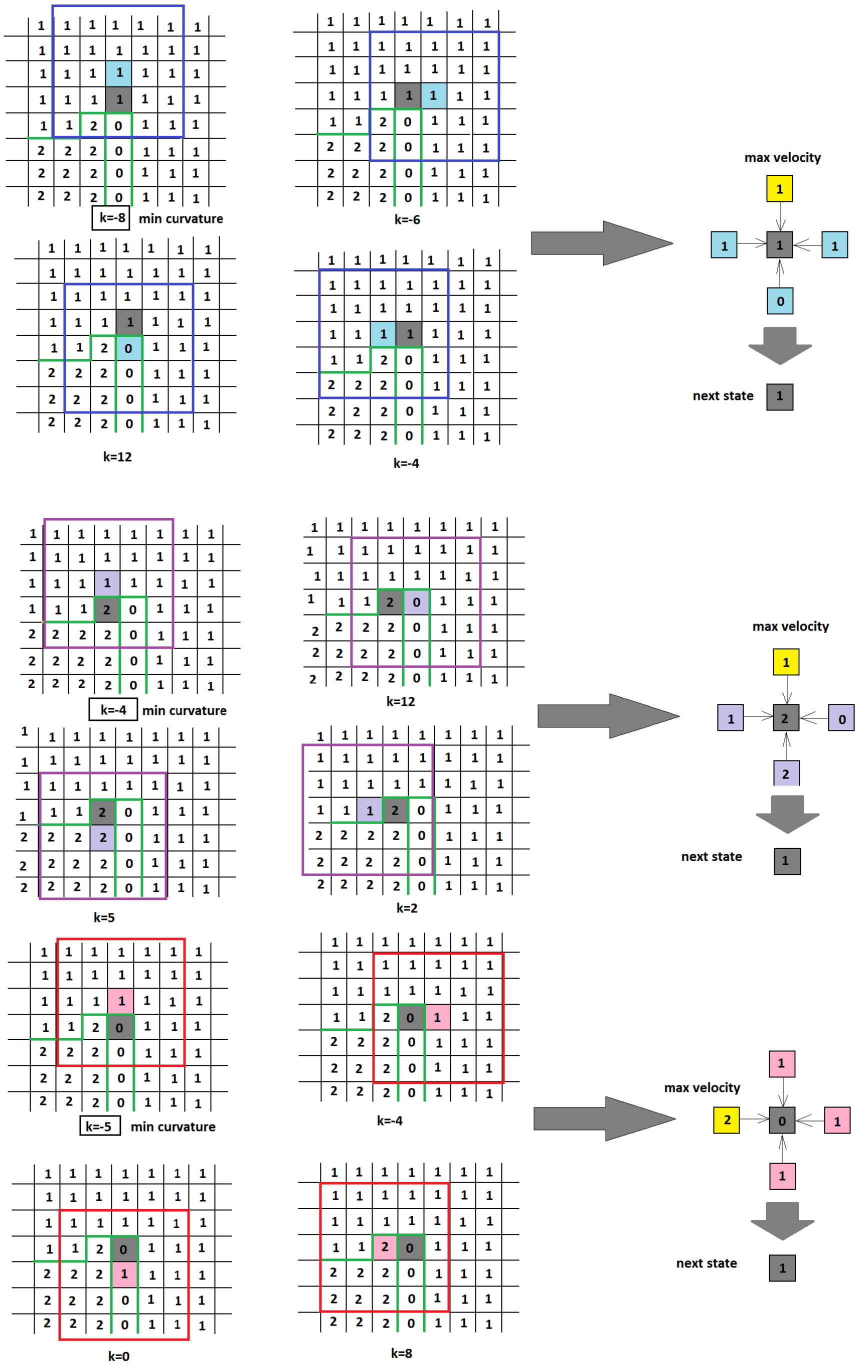


Figure A.4: The schematic representation of application the CA rules for calculation curvature for neighbours (colourful) of the central (grey) cell.

time increment at the iteration n. If the fraction of i neighbour at the iteration n becomes greater than 1 $f_i \geq 1$ then the cell changes its state to the state of the i neighbour and all fraction variables become zero: $f_{i=1..s}^{n} = 0$.

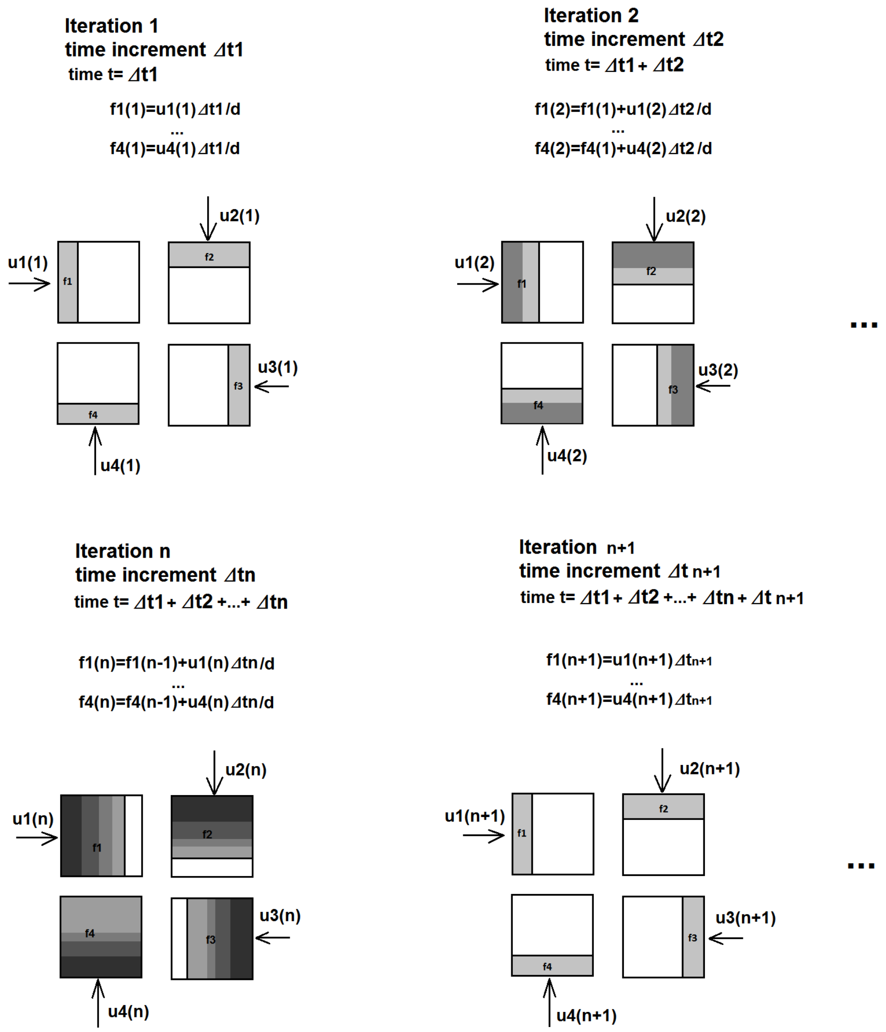


Figure A.5: Schematic representations for calculations of the fraction variables. At iteration n bottom neighbour fully passed cell of the size d and cell will change its state to the state of the bottom neighbour. After that all fractions become again zero and at iteration $n+1$ the fractions are calculated again.

Appendix B

Misorientation calculations

The initial orientations are obtained via the Bunge convention from three random numbers $\xi_1, \xi_2, \xi_3 \in [0..1]$ as follows:

$$\begin{aligned} \phi_1 &= 2\pi\xi_1 \\ \Phi &= \cos^{-1}(1-2\xi_2) \\ \phi_2 &= 2\pi\xi_3 \end{aligned} \tag{B.1}$$

The two corresponding rotation matrices R_i and R_j can be constructed for grains i and j with the given Euler angles $\{\phi_1^i, \Phi^i, \phi_2^i\}$ and $\{\phi_1^j, \Phi^j, \phi_2^j\}$ as follows:

$$R_i = \begin{pmatrix} \cos\phi_1^i \cos\phi_2^i - \sin\phi_1^i \sin\phi_2^i \cos\Phi^i & \sin\phi_1^i \cos\phi_2^i + \cos\phi_1^i \sin\phi_2^i \cos\Phi^i & \sin\phi_2^i \sin\Phi^i \\ -\cos\phi_1^i \sin\phi_2^i - \sin\phi_1^i \cos\phi_2^i \cos\Phi^i & -\sin\phi_1^i \sin\phi_2^i + \cos\phi_1^i \cos\phi_2^i \cos\Phi^i & \cos\phi_2^i \sin\Phi^i \\ \sin\phi_1^i \sin\Phi^i & -\cos\phi_1^i \sin\Phi^i & \cos\Phi^i \end{pmatrix} \tag{B.2}$$

$$R_j = \begin{pmatrix} \cos\phi_1^j \cos\phi_2^j - \sin\phi_1^j \sin\phi_2^j \cos\Phi^j & \sin\phi_1^j \cos\phi_2^j + \cos\phi_1^j \sin\phi_2^j \cos\Phi^j & \sin\phi_2^j \sin\Phi^j \\ -\cos\phi_1^j \sin\phi_2^j - \sin\phi_1^j \cos\phi_2^j \cos\Phi^j & -\sin\phi_1^j \sin\phi_2^j + \cos\phi_1^j \cos\phi_2^j \cos\Phi^j & \cos\phi_2^j \sin\Phi^j \\ \sin\phi_1^j \sin\Phi^j & -\cos\phi_1^j \sin\Phi^j & \cos\Phi^j \end{pmatrix} \tag{B.3}$$

The terms of rotation matrices and the misorienation matrix ΔR_{ij} can be defined as follows:

$$\begin{aligned} R_j &= \Delta R_{ij} R_i \\ \Delta R_{ij} &= R_j R_i^{-1}, \end{aligned} \tag{B.4}$$

where R_i^{-1} is the inverse matrix of R_i.

$$\Delta R_{ij} = \begin{pmatrix} \Delta R_{11} & \Delta R_{12} & \Delta R_{13} \\ \Delta R_{21} & \Delta R_{22} & \Delta R_{23} \\ \Delta R_{31} & \Delta R_{32} & \Delta R_{33} \end{pmatrix} \tag{B.5}$$

The scalar value of misorienation θ between two grains i and j is as follows:

$$\theta = \cos^{-1}\left[\frac{\Delta R_{11} + \Delta R_{22} + \Delta R_{33} - 1}{2}\right] \tag{B.6}$$

The effect of crystal symmetry leads to the following corrections. No two cubic lattices can be different by more than $\sim\ 62.8^\circ$ because cube crystals have 24 symmetries (full octahedral symmetry, O_{432}): if there are two cubes A (reference cube) and B (free to rotate), B can be rotated 24 definite ways to coincide with A: 1 is identity (1), other 23 are rotations: 1) about each of the 3 axes through the centres of opposite faces, through $\pi, \pi/2, 3\pi/2$ $(3 \times 3 = 9)$ 2) through π about each of the 6 axes through the midpoints of opposite edges $(1 \times 6 = 6)$, and 3) about each of the 4 axes through opposite vertices, through

	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$	$s_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$	$s_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$s_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \rightarrow R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$	$s_5 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$	$s_6 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$s_7 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \rightarrow R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$s_8 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$s_9 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$s_{10} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table B.1: Rotation matrices about the axes through the centres of opposite face.

$2\pi/3, 4\pi/3$ $(4 \times 2 = 8)$, hence $1 + 9 + 6 + 8 = 24$. Therefore, 24 symmetry operators need to be applied to the misorienation matrix Δq:

$$\theta = \min \left| \cos^{-1} \left\{ \frac{tr(O_{432}\Delta g) - 1}{2} \right\} \right|, \tag{B.7}$$

where the operator O_{432} is represented by the matrix for a rotation by an angle of θ about an axis in the direction of $\mathbf{u}$:

$$\begin{bmatrix} \cos\theta + u_x^2(1 - cos\theta) & u_x u_y(1-\cos\theta) - u_z \sin\theta & u_x u_z(1-\cos\theta) + u_y \sin\theta \\ u_y u_x(1-\cos\theta) + u_z \sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x \sin\theta \\ u_z u_x(1-\cos\theta) - u_y \sin\theta & u_z u_y(1-\cos\theta) + u_x \sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix} \tag{B.8}$$

1) Identity $(\theta = 0, \mathbf{u} = 0)$

$$s_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2) Rotation matrices about the axes through the centres of opposite face (see Table B.1);

$\begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$	$s_{11} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$	$s_{12} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$	$s_{13} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$	$s_{14} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
$\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}$	$s_{15} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}$	$s_{16} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

Table B.2: The rotation matrices about the 6 axes through the midpoints of opposite edges, $\theta = \pi$.

	$2\pi/3$	$4\pi/3$
$\begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}$	$s_{17} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$s_{18} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} -1/\sqrt{3} \\ -1/\sqrt{3} \\ 1\sqrt{3} \end{bmatrix}$	$s_{19} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$	$s_{20} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ -1/\sqrt{3} \end{bmatrix}$	$s_{21} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$s_{22} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1/\sqrt{3} \\ -1/\sqrt{3} \\ 1/\sqrt{3} \end{bmatrix}$	$s_{23} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}$	$s_{24} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$

Table B.3: The rotation matrices about each of the 4 axes through opposite vertices.

3) The rotation matrices about the 6 axes through the midpoints of opposite edges, $\theta = \pi$ (see Table B.2);

4) The rotation matrices about each of the 4 axes through opposite vertices (see Table B.3);

Consider two grains: the recrystallized R with the orientations defined by Euler angles $\{\phi_2^r, \Phi^r, \phi_2^r\}$ and the deformed D grain $\{\phi_2^d, \Phi^d, \phi_2^d\}$. There orientation matrices are as follows:

$$r = \begin{pmatrix} \cos\phi_1^r \cos\phi_2^r - \sin\phi_1^r \sin\phi_2^r \cos\Phi^r & \sin\phi_1^r \cos\phi_2^r + \cos\phi_1^r \sin\phi_2^r \cos\Phi^r & \sin\phi_2^r \sin\Phi^r \\ -\cos\phi_1^r \sin\phi_2^r - \sin\phi_1^r \cos\phi_2^r \cos\Phi^r & -\sin\phi_1^r \sin\phi_2^r + \cos\phi_1^r \cos\phi_2^r \cos\Phi^r & \cos\phi_2^r \sin\Phi^r \\ \sin\phi_1^r \sin\Phi^r & -\cos\phi_1^r \sin\Phi^r & \cos\Phi^r \end{pmatrix} \tag{B.9}$$

$$d = \begin{pmatrix} \cos\phi_1^d \cos\phi_2^d - \sin\phi_1^d \sin\phi_2^d \cos\Phi^d & \sin\phi_1^d \cos\phi_2^d + \cos\phi_1^d \sin\phi_2^d \cos\Phi^d & \sin\phi_2^d \sin\Phi^d \\ -\cos\phi_1^d \sin\phi_2^d - \sin\phi_1^d \cos\phi_2^d \cos\Phi^d & -\sin\phi_1^d \sin\phi_2^d + \cos\phi_1^d \cos\phi_2^d \cos\Phi^d & \cos\phi_2^d \sin\Phi^d \\ \sin\phi_1^d \sin\Phi^d & -\cos\phi_1^d \sin\Phi^d & \cos\Phi^d \end{pmatrix} \tag{B.10}$$

The misorientation matrix is $\Delta g_{dr} = g_r g_d^{-1}$ and we will notate it for simplicity as follows:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{B.11}$$

This matrix is orthogonal, which means that:

$$\Delta g_{dr} \Delta g_{dr}^T = g_r g_d^{-1} (g_r g_d)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.12}$$

The application of 24 symmetry operators s_i, $i = 1..24$ leads to 24 variations of R as follows:

$$R^1 = s_1 R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{B.13}$$

$$tr^1 = R_{11} + R_{22} + R_{33} \tag{B.14}$$

$$R^2 = s_2 R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ -R_{31} & -R_{32} & -R_{33} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \tag{B.15}$$

$$tr^2 = R_{11} - R_{32} + R_{23} \tag{B.16}$$

$$R^3 = s_3 R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ -R_{21} & -R_{22} & -R_{23} \\ -R_{31} & -R_{32} & -R_{33} \end{bmatrix} \tag{B.17}$$

$$tr^3 = R_{11} - R_{22} - R_{33} \tag{B.18}$$

$$R^4 = s_4 R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{31} & R_{32} & R_{33} \\ -R_{21} & -R_{22} & -R_{23} \end{bmatrix} \tag{B.19}$$

$$tr^4 = R_{11} + R_{32} - R_{23} \tag{B.20}$$

$$R^5 = s_5 R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{31} & R_{32} & R_{33} \\ R_{21} & R_{22} & R_{23} \\ -R_{11} & -R_{12} & -R_{13} \end{bmatrix} \tag{B.21}$$

$$tr^5 = R_{31} + R_{22} - R_{13} \tag{B.22}$$

$$R^6 = s_6 R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{11} & -R_{12} & -R_{13} \\ R_{21} & R_{22} & R_{23} \\ -R_{31} & -R_{32} & -R_{33} \end{bmatrix} \tag{B.23}$$

$$tr^6 = -R_{11} + R_{22} - R_{33} \tag{B.24}$$

$$R^7 = s_7 R = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{31} & -R_{32} & -R_{33} \\ R_{21} & R_{22} & R_{23} \\ R_{11} & R_{12} & R_{13} \end{bmatrix} \tag{B.25}$$

$$tr^7 = -R_{31} + R_{22} + R_{13} \tag{B.26}$$

$$R^8 = s_8 R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{21} & -R_{22} & -R_{23} \\ R_{11} & R_{12} & R_{13} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{B.27}$$

$$tr^8 = -R_{21} + R_{12} + R_{33} \tag{B.28}$$

$$R^9 = s_9 R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{11} & -R_{12} & -R_{13} \\ -R_{21} & -R_{22} & -R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{B.29}$$

$$tr^9 = -R_{11} - R_{22} + R_{33} \tag{B.30}$$

$$R^{10} = s_{10} R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{21} & R_{22} & R_{23} \\ -R_{11} & -R_{12} & -R_{13} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{B.31}$$

$$tr^{10} = R_{21} - R_{12} + R_{33} \tag{B.32}$$

$$R^{11} = s_{11} R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{31} & R_{32} & R_{33} \\ -R_{21} & -R_{22} & -R_{23} \\ R_{11} & R_{12} & R_{13} \end{bmatrix} \tag{B.33}$$

$$tr^{11} = R_{31} - R_{22} + R_{13} \tag{B.34}$$

$$R^{12} = s_{12}R = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{31} & -R_{32} & -R_{33} \\ -R_{21} & -R_{22} & -R_{23} \\ -R_{11} & -R_{12} & -R_{13} \end{bmatrix} \tag{B.35}$$

$$tr^{12} = -R_{31} - R_{22} - R_{13} \tag{B.36}$$

$$R^{13} = s_{13}R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{11} & -R_{12} & -R_{13} \\ R_{31} & R_{32} & R_{33} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \tag{B.37}$$

$$tr^{13} = -R_{11} + R_{32} + R_{23} \tag{B.38}$$

$$R^{14} = s_{14}R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{11} & -R_{12} & -R_{13} \\ -R_{31} & -R_{32} & -R_{33} \\ -R_{21} & -R_{22} & -R_{23} \end{bmatrix} \tag{B.39}$$

$$tr^{14} = -R_{11} - R_{32} - R_{23} \tag{B.40}$$

$$R^{15} = s_{15}R = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{21} & R_{22} & R_{23} \\ R_{11} & R_{12} & R_{13} \\ -R_{31} & -R_{32} & -R_{33} \end{bmatrix} \tag{B.41}$$

$$tr^{15} = R_{21} + R_{12} - R_{33} \tag{B.42}$$

$$R^{16} = s_{16}R = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{21} & -R_{22} & -R_{23} \\ -R_{11} & -R_{12} & -R_{13} \\ -R_{31} & -R_{32} & -R_{33} \end{bmatrix} \tag{B.43}$$

$$tr^{16} = -R_{21} - R_{12} - R_{33} \tag{B.44}$$

$$R^{17} = s_{17}R = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{31} & R_{32} & R_{33} \\ R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \tag{B.45}$$

$$tr^{17} = R_{31} + R_{12} + R_{23} \tag{B.46}$$

$$R^{18} = s_{18}R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \\ R_{11} & R_{12} & R_{13} \end{bmatrix} \tag{B.47}$$

$$tr^{18} = R_{21} + R_{32} + R_{13} \tag{B.48}$$

$$R^{19} = s_{19}R = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{31} & -R_{32} & -R_{33} \\ R_{11} & R_{12} & R_{13} \\ -R_{21} & -R_{22} & -R_{23} \end{bmatrix} \tag{B.49}$$

$$tr^{19} = -R_{31} + R_{12} - R_{23} \tag{B.50}$$

$$R^{20} = s_{20}R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{21} & R_{22} & R_{23} \\ -R_{31} & -R_{32} & -R_{33} \\ -R_{11} & -R_{12} & -R_{13} \end{bmatrix} \tag{B.51}$$

$$tr^{20} = R_{21} - R_{32} - R_{13} \tag{B.52}$$

$$R^{21} = s_{21}R = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{31} & -R_{32} & -R_{33} \\ -R_{11} & -R_{12} & -R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \tag{B.53}$$

$$tr^{21} = -R_{31} - R_{12} + R_{23} \tag{B.54}$$

$$R^{22} = s_{22}R = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{21} & -R_{22} & -R_{23} \\ R_{31} & R_{32} & R_{33} \\ -R_{11} & -R_{12} & -R_{13} \end{bmatrix} \tag{B.55}$$

$$tr^{22} = -R_{21} + R_{32} - R_{13} \tag{B.56}$$

$$R^{23} = s_{23}R = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} -R_{21} & -R_{22} & -R_{23} \\ -R_{31} & -R_{32} & -R_{33} \\ R_{11} & R_{12} & R_{13} \end{bmatrix} \tag{B.57}$$

$$tr^{23} = -R_{31} - R_{12} - R_{23} \tag{B.58}$$

$$R^{24} = s_{24}R = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} R_{31} & R_{32} & R_{33} \\ -R_{11} & -R_{12} & -R_{13} \\ -R_{21} & -R_{22} & -R_{23} \end{bmatrix} \tag{B.59}$$

$$tr^{24} = R_{31} - R_{12} - R_{23} \tag{B.60}$$

Leading therefore, to 24 different misorientation angles, among which the least should be chosen. The distribution for cubes with random orientations was obtained first by Mackenzie [124, 125], and sometimes the angle/density distribution plot is called the Mackenzie distribution or simply the Mackenzie plot.

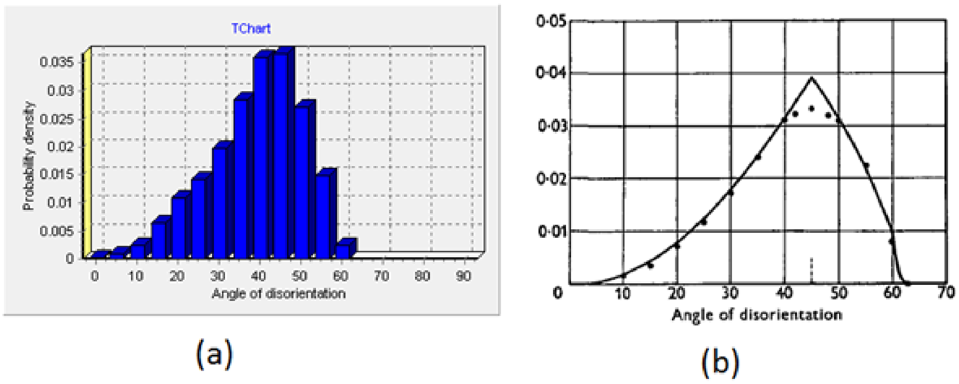


Figure B.1: Misorientation histogram calculated in this work (a) and by Mackenzie (b) for distribution of cubes with random orientations

.

The misorientation histograms derived by Mackenzie (1958) [125] and obtained in this work are represented in **Fig. B.1**. As can be seen, both plots have a peak near 45° and the limit is about 62°. The values of the probability density are also lie in the same range. Therefore, one can conclude that the algorithm for calculation the misorientation described above is correct and can be used future work.

Appendix C

Example of MATLAB code for 2D simulation of a single grain shrinkage due to evolving dislocation density jump

C.1 Main Program

C.1.1 singlegrain

```
%  This code simulates a single grain growth due to evolving dislocation den
%  Daliya Aflyatunova

% load(’cur_diff_cell_sizes.mat’) %file contains values of the  parameter A
fixed=1; %  "1" for  fixed, "0" for updated time incremenet
Kink=12;
```

```
step_div=1; % time increment delta_t/step_div
for i_iter=10:10
    tic
    %% grain and cell sizes
    r_cells=15+i_iter*5 %grain radius
    d_cell=0.000125/r_cells; % cell size
    r0_true=125e-6; % true radius
    r=round(r0_true/d_cell); % initial radius
    %% Domain parameters
    d=2;  % dimension: "1" -1D, "2"-2D, "3"-3D
    x0=5.5+r_cells; y0=x0; z0=1; % grain centre
    %% area size
    Nx=x0*2-1;
    if d==2
        Ny=Nx;
        Nz=1;
    elseif d==1
        Ny=1;
        Nz=1;
    elseif d==3
        Ny=Nx;
        Nz=Nx;
        z0=x0;
    end;
% Physical constants
PhysicalConstants
% make a single grain
```

```
indvector=[1:Nx*Ny*Nz]’;
[grnumstring, phi1, phi,phi2, status]= arrayfun(@(inum) ...
    Makecircle(inum,Nx, Ny, Nz,x0, y0, z0, r,d), indvector);
grnumstringstep=grnumstring;
grmax=max(grnumstring);
gr_properties=zeros(6,grmax);
for i=1:grmax
    gr_properties(1,i)=i;
end;
for i=1:grmax
    gr_properties(2,i)=nnz(grnumstring==gr_properties(1,i))
end;
gr_properties(3,1)=0; %recrystallized
gr_properties(3,2)=-Delta_t_c; % deformed
for i=1:grmax
    gr_properties(4,i)=phi1(find(grnumstring==i,1));
    gr_properties(5,i)=phi(find(grnumstring==i,1));
    gr_properties(6,i)=phi2(find(grnumstring==i,1));
end;
clear  phi1 phi phi2;
%% Find a border
boundarycells=zeros(1,Nx*Ny*Nz);
for inum=1:Nx*Ny*Nz
    BorderChange % finds grain boundary cells
end;
%% Calculate grain size
gr_num=grmax;
```

```
n=1;
CalculateGrainSize %calculates grain size
%cur_parameter=analizeddata_fixed_12(i_iter, 2);
%% Run simulation
Run
%SaveVTK % saves structure for visualization in ParaView
end;
```

C.1.2 PhysicalConstants

```
%% Physical constants
rho_c=5.510*10^(13); %[m^-2] %
delta_D_ob=7.50*10^(-15); %[m^3 s^-1]
burg_vec=2.560*10^(-10); %[m]
Q_b=110.0*10^3; %[J/mol]
R_gas=8.310; %[J mol^-1 K^-1]
k_boltz=1.380*10^(-23); %[J K^-1]
Temp=750.0; %[K]
M_0=delta_D_ob*burg_vec*exp(-Q_b/(R_gas*Temp))/(k_boltz*Temp);
gamma_m=0.625; %[Jm^2]
k1=2.970*10^8;
A1_stress=2.0*10^44;
eps_rate=0.0020; %[s^-1]
Q_a=275.0*10^3; %[J/mol]
A2_stress=7.60;
sigma_st=(A1_stress*eps_rate*exp(Q_a/R_gas/Temp))^(1/A2_stress);
```

```
mu_0=4.21*10^10; %[Pa] //G in Kugler's notation
Tm_mu0dmu_dT=-0.54;
Temp_m=1356; %[K]
alpha=0.50;
mu=mu_0*(1+Tm_mu0dmu_dT*(Temp-300)/Temp_m);
k2_dens=alpha*mu*burg_vec*k1/sigma_st;
tau=0.50*mu*burg_vec^2; %dislocation line energy
M=M_0*(1-exp(-5*(45/15)^4));
rho_ini=0;
cur_parameter=1;
Delta_t_c=-2/(k2_dens*eps_rate)*log((k1-sqrt(rho_c)*k2_dens)/(k1-sqrt(rho_in
```

C.1.3 Makecircle

```
%% Create circle
 function [grnumstring_el, phi1_el, phi_el,phi2_el, status_el]=Makecircle(i,
 z=fix((i-1)/Nx/Ny)+1; y=fix((i-1-Nx*Ny*(z-1))/Nx)+1; x= mod((i-1-Nx*Ny*z),
 if d==2
     if ((x-x0)^2+(y-y0)^2<=r^2)&(z==1)
         grnumstring_el=2;
         phi1_el=degtorad(45);
         phi_el=0;
         phi2_el=0;
         %status_el=1;
         status_el=0;
     else
```

```
        grnumstring_el=1;
        phi1_el=0;
        phi_el=0;
        phi2_el=0;
        %status_el=0;
        status_el=1;
    end;
elseif d==1
    if ((x-x0)^2<=r^2)&(z==1)&(y==1)
        grnumstring_el=2;
        phi1_el=degtorad(45);
        phi_el=0;
        phi2_el=0;
        %status_el=1;
        status_el=0;
    else
        grnumstring_el=1;
        phi1_el=0;
        phi_el=0;
        phi2_el=0;
        %status_el=0;
        status_el=1;
    end;
elseif d==3
    if ((x-x0)^2+(y-y0)^2+(z-z0)^2<=r^2)
        grnumstring_el=2;
        phi1_el=degtorad(45);
```

```
         phi_el=0;
         phi2_el=0;
         %status_el=1;
         status_el=0;
     else
         grnumstring_el=1;
         phi1_el=0;
         phi_el=0;
         phi2_el=0;
         %status_el=0;
         status_el=1;
     end;
 end
```

C.1.4 BorderChange

```
%% Boundary Change
z=fix((inum-1)/Nx/Ny)+1; y=fix((inum-1-Nx*Ny*(z-1))/Nx)+1; x= mod((inum-1-Nx
[dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z, Nx, Ny, Nz);
istring2=zeros(1,25); istring3=zeros(1,25);
switch d
    case 1
        istring3(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(23)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        for i=21:23
```

```
            condition2=false;
            if boundarycells(istring3(i))==0
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction
                istring2(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(23)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=21:23
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        boundarycells(istring3(i))=1;
                        break
                    end;
                end;
            else
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction
                istring2(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(23)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=21:23
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        condition2=true;
                        break
                    end;
                end;
                if condition2==false
                    boundarycells(istring3(i))=0;
                end;
            end;
        end;
    case 2
        istring3(23)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(24)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(25)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        for i=21:25
            condition2=false;
            if boundarycells(istring3(i))==0
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-Nx*Ny*(
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z,
                istring2(23)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(24)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(25)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=21:25
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        boundarycells(istring3(i))=1;
                        break
                    end;
                end;
            else
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction
                istring2(23)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(24)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(25)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=21:25
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        condition2=true;
                        break
                    end;
                end;
                if condition2==false
                    boundarycells(istring3(i))=0;
                end;
            end;
        end;
    case 3
        istring3(1)=(x-1)+(y-1)*Nx+(dirz2-1)*Nx*Ny+1;
        istring3(2)=(x-1)+(y-1)*Nx+(dirz1-1)*Nx*Ny+1;
        istring3(3)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(4)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(5)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(6)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
        istring3(7)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        for i=1:7
            condition2=false;
            if boundarycells(istring3(i))==0
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-Nx*Ny*(
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z,
                istring2(1)=(x-1)+(y-1)*Nx+(dirz2-1)*Nx*Ny+1;
                istring2(2)=(x-1)+(y-1)*Nx+(dirz1-1)*Nx*Ny+1;
                istring2(3)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(4)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(5)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(6)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(7)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=1:7
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        boundarycells(istring3(i))=1;
                        break
                    end;
                end;
            else
                z=fix((istring3(i)-1)/Nx/Ny)+1; y=fix((istring3(i)-1-
                [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction
                istring2(1)=(x-1)+(y-1)*Nx+(dirz2-1)*Nx*Ny+1;
                istring2(2)=(x-1)+(y-1)*Nx+(dirz1-1)*Nx*Ny+1;
                istring2(3)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(4)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(5)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(6)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
                istring2(7)=(x-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
                count=0;
                for ii=1:7
                    if grnumstring(istring2(ii))~=grnumstring(istring
                        count=count+1;
                    end;
                    if count>0
                        condition2=true;
                        break
                    end;
                end;
                if condition2==false
                    boundarycells(istring3(i))=0;
                end;
            end;
        end;
end;
```

C.1.5 Direction

```
%% Direction
function [dirx1, dirx2, diry1, diry2, dirz1, dirz2] = Direction (x,y,z, Nx,
dirx1=x-1; dirx2=x+1;
diry1=y-1; diry2=y+1;
dirz1=z-1; dirz2=z+1;
if dirx1<1
    dirx1=Nx;
end;
if dirx2>(Nx)
    dirx2=1;
end;
if diry1<1
    diry1=Ny;
end;
if diry2>(Ny)
```

```
    diry2=1;
end;
if dirz1<1
    dirz1=Nz;
end;
if dirz2>(Nz)
    dirz2=1;
end;
end
```

C.1.6 CalculateGrainSize

```
switch d
    case 1
        minsize=0;
    case 2
        minsize=0;
    case 3
        minsize=4;
end;
gr_properties_copy=[];
num=0;
for i=(n+1):2 %gr_num
    if gr_properties(2,i)>minsize
        gr_properties_copy=[gr_properties_copy gr_properties(2,i)];
        num=num+1;
    end;
end;
diam=zeros(1,num);
if num>0
    for i=1:num
        if d==3
            diam(i)=(6*gr_properties_copy(i)/3.14159265359)^(1/3);
        end;
        if d==2
            diam(i)=(4*gr_properties_copy(i)/3.14159265359)^(1/2);
        end;
        if d==1
            diam(i)=gr_properties_copy(i);
        end;
    end;
    jj=0;
    for i=1:num
        jj=jj+diam(i);
    end;
    jj=jj/(num);
    aver3dnew=jj; %average diameter
    diam_av=zeros(1,num);
    for i=1:num
        diam_av(i)=log10(diam(i)/jj); %tnositel'no srednego
    end;
    diam=[];
    diam_av=[];
```

```
elseif num==0
    aver3dnew=0;  %average diameter
end;
```

C.2 Run Simulation

C.2.1 Run

```
%% Run Program
dis_weight=1;
cur_weight=0;
name=['C:/Users/uos/Documents/MATLAB/example_for_latex/res/simulated_
fraction25=zeros(Nx*Ny*Nz, 25);
fraction25_1=zeros(Nx*Ny*Nz, 1);
fraction25_2=zeros(Nx*Ny*Nz, 1);
fraction25_3=zeros(Nx*Ny*Nz, 1);
fraction25_4=zeros(Nx*Ny*Nz, 1);
fraction25_5=zeros(Nx*Ny*Nz, 1);
fraction25_6=zeros(Nx*Ny*Nz, 1);
v_neighb=zeros(1,25);
v_neighb_dis=zeros(1,25);
v_neighb_cur=zeros(1,25);
calc_neighb=zeros(1,25);
grnumini=length(gr_properties(1, :));
CAS=0;
time_inc=0;
```

```
time=time_inc*CAS;
eps=eps_rate*CAS*time_inc;
n=1;
CalculateGrainSize
switch d
    case 1
        nummdiv=25;
        nummdiv2=Kink;
    case 2
        nummdiv=25;
        nummdiv2=Kink;
    case 3
        nummdiv=125;
        nummdiv2=Kink;
end;
f=fopen( name, 'w' );
if d==1
    numm=21;
    numm2=22;
elseif d==2
    numm=21;
    numm2=24;
elseif d==3
    numm=1;
    numm2=6;
end;
while (time<800)
```

```
    v_max_real=0;
    v_max_real_dis=0;
    v_max_real_cur=0;
    N_i_min=24;
    frmax=0;
    %% V MAX non-fixed
    if fixed==0
        i=(find(boundarycells==1))’;
        [v_max_real(i)]=arrayfun(@( i) ...
            v_max_fun( i, grnumstring, Nx, Ny, Nz, d, N_i_min, v_max_
            cur_weight, cur_parameter, nummdiv2, nummdiv, d_cell, gam
            gr_properties, time, k2_dens, eps_rate, k1, rho_ini), i,
        v_max_real=max(v_max_real);
    elseif fixed==1
        %% %%% OR fixed
        %v_max_real=cur_weight*M_0*gamma_m*cur_parameter*(nummdiv2-0)
        t_dis_max=(0-2/(k2_dens*eps_rate)*log(-k1/(exp(k2_dens*eps_ra
            sqrt(rho_ini)*k2_dens-k1)));
        rho_d=1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(t_dis_max-(-Delt
            *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
        rho_r=1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(t_dis_max-0))/2)
            *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
        rho_dis_jump_max=rho_d-rho_r;
        v_max_real=cur_weight*M_0*gamma_m*cur_parameter*(nummdiv2-0)/
            dis_weight*M_0*tau*rho_dis_jump_max; %evolving  dis jump
    end;
    if v_max_real==0
```

```
        delta_t_dis=0;
    else
        delta_t_dis=d_cell/v_max_real/step_div;
    end;
    delta_t_cur=delta_t_dis;
    delta_t_both=delta_t_dis;
    delta_t=delta_t_both;
    time_inc=delta_t;
    if sum(gr_properties(2,:))~=Nx*Ny*Nz
        error('Error Message')
    end;
    fprintf(f,'%6.5f %12.12f\r\n', [time;aver3dnew/2*d_cell]);
    DynamicGrowth;
    CalculateGrainSize;
    if sum(gr_properties(2,:))~=Nx*Ny*Nz
        error('Error Message')
    end;
    CAS=CAS+1;
    time=time+time_inc;
    eps=eps+eps_rate*time_inc;
end;
toc
fclose(f);
```

C.2.2 v_max_fun

```
%% maximum velocity
function [v_max_real]=v_max_fun( i_recr, grnumstring, Nx, Ny,Nz, d,
    v_max_real, dis_weight, M_0, tau, cur_weight, cur_parameter, numm
    d_cell, gamma_m, v_max_real_dis, v_max_real_cur, gr_properties, t
grain1or2=grnumstring(i_recr);
z=fix((i_recr-1)/Nx/Ny)+1; y=fix((i_recr-1-Nx*Ny*(z-1))/Nx)+1; x= mo
[dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z, Nx, Ny, N
switch d
    case 1
        % calc_neighb(1)=dirx1+diry1*Nx+dirz2*Nx*Ny;
        % calc_neighb(2)=dirx2+diry1*Nx+dirz2*Nx*Ny;
        %       calc_neighb(3)=dirx2+diry1*Nx+dirz1*Nx*Ny;
        %       calc_neighb(4)=dirx1+diry1*Nx+dirz1*Nx*Ny;
        %       calc_neighb(5)=dirx1+diry2*Nx+dirz2*Nx*Ny;
        %       calc_neighb(6)=dirx2+diry2*Nx+dirz2*Nx*Ny;
        %       calc_neighb(7)=dirx2+diry2*Nx+dirz1*Nx*Ny;
        %       calc_neighb(8)=dirx1+diry2*Nx+dirz1*Nx*Ny;

        %       calc_neighb(9)=x+diry1*Nx+dirz2*Nx*Ny;
        %       calc_neighb(10)=x+diry1*Nx+dirz1*Nx*Ny;
        %       calc_neighb(11)=dirx1+y*Nx+dirz2*Nx*Ny;
        %       calc_neighb(12)=dirx2+y*Nx+dirz2*Nx*Ny;
        %       calc_neighb(13)=dirx2+y*Nx+dirz1*Nx*Ny;
        %       calc_neighb(14)=dirx1+y*Nx+dirz1*Nx*Ny;
        %       calc_neighb(15)=x+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(16)=x+diry2*Nx+dirz1*Nx*Ny;
        %               calc_neighb(25)=x+y*Nx+dirz2*Nx*Ny;

        %               calc_neighb(26)=x+y*Nx+dirz1*Nx*Ny;
        %
        %               calc_neighb(19)=dirx2+diry1*Nx+z*Nx*Ny;
        %               calc_neighb(20)=dirx1+diry1*Nx+z*Nx*Ny;
        %
        %               calc_neighb(17)=dirx2+diry2*Nx+z*Nx*Ny;
        %               calc_neighb(18)=dirx1+diry2*Nx+z*Nx*Ny;

        %               calc_neighb(23)=x+diry1*Nx+z*Nx*Ny;
        calc_neighb(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        %               calc_neighb(24)=x+diry2*Nx+z*Nx*Ny;
        numm=21; numm2=22;
    case 2
        %               calc_neighb(1)=dirx1+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(2)=dirx2+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(3)=dirx2+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(4)=dirx1+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(5)=dirx1+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(6)=dirx2+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(7)=dirx2+diry2*Nx+dirz1*Nx*Ny;
        %               calc_neighb(8)=dirx1+diry2*Nx+dirz1*Nx*Ny;
        %
        %               calc_neighb(9)=x+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(10)=x+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(11)=dirx1+y*Nx+dirz2*Nx*Ny;
        %               calc_neighb(12)=dirx2+y*Nx+dirz2*Nx*Ny;
        %               calc_neighb(13)=dirx2+y*Nx+dirz1*Nx*Ny;
        %               calc_neighb(14)=dirx1+y*Nx+dirz1*Nx*Ny;
        %               calc_neighb(15)=x+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(16)=x+diry2*Nx+dirz1*Nx*Ny;
        %
        %               calc_neighb(25)=x+y*Nx+dirz2*Nx*Ny;
        %               calc_neighb(26)=x+y*Nx+dirz1*Nx*Ny;
        %
        %               calc_neighb(19)=dirx2+diry1*Nx+z*Nx*Ny;
        %               calc_neighb(20)=dirx1+diry1*Nx+z*Nx*Ny;
        %
        %               calc_neighb(17)=dirx2+diry2*Nx+z*Nx*Ny;
        %               calc_neighb(18)=dirx1+diry2*Nx+z*Nx*Ny;

        calc_neighb(23)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(24)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
        numm=21; numm2=24;
        %  nummdiv=25; nummdiv2=15;//15;
    case 3
        %               calc_neighb(7)=dirx1+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(8)=dirx2+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(9)=dirx2+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(10)=dirx1+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(11)=dirx1+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(12)=dirx2+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(13)=dirx2+diry2*Nx+dirz1*Nx*Ny;
        %               calc_neighb(14)=dirx1+diry2*Nx+dirz1*Nx*Ny;
        %
        %               calc_neighb(15)=x+diry1*Nx+dirz2*Nx*Ny;
        %               calc_neighb(16)=x+diry1*Nx+dirz1*Nx*Ny;
        %               calc_neighb(17)=dirx1+y*Nx+dirz2*Nx*Ny;
        %               calc_neighb(18)=dirx2+y*Nx+dirz2*Nx*Ny;
        %               calc_neighb(19)=dirx2+y*Nx+dirz1*Nx*Ny;
        %               calc_neighb(20)=dirx1+y*Nx+dirz1*Nx*Ny;
        %               calc_neighb(21)=x+diry2*Nx+dirz2*Nx*Ny;
        %               calc_neighb(22)=x+diry2*Nx+dirz1*Nx*Ny;

        calc_neighb(1)=(x-1)+(y-1)*Nx+(dirz2-1)*Nx*Ny+1;
        calc_neighb(2)=(x-1)+(y-1)*Nx+(dirz1-1)*Nx*Ny+1;
        %               calc_neighb(23)=dirx2+diry1*Nx+z*Nx*Ny;
        %               calc_neighb(24)=dirx1+diry1*Nx+z*Nx*Ny;
        %               calc_neighb(25)=dirx2+diry2*Nx+z*Nx*Ny;
        %               calc_neighb(26)=dirx1+diry2*Nx+z*Nx*Ny;
        calc_neighb(3)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(4)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(5)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
        calc_neighb(6)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
        numm=1;  numm2=6;
        % nummdiv=125; nummdiv2= 62; //nummdiv2=75;
```

```
end;
%% unique
unique_num=1;
switch d
    case 1
        unique_NBH_Moore(1)=grnumstring(calc_neighb(21));
    case 2
        unique_NBH_Moore(1)=grnumstring(calc_neighb(21));
    case 3
        unique_NBH_Moore(1)=grnumstring(calc_neighb(1));
end;
for kk=numm:numm2
    uniq_boolean=true;
    for uniq=1:unique_num
        if grnumstring(calc_neighb(kk))==unique_NBH_Moore(uniq)
            uniq_boolean=false;
        end;
    end;
    if uniq_boolean==true
        unique_num=unique_num+1;
        unique_NBH_Moore(unique_num)=grnumstring(calc_neighb(kk));
    end;
end;
for kk=numm:numm2
    for uniq=1:unique_num
         gr_neighb=grnumstring(calc_neighb(kk));

        if gr_neighb==unique_NBH_Moore(uniq) %and (fraction25(ii,k)>1)
            if  gr_neighb==grain1or2
                v_max_real=v_max_real;
            else
                if curvature_f(calc_neighb(kk), Nx, Ny, Nz, grnumstring)< N_
                    N_i_min=curvature_f(calc_neighb(kk), Nx, Ny, Nz, grnumst
                end;
                rho_gr_neighb=1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(time-gr
                    *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
                rho_grain1or2 =1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(time-g
                    *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
                v_neighb_dis(kk)=(-1)*dis_weight*(M_0*tau*(rho_gr_neighb-rho
                if v_max_real<(-1)*dis_weight*(M_0*tau*(rho_gr_neighb-rho_gr
                    v_max_real=(-1)*dis_weight*(M_0*tau*(rho_gr_neighb-rho_g
%                 if v_max_real<dis_weight*(M_0*tau*(gr_properties(3,gr_neig
%                     v_max_real=dis_weight*(M_0*tau*(gr_properties(3,gr_nei
                else
                    v_max_real=v_max_real;
                end;
            end;
        end;
    end;
end;
end
```

C.2.3 DynamicGrowth

```
%% Dynamic Growth
gr_properties_copy=gr_properties;
%% Dislocation density
grnumstringstep=grnumstring;
statusstep=status;
i=find(boundarycells==1)';
changedcells=zeros(1, Nx*Ny*Nz);
[fraction25_1(i), fraction25_2(i), fraction25_3(i), fraction25_4(i),
DG( i, fraction25_1,fraction25_2, fraction25_3, fraction25_4,  fracti
 dis_weight, M_0, tau, cur_weight, cur_parameter, nummdiv2, nummdiv,
 gamma_m, delta_t_dis, delta_t_cur, numm, numm2, gr_properties, CAS,
if sum(changedcells)>0
    changedind=find(changedcells==1);
    for inum=changedind
        gr1=grnumstring(inum);%old
        gr2=grnumstringstep(inum); %new
        status(inum)=statusstep(inum);
        grnumstring(inum)=gr2;
        BorderChange;
        gr_properties(2,gr1)=gr_properties(2,gr1)-1;
        gr_properties(2,gr2)=gr_properties(2,gr2)+1;
    end;
end;
```

C.2.4 DG

```
function [fraction25_1, fraction25_2, fraction25_3, fraction25_4, fraction25_
 tau, cur_weight, cur_parameter, nummdiv2, nummdiv, d_cell, gamma_m, delta_t
 eps_rate, time,k1, rho_ini,d)
grain1or2=grnumstring(i);
z=fix((i-1)/Nx/Ny)+1; y=fix((i-1-Nx*Ny*(z-1))/Nx)+1; x= mod((i-1-Nx*Ny*z), N
[dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z, Nx, Ny, Nz);
if d==2
    calc_neighb(23)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(24)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
elseif d==1
    calc_neighb(22)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(21)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
elseif d==3
    calc_neighb(1)=(x-1)+(y-1)*Nx+(dirz2-1)*Nx*Ny+1;
    calc_neighb(2)=(x-1)+(y-1)*Nx+(dirz1-1)*Nx*Ny+1;
    calc_neighb(3)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(4)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(5)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    calc_neighb(6)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
end;
%% Fraction
for kk=numm:numm2
    gr_neighb=grnumstring(calc_neighb(kk));
```

```
     if gr_neighb==grain1or2
        v_neighb_dis(kk)=0;
        v_neighb_cur(kk)=0;
     else
         rho_gr_neighb=1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(time-gr
             *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
         rho_grain1or2 =1/(k2_dens^2)*(exp(-(k2_dens*eps_rate*(time-g
             *(k1-k2_dens*sqrt(rho_ini))-k1)^2;
         v_neighb_dis(kk)=(-1)*dis_weight*(M_0*tau*(rho_gr_neighb-rho
         if cur_weight==0
             v_neighb_cur(kk)=0;
         else
             v_neighb_cur(kk)=-cur_weight*cur_parameter*(nummdiv2-cur
         end;
     end;
end;
v_neighb_cur(v_neighb_cur<0)=0; %%%%UNCOMEMT
v_neighb_dis(v_neighb_dis<0)=0;
if d==1
    fraction25_1=fraction25_1(i)+(v_neighb_dis(21)*delta_t_dis+v_neig
    fraction25_2=fraction25_2(i)+(v_neighb_dis(22)*delta_t_dis+v_neig
    fraction25_3=0;
    fraction25_4=0;
    fraction25_5=0;
    fraction25_6=0;
    fr25string(numm:numm2)=[fraction25_1 fraction25_2];
elseif d==2
```

```
    fraction25_1=fraction25_1(i)+(v_neighb_dis(21)*delta_t_dis+v_neighb_cur(
    fraction25_2=fraction25_2(i)+(v_neighb_dis(22)*delta_t_dis+v_neighb_cur(
    fraction25_3=fraction25_3(i)+(v_neighb_dis(23)*delta_t_dis+v_neighb_cur(
    fraction25_4=fraction25_4(i)+(v_neighb_dis(24)*delta_t_dis+v_neighb_cur(
    fraction25_5=0;
    fraction25_6=0;
    fr25string(numm:numm2)=[fraction25_1 fraction25_2 fraction25_3 fraction2
elseif d==3
    fraction25_1=fraction25_1(i)+(v_neighb_dis(1)*delta_t_dis+v_neighb_cur(1
    fraction25_2=fraction25_2(i)+(v_neighb_dis(2)*delta_t_dis+v_neighb_cur(2
    fraction25_3=fraction25_3(i)+(v_neighb_dis(3)*delta_t_dis+v_neighb_cur(3
    fraction25_4=fraction25_4(i)+(v_neighb_dis(4)*delta_t_dis+v_neighb_cur(4
    fraction25_5=fraction25_5(i)+(v_neighb_dis(5)*delta_t_dis+v_neighb_cur(5
    fraction25_6=fraction25_6(i)+(v_neighb_dis(6)*delta_t_dis+v_neighb_cur(6
    fr25string(numm:numm2)=[fraction25_1 fraction25_2 fraction25_3 fraction2
end;
%% Creating 4 rows array unique_NBH_Moore : 1st row unique elements, 2nd  di
unique_num=1;
unique_NBH_Moore=[];
unique_NBH_Moore(1,1)=grnumstring(calc_neighb(numm));
for ii=numm:numm2
  uniq_boolean=true;
  for uniq=1:unique_num
      if grnumstring(calc_neighb(ii))==unique_NBH_Moore(uniq)
         uniq_boolean=false;
      end;
  end;
  if uniq_boolean==true
     unique_num=unique_num+1;
     unique_NBH_Moore(unique_num)=grnumstring(calc_neighb(ii));  %uni
  end;
end;
unique_NBH_Moore=[unique_NBH_Moore; zeros(1,  unique_num)];
for ii=numm:numm2
if fr25string(ii)>0
     for uniq=1:unique_num
        if (grnumstring(calc_neighb(ii))==unique_NBH_Moore(1,uniq))
                unique_NBH_Moore(2,uniq)=unique_NBH_Moore(2,uniq)+fr2
        end;
     end;
end;
end;
%% Find max and max indices in 2nd row (max distance)
uniq_max=unique_NBH_Moore(2,1);
for uniq=1:unique_num
   if unique_NBH_Moore(2,uniq)>uniq_max
       uniq_max=unique_NBH_Moore(2,uniq);
   end;
end;
c1=[];
for uniq=1:unique_num
    if unique_NBH_Moore(2,uniq)==uniq_max
       c1=[c1 uniq];
    end;
```

```
end;
%% Make decision to change cell state
if ( fraction25_1>=1|fraction25_2>=1|fraction25_3>=1|fraction25_4>=1|fractio
    if length(c1)>1
        index_c1=c1(randi(length(c1)));
        grnumstringstep(i)=unique_NBH_Moore(1,index_c1);
        gr1=grnumstring(i); %prev gr
        gr2=unique_NBH_Moore(1,index_c1);  %new grain
        grnumstringstep=gr2;
        changedcells=1;
        statusstep=1;
        fraction25_1=0;
        fraction25_2=0;
        fraction25_3=0;
        fraction25_4=0;
        fraction25_5=0;
        fraction25_6=0;
    elseif length(c1)==1
           gr1=grnumstring(i);  %prev gr
           gr2=unique_NBH_Moore(1,c1);  %new grain
           grnumstringstep=gr2;
           changedcells=1;
           statusstep=1;
           fraction25_1=0;
           fraction25_2=0;
           fraction25_3=0;
           fraction25_4=0;
           fraction25_5=0;
           fraction25_6=0;
    end;
else
    grnumstringstep=grain1or2;
    changedcells=0;
    statusstep=status(i);
end;
end
```

C.2.5 curvature_f

```
function  [result]=curvature_f (inum,Nx, Ny, Nz, grnumstring)
if isempty(inum)==1
    result=0;
end;
for i=1:length(inum)
    z=fix((inum(i)-1)/Nx/Ny)+1; y=fix((inum(i)-1-Nx*Ny*(z-1))/Nx)+1;
    [dirx1, dirx2, diry1, diry2, dirz1, dirz2]=Direction(x,y,z, Nx, N
    [dirx_1, dirx_2, diry_1, diry_2, dirz_1, dirz_2]=Direction2(x,y,z

    istring3(1)=(dirx_1-1)+(diry_1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(2)=(dirx1-1)+(diry_1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(3)=(x-1)+(diry_1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(4)=(dirx2-1)+(diry_1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(5)=(dirx_2-1)+(diry_1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(6)=(dirx_1-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(7)=(dirx1-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(8)=(x-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(9)=(dirx2-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(10)=(dirx_2-1)+(diry1-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(11)=(dirx_1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(12)=(dirx1-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(13)=(dirx2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(14)=(dirx_2-1)+(y-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(15)=(dirx_1-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(16)=(dirx1-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(17)=(x-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(18)=(dirx2-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(19)=(dirx_2-1)+(diry2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(20)=(dirx_1-1)+(diry_2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(21)=(dirx1-1)+(diry_2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(22)=(x-1)+(diry_2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(23)=(dirx2-1)+(diry_2-1)*Nx+(z-1)*Nx*Ny+1;
    istring3(24)=(dirx_2-1)+(diry_2-1)*Nx+(z-1)*Nx*Ny+1;
    numm=1; numm2=24;
    result(i)=sum(grnumstring(istring3(numm:numm2))==grnumstring(inum(i)));
end;
end
```

C.2.6 Direction2

```
function [dirx_1, dirx_2, diry_1, diry_2, dirz_1, dirz_2]=Direction2(
dirx_1=x-2; dirx_2=x+2;
diry_1=y-2; diry_2=y+2;
dirz_1=z-2; dirz_2=z+2;
if dirx_1<1
    dirx_1=Nx;
end;
if dirx_2>Nx
    dirx_2=1;
end;
if diry_1<1
    diry_1=Ny;
end;
if diry_2>Ny
    diry_2=1;
end;
if dirz_1<1
    dirz_1=Nz;
end;
if dirz_2>Nz
    dirz_2=1;
end;
end
```

Bibliography

[1] K.G.F. Janssens. An introductory review of cellular automata modeling of moving grain boundaries in polycrystalline materials. *Mathematics and Computers in Simulation*, 80(7):1361 – 1381, March 2010. Multiscale modeling of moving interfaces in materials.

[2] S. Wolfram. *A New Kind of Science*. Wolfram Research, 2002.

[3] J. L. Schiff. *Cellular Automata: A Discrete View of the World*. Wiley Interscience, 2008.

[4] M.Hatherly F.J. Humphreys. *Recrystallization and related annealing phenomena*. Elsevier, 2004.

[5] R. E. Reed-Hill. *Physical metallurgy principles*. D. Van Nostrand Company, 1973.

[6] D. McLean. *Grain boundaries in metals*. Clarendon Press Oxford, 1957.

[7] H.Hu P.A. Beck. *J. Metals*, 185:627, 1949. quited in [6] p. 252.

[8] F. Haessner. *Recrystallization of metalic materials*. Dr. Reiderer Verlag GmbH, 1978.

[9] P. R. Rios, F. Siciliano Jr, Hugo R. Z. Sandim, R. L. Plaut, and A. F. Padilha. Nucleation and growth during recrystallization. *Materials Research*, 8:225 – 238, 09 2005.

[10] M. Muramatsu, Y. Aoyagi, Y. Tadano, and K. Shizawa. Phase-field simulation of static recrystallization considering nucleation from subgrains and nucleus growth with incubation period. *Computational Materials Science*, 87(Supplement C):112 – 122, 2014.

[11] W. Roberts and B. Ahlblom. A nucleation criterion for dynamic recrystallization during hot working. *Acta Metallurgica*, 26(5):801 – 813, 1978.

[12] J. W. Cahn. The kinetics of grain boundary nucleated reactions. *Acta Metallurgica*, 4(5):449 – 459, 1956.

[13] H. Mecking and U.F. Kocks. Kinetics of flow and strain-hardening. *Acta Metallurgica*, 29(11):1865 – 1875, 1981.

[14] R. Ding and Z.X Guo. Coupled quantitative simulation of microstructural evolution and plastic flow during dynamic recrystallization. *Acta Materialia*, 49(16):3163 – 3175, September 2001.

[15] S. Takeuchi and A. S. Argon. Steady-state creep of single-phase crystalline matter at high temperature. *Journal of Materials Science*, 11(8):1542–1566, Aug 1976.

[16] T. Takaki, A. Yamanaka, and Yoshihiro Tomita. Multi-phase-field simulations of dynamic recrystallization during transient deformation. *ISIJ International*, 51(10):1717–1723, 2011.

[17] T. Takaki, T. Hirouchi, Y. Hisakuni, A. Yamanaka, and Y. Tomita. Multi-phase-field model to simulate microstructure evolutions during dynamic recrystallization. *MATERIALS TRANSACTIONS*, 49(11):2559–2565, 2008.

[18] I. Steinbach and F. Pezzolla. A generalized field method for multiphase transformations using interface fields. *Physica D: Nonlinear Phenomena*, 134(4):385 – 393, 1999.

[19] S. G. Kim, D. I. Kim, W. T. Kim, and Y. B. Park. Computer simulations of two-dimensional and three-dimensional ideal grain growth. *Phys. Rev. E*, 74:061605, Dec 2006.

[20] H. Hallberg. Approaches to modeling of recrystallization. *Metals, special issue on ´´Processing and Properties of Bulk Nanostructured Materials´´*, 1(1):16–48, 2011.

[21] J.E. Burke and D. Turnbull. Recrystallization and grain growth. *Progress in Metal Physics*, 3:220 – 292, 1952.

[22] H.V. Atkinson. Overview no. 65: Theories of normal grain growth in pure single phase systems. *Acta Metallurgica*, 36(3):469 – 491, March 1988.

[23] O. Hunderi and N. Ryum. The kinetics of normal grain growth. *Journal of Materials Science*, 15(5):1104–1108, May 1980.

[24] M Hillert. On the theory of normal and abnormal grain growth. *Acta Metallurgica*, 13(3):227 – 238, March 1965.

[25] P. Feltham. Grain growth in metals. *Acta Metallurgica*, 5(2):97 – 105, February 1957.

[26] N.P Louat. On the theory of normal grain growth. *Acta Metallurgica*, 22(6):721 – 724, June 1974.

[27] M.P Anderson, D.J Srolovitz, G.S Grest, and P.S Sahni. Computer simulation of grain growth - i. kinetics. *Acta Metallurgica*, 32(5):783 – 791, May 1984.

[28] N. Gao and T.N. Baker. Austenite grain growth behavior of microalloyed al-v-n and al-v-ti-n steels. *ISIJ International*, 38(7):744–751, 1998. Iron and Steel Inistitute of Japan.

[29] G. Abbruzzese and K. Lücke. A theory of texture controlled grain growth – i. derivation and general discussion of the model. *Acta Metallurgica*, 34(5):905 – 914, May 1986.

[30] G. Gottstein. *Physical Foundations of Materials Science*. Springer, 2005.

[31] D. G. Cole, P. Feltham, and E. Gillam. On the Mechanism of Grain Growth in Metals, with Special Reference to Steel. *Proceedings of the Physical Society B*, 67:131–137, February 1954.

[32] K. Lücke and H.P. Stüwe. On the theory of grain boundary motion. In L. Himmel, editor, *Recovery and Recrystallization of Metals*, pages 171–210. New York: Interscience, Oxford, 1963.

[33] C.H.J. Davies. Growth of nuclei in a cellular automaton simulation of recrystallisation. *Scripta materialia*, 36(1):35–40, 1997.

[34] Y.J. Lan, D.Z. Li, and Y.Y. Li. A mesoscale cellular automaton model for curvature-driven grain growth. *Metallurgical and Materials Transactions B*, 37(1):119–129, 2006.

[35] C. Zheng, N. Xiao, D. Li, and Y. Li. Mesoscopic modeling of austenite static recrystallization in a low carbon steel using a coupled simulation method. *Computational Materials Science*, 45(2):568 – 575, 2009.

[36] C. Zheng, N. Xiao, L. Hao, D. Li, and Y. Li. Numerical simulation of dynamic strain-induced austenite–ferrite transformation in a low carbon steel. *Acta Materialia*, 57(10):2956 – 2968, 2009.

[37] F. Han, B. Tang, H. Kou, J. Li, and Y. Feng. Cellular automata simulations of grain growth in the presence of second-phase particles. *Modelling and Simulation in Materials Science and Engineering*, 23(6):065010, 2015.

[38] R. Ding and Z.X. Guo. Microstructural modelling of dynamic recrystallisation using an extended cellular automaton approach. *Computational Materials Science*, 23(1):209 – 218, 2002.

[39] G. Kugler and R. Turk. Modeling the dynamic recrystallization under multi-stage hot deformation. *Acta Materialia*, 52(15):4659 – 4668, 2004.

[40] F.J. Humphreys. A unified theory of recovery, recrystallization and grain growth, based on the stability and growth of cellular microstructures—i. the basic model. *Acta Materialia*, 45(10):4231 – 4240, 1997.

[41] H. Hallberg, B. Svendsen, T. Kayser, and M. Ristinmaa. Microstructure evolution during dynamic discontinuous recrystallization in particle-containing cu. *Computational Materials Science*, 84(Supplement C):327 – 338, 2014.

[42] H. Hallberg, M. Wallin, and M. Ristinmaa. Simulation of discontinuous dynamic recrystallization in pure cu using a probabilistic cellular automaton. *Computational Materials Science*, 49(1):25 – 34, 2010.

[43] H. J. Frost and M. F. Ashby. *Deformation-Mechanism Maps: The Plasticity and Creep of Metals and Ceramics*. Pergamon Press, 1982.

[44] W.A. Johnson and R.F. Mehl. Trans. In *AIME,* volume 135, page 416, 1939.

[45] A.N. Kolmogorov et al. An kolmogorov, bull. acad. sci. ussr, phys. ser. 3, 355 (1937). *Bull. Acad. Sci. USSR, Phys. Ser.*, 3:355, 1937.

[46] M. Avrami. Kinetics of phase change. i general theory. *The Journal of chemical physics*, 7(12):1103–1112, 1939.

[47] W. T. Read and W. Shockley. Dislocation models of crystal grain boundaries. *Phys. Rev.*, 78:275–289, May 1950.

[48] M. L. Holzwoorth P. A. Beck and P. R. Sperry. Effect of a dispersed second phase on grain growth in al-mn alloys. *Transactions of the Metallurgical Society AIME*, 180:163–192, 1949.

[49] C. S. Smith. Some elementary principles of polycrystalline microstructure. *Metallurgical Reviews*, 9(1):1–48, January 1964.

[50] F. N. Rhines, K. R. Craig, and R. T. DeHoff. Mechanism of steady-state grain growth in aluminum. *Metallurgical Transactions*, 5(2):413–425, February 1974.

[51] J. B. Salem and S. Wolfram. Thermodynamics and hydrodynamics with cellular automata. *Thinking Machines Corp. Tech. Rep. Series*, 1985.

[52] S. Wolfram. Statistical mechanics of cellular automata. *Rev. Mod. Phys.*, 55(3):601–644, July 1983.

[53] S. Wolfram. Computation theory of cellular automata. *Communications in Mathematical Physics*, 96(1):15–57, 1984.

[54] D. Raabe. *Computational Materials Science. The simulation of Materials Microstructure and Properties*. Wiley-Vch, 1998.

[55] V. K. Vanag. Study of spatially extended dynamical systems using probabilistic cellular automata. *Physics-Uspekhi*, 42(5):413, 1999.

[56] O. M., A.M. Odlyzko, and S. Wolfram. Algebraic properties of cellular automata. *Communications in Mathematical Physics*, 93(2):219–258, 1984.

[57] S. Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35, 1984.

[58] S. Wolfram. Undecidability and intractability in theoretical physics. *Phys. Rev. Lett.*, 54(8):735–738, February 1985.

[59] N.H. Packard and S. Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38(5-6):901–946, March 1985.

[60] S. Wolfram. Origins of randomness in physical systems. *Phys. Rev. Lett.*, 55(5):449–452, July 1985.

[61] D.J. Srolovitz, M.P. Anderson, P.S. Sahni, and G.S. Grest. Computer simulation of grain growth—ii. grain size distribution, topology, and local dynamics. *Acta Metallurgica*, 32(5):793 – 802, May 1984.

[62] D.J. Srolovitz, G.S. Grest, and M.P. Anderson. Computer simulation of grain growth—v. abnormal grain growth. *Acta Metallurgica*, 33(12):2233 – 2247, December 1985.

[63] D.J. Srolovitz, G.S. Grest, and M.P. Anderson. Computer simulation of recrystallization—i. homogeneous nucleation and growth. *Acta Metallurgica*, 34(9):1833 – 1845, September 1986.

[64] S. G. R. Brown and J. A. Spittle. Computer simulation of grain growth and macrostructure development during solidification. *Materials Science and Technology*, 5(4):362–368, 1989.

[65] J.A. Spittle and S.G.R. Brown. A computer simulation of the influence of processing conditions on as-cast grain structures. *Journal of Materials Science*, 24(5):1777–1781, 1989.

[66] J.A Spittle and S.G.R Brown. Computer simulation of the effects of alloy variables on the grain structures of castings. *Acta Metallurgica*, 37(7):1803 – 1810, 1989.

[67] S.G.R. Brown and J.A. Spittle. Rule-based lattice computer models for simulating dendritic growth. *Scripta Metallurgica et Materialia*, 27(11):1599 – 1603, 1992.

[68] S.G.R. Brown, T. Williams, and J.A. Spittle. A cellular automaton model of the steady-state "free" growth of a non-isothermal dendrite. *Acta Metallurgica et Materialia*, 42(8):2893 – 2898, 1994.

[69] S. G. R. Brown, G. P. Clarke, and A. J. Brooks. Morphological variations produced by cellular automaton model of non-isothermal 'free' dendritic growth. *Materials Science and Technology*, 11(4):370–374, 1995.

[70] J.A. Spittle and S.G.R. Brown. A cellular automaton model of steady-state columnar-dendritic growth in binary alloys. *Journal of Materials Science*, 30(16):3989–3994, 1995.

[71] P. Zhu and R.W. Smith. Dynamic simulation of crystal growth by monte carlo method—i. model description and kinetics. *Acta Metallurgica et Materialia*, 40(4):683 – 692, 1992.

[72] M. Rappaz and Ch.-A. Gandin. Probabilistic modelling of microstructure formation in solidification processes. *Acta Metallurgica et Materialia*, 41(2):345 – 360, 1993.

[73] Ch.-A. Gandin, M. Rappaz, and R. Tintillier. Three-dimensional probabilistic simulation of solidification grain structures: Application to superalloy precision castings. *Metallurgical Transactions A*, 24(2):467–479, 1993.

[74] Ch.-A. Gandin and M. Rappaz. A coupled finite element-cellular automaton model for the prediction of dendritic grain structures in solidification processes. *Acta Metallurgica et Materialia*, 42(7):2233 – 2246, 1994.

[75] Ch.-A Gandin and M Rappaz. A 3d cellular automaton algorithm for the prediction of dendritic grain growth. *Acta Materialia*, 45(5):2187 – 2195, 1997.

[76] Ch.-A. Gandin, J.-L. Desbiolles, M. Rappaz, and Ph. Thevoz. A three-dimensional cellular automation-finite element model for the prediction of solidification grain structures. *Metallurgical and Materials Transactions A*, 30(12):3153–3165, 1999.

[77] Ch.-A. Gandin. Stochastic modeling of dendritic grain structures. *Advanced Engineering Materials*, 3(5):303–306, 2001.

[78] M.B. Cortie. Simulation of metal solidification using a cellular automaton. *Metallurgical Transactions B*, 24(6):1045–1053, 1993.

[79] H.W. Hesselbarth and I.R. Göbel. Simulation of recrystallization by cellular automata. *Acta Metallurgica et Materialia*, 39(9):2135 – 2143, 1991.

[80] C.H.J. Davies. The effect of neighbourhood on the kinetics of a cellular automaton recrystallisation model. *Scripta Metallurgica et Materialia*, 33(7):1139 – 1143, October 1995.

[81] C.F. Pezzee and D.C. Dunand. The impingement effect of an inert, immobile second phase on the recrystallization of a matrix. *Acta Metallurgica et Materialia*, 42(5):1509 – 1524, May 1994.

[82] R.K. Shelton and D.C. Dunand. Computer modeling of particle pushing and clustering during matrix crystallization. *Acta Materialia*, 44(11):4571 – 4585, 1996.

[83] C.H.J. Davies and L. Hong. The cellular automaton simulation of static recrystallization in cold-rolled {AA1050}. *Scripta Materialia*, 40(10):1145 – 1150, 1999.

[84] V. Marx, D. Raabe, O. Engler, and G. Gottstein. Simulation of the texture evolution during annealing of cold rolled bcc and fcc metals using a cellular automation approach. *Texture, Stress, and Microstructure*, 28(3-4):211–218, 1997.

[85] V. Marx, F.R. Reher, and G. Gottstein. Simulation of primary recrystallization using a modified three-dimensional cellular automaton. *Acta Materialia*, 47(4):1219 – 1230, March 1999.

[86] D. Raabe, F. Roters, and V. Marx. Experimental investigation and numerical simulation of the correlation of recovery and texture in bcc metals and alloys. *Textures and Microstructures*, 27(1):611, 1996.

[87] D. Raabe. Introduction of a scalable three-dimensional cellular automaton with a probabilistic switching rule for the discrete mesoscale simulation of recrystallization phenomena. *Philosophical Magazine A*, 79(10):2339–2358, 1999.

[88] R.L. Goetz and V. Seetharaman. Static recrystallization kinetics with homogeneous and heterogeneous nucleation using a cellular automata model. *Metallurgical and Materials Transactions A*, 29(9):2307–2321, September 1998.

[89] R.A. Vandermeer and R.A. Masumura. The microstructural path of grain-boundary-nucleated phase transformations. *Acta Metallurgica et Materialia*, 40(4):877 – 886, April 1992.

[90] R.L. Goetz and V. Seetharaman. Modeling dynamic recrystallization using cellular automata. *Scripta Materialia*, 38(3):405–413, 1998.

[91] D. Raabe. Cellular automata in materials science with particular reference to recrystallization simulation. *Annual review of materials research*, 32(1):53–76, 2002.

[92] D. Raabe and R. C. Becker. Coupling of a crystal plasticity finite-element model with a probabilistic cellular automaton for simulating primary static recrystallization in aluminium. *Modelling and Simulation in Materials Science and Engineering*, 8(4):445, 2000.

[93] D. Raabe and L. Hantcherli. 2d cellular automaton simulation of the recrystallization texture of an {IF} sheet steel under consideration of zener pinning. *Computational Materials Science*, 34(4):299 – 313, December 2005.

[94] D. Raabe. Yield surface simulation for partially recrystallized aluminum polycrystals on the basis of spatially discrete data. *Computational Materials Science*, 19(1-4):13 – 26, 2000.

[95] A.D. Rollett and D. Raabe. A hybrid model for mesoscopic simulation of recrystallization. *Computational Materials Science*, 21(1):69 – 78, May 2001.

[96] L. Zhang, C.B. Zhang, X.H. Liu, G.D. Wang, Y.M. Wang, et al. Modeling recrystallization of austenite for c-mn steels during hot deformation by cellular automaton. *J. Mater. Sci. Technol.*, 18(2), 2002.

[97] J.A. Spittle and S.G.R. Brown. A 3d cellular automaton model of coupled growth in two component systems. *Acta Metallurgica et Materialia*, 42(6):1811 – 1815, June 1994.

[98] Y. Liu, T. Baudin, and R. Penelle. Simulation of normal grain growth by cellular automata. *Scripta Materialia*, 34(11):1679 – 1683, June 1996.

[99] M. Kumar, R. Sasikumar, and P. Kesavan Nair. Competition between nucleation and early growth of ferrite from austenite - studies using cellular automaton simulations. *Acta Materialia*, 46(17):6291 – 6303, 1998.

[100] S.G.R. Brown. Simulation of diffusional composite growth using the cellular automaton finite difference (cafd) method. *Journal of Materials Science*, 33(19):4769–4773, 1998.

[101] J. Geiger, A. Roosz, and P. Barkoczy. Simulation of grain coarsening in two dimensions by cellular-automaton. *Acta Materialia*, 49(4):623 – 629, February 2001.

[102] W. Yu. *Cellular automata modelling of austenite grain coarsening during reheating*. PhD thesis, University of Sheffield (United Kingdom), 2002.

[103] W. Yu, C.D. Wright, S.P. Banks, and E.J. Palmiere. Cellular automata method for simulating microstructure evolution. *Science, Measurement and Technology, IEE Proceedings* -, 150(5):211–213, Sept 2003.

[104] E.J. Palmiere and S.P. Banks. Cellular automata modelling of grain coarsening during reheating and validation with the experimental results. *Acta Metallurgica Sinica (English Letters)*, 2:004, 2005.

[105] Y. J. Lan, D. Z. Li, C. J. Huang, and Y. Y. Li. A cellular automaton model for austenite to ferrite transformation in carbon steel under non-equilibrium interface conditions. *Modelling and Simulation in Materials Science and Engineering*, 12(4):719, 2004.

[106] C. Zheng, N. Xiao, D. Li, and Y. Li. Microstructure prediction of the austenite recrystallization during multi-pass steel strip hot rolling: A cellular automaton modeling. *Computational Materials Science*, 44(2):507 – 514, 2008.

[107] K.G.F. Janssens, J.N. Reissner, and F. Vanini. Thermodynamic and kinetic coupling of a random grid cellular automaton for the simulation of grain growth. *Advanced Engineering Materials*, 4(4):200–202, 2002.

[108] K. G. F. Janssens, D. Raabe, E. Kozeschnik, M. A. Miodownik, and B. Nestler. *Computational materials engineering: an introduction to microstructure evolution*. Academic Press, 2010.

[109] K. G F Janssens. Random grid, three-dimensional, space-time coupled cellular automata for the simulation of recrystallization and grain growth. *Modelling and Simulation in Materials Science and Engineering*, 11(2):157, March 2003.

[110] Ye. Vertyagina, M. Mahfouf, and X. Xu. 3d modelling of ferrite and austenite grain coarsening using real-valued cellular automata based on transition function. *Journal of Materials Science*, 48(16):5517–5527, August 2013.

[111] Ye. Vertyagina and M. Mahfouf. A 3d cellular automata model of the abnormal grain growth in austenite. *Journal of Materials Science*, 50(2):745–754, January 2015.

[112] B. Zhu, Y. Zhang, C. Wang, P. Liu, W.K. Liang, and J. Li. Modeling of the austenitization of ultra-high strength steel with cellular automation

method. *Metallurgical and Materials Transactions A*, 45(7):3161–3171, 2014.

[113] C. Zheng, D. Raabe, and D. Li. Prediction of post-dynamic austenite-to-ferrite transformation and reverse transformation in a low-carbon steel by cellular automaton modeling. *Acta Materialia*, 60(12):4768 – 4779, 2012. 2012.

[114] T. Zhou, R.J. O'malley, and H.S. Zurob. Study of grain-growth kinetics in delta-ferrite and austenite with application to thin-slab cast direct-rolling microalloyed steels. *Metallurgical and Materials Transactions A*, 41(8):2112–2120, August 2010.

[115] K. Kremeyer. Cellular automata investigations of binary solidification. *Journal of Computational Physics*, 142(1):243 – 263, 1998.

[116] H. Yang, C. Wu, H.W. Li, and X.G. Fan. Review on cellular automata simulations of microstructure evolution during metal forming process: Grain coarsening, recrystallization and phase transformation. *Science China Technological Sciences*, 54(8):2107–2118, Aug 2011.

[117] O. I. Frette, G. Virnovsky, and D. Silin. Estimation of the curvature of an interface from a digital 2d image. *Computational Materials Science*, 44(3):867 – 875, 2009.

[118] J.W. Bullard, E.J. Garboczi, W.C. Carter, and E.R. Fuller. Numerical methods for computing interfacial mean curvature. *Computational Materials Science*, 4(2):103 – 116, 1995.

[119] N. Xiao, C. Zheng, D. Li, and Y. Li. A simulation of dynamic recrystallization by coupling a cellular automaton method with a topology deformation technique. *Computational Materials Science*, 41(3):366 – 374, 2008.

[120] N. Yazdipour, C.H.J. Davies, and P.D. Hodgson. Microstructural modeling of dynamic recrystallization using irregular cellular automata. *Computational Materials Science*, 44(2):566 – 576, 2008.

[121] J. Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, 48(1):149 – 182, 1994.

[122] J. Kari. Reversibility of 2d cellular automata is undecidable. *Physica D: Nonlinear Phenomena*, 45(1):379 – 385, 1990.

[123] N. Margolus. Physics-like models of computation. *Physica D: Nonlinear Phenomena*, 10(1):81 – 95, 1984.

[124] J. K. Mackenzie and M. J. Thomson. Some statistics associated with the random disorientation of cubes. *Biometrika*, 44(1-2):205–210, 1957.

[125] J. K. Mackenzie. Second paper on statistics associated with the random disorientation of cubes. *Biometrika*, 45(1/2):229–240, 1958.