EVOLUTION OF ANNEALING TWINS

IN SPUTTERED CU THIN FILMS

By

CHANG-KYU YOON

A thesis submitted in partial fulfillment of
the requirement for the degree of

MASTER OF SCIENCE IN MATERIALS SCIENCE AND ENGINEERING

WASHINGTON STATE UNIVERSITY
School of Mechanical and Materials Engineering

AUGUST 2009

To the Faculty of Washington State University:

The members of the Committee appointed to examine the thesis of CHANG-KYU YOON find it satisfactory and recommend that it be accepted.

____
David Field, Ph. D., Chair

____
David Bahr, Ph. D.

____
Sergey Medyanik, Ph. D.

ACKNOWLEDGMENT

Before I begin to acknowledge all amazing people whom I met at Washington State University, I am now thinking of what all I have gone through to be here and I must admit that this thesis would not have been completed without a sincere advices and encouragement of my advisor, Professor David Field. I would like to thank Dr. Field not only for his superb guidance and academic advice but also being a great role model and mentor. He has taught and shown me how to become a real knowledgeable and philosophical man with humor. He always has endured and encouraged me to pursue a goal consistently with positive attitude. Dr. Field, It is definite blessing for me to meet you in my whole life. I am forever indebted to you. Thank you so much for being a trustworthy and warm adviser.

My deepest and sincerest gratitude should also go to my family, who have always supported and prayed for me. My wife, Jeong-Huyn, I have taken a spiritual rest and have completed this thesis through your prayer and sacrifice. My daughter, Esther, you are always be my joy and energy. In addition, my family living in Korea, I know you are always praying for me and I can say I am nothing without your all love.

Finally, I give my most sincere appreciation to my God with following confess:
O give thanks unto the LORD, for he is good: for his mercy endures forever (Psalms 107:1).

EVOLUTION OF ANNEALING TWINS

IN SPUTTERED CU THIN FILMS

Abstract

by Chang-Kyu Yoon, M.S.
Washington State University
AUGUST 2009

Chair: David P Field

The Monte Carlo Potts model with N-fold method has been used to simulate grain structure evolution in thin Cu films according to energetic competition principles. The energetic considerations of surface/interface, grain boundary and strain energy factors were applied to determine grain growth and crystallographic texture evolution as a function of film thickness such as 100 nm, 500 nm and 800 nm.

Furthermore, annealing twins in FCC metals with low stacking fault energy are typically related to the parent by 60° rotation about the <111> crystal direction. These twins can change the crystallographic texture and grain boundary character distribution in films and line structures. These factors can be simulated through specific criteria that arbitrarily insert twin grains into the structure through grain boundary energy considerations.

Four different types of microstructures have been observed experimentally and these have been simulated by the Monte Carlo technique. The focus is on the texture and grain structure evolution and how these can be modeled using the Monte Carlo Potts model. Using the overall energy minimization concept to model microstructure evolution in Cu thin films, the capabilities and difficulties of using Monte Carlo Potts Model are demonstrated

.

TABEL OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

Chapter Three

Chapter Four

LIST OF FIGURES

Chapter Two

Chapter Three

Chapter Four

Chapter Five

CHAPTER 1

INTRODUCTION

The characteristic microstructure of a thin film affects its function in electronic applications. Modern Cu films and line structures, which have replaced aluminum due to its high electrical and thermal conductivity, often contain a large fraction of annealing twins. Annealing twins in FCC metals with low stacking fault energy are related to the parent by 60° rotation about the <111> crystal direction. These twins, which are associated with recrystallization and grain growth, can change the crystallographic texture and grain boundary character distribution in films and line structures. Because these annealing twins have properties that affect microstructure evolution, it is important to include these in considerations of structural evolution. Grain growth in thin films generally plays an important role in defining the microstructural characteristics of the films [1].

During annealing, normal and abnormal grain growth occur in a manner that lowers the total energy accumulated in the film. The total energy is the sum of surface energy, interface energy, strain energy and grain boundary energy [2]. These combined energy variables have a tendency to be minimized during grain growth and this generally results in texture transformation from the original state. It is well known that texture evolution during grain growth of thin films depends upon a competition between principles of surface/interface energy minimization and strain energy minimization [1, 3, 4]. {111} out of plane fiber textures result in thinner films where surface energy minimization controls grain growth, and {100} out of plane fiber textures are observed in thicker films at higher temperatures as a consequence of strain energy minimization [1]. In the FCC Cu films, the {111} close-packed surface is associated with the lowest surface free energy, so strong {111} out of plane texture is present with little dependence on deposition and annealing conditions [5].

The conventional Monte Carlo Potts model has been shown to be capable of making significant and precise predictions of grain growth by various researchers since the 1990s [6-10]. A two dimensional model has been applied with the assumption that thin films are two dimensional in character. Of course, three-dimensional simulations yield more realistic predictions but at the cost of greater computational effort. In this work it is shown that thickness dependence of structure evolution can be captured by a two-dimensional model. The focus is on the texture and grain structure evolution and how these can be modeled using the Monte Carlo Potts model according to the overall energy minimization concept.

Experimentally, many researchers have attempted to understand grain structure (grain growth and grain boundary properties) and crystallographic texture as a function of various factors such as stress state [11-13], stacking fault energy [14,15] , grain size [16,17] and film thickness [2, 5, 18, 19]. The purpose of this thesis is to demonstrate the capabilities and difficulties associated with including these important factors of the microstructure evolution with twins in Cu thin films by using Monte Carlo Potts Model.

CHAPTER 2

BACKGROUND

2.1 Recrystallization

In order to understand and analyze microstructure evolution in Cu thin films, it is required to know about recrystallization that occurs during deposition of thin films or during post deposition processing. Primary recrystallization includes nucleation of new grains, coarsening, and grain growth [20]. During isothermal annealing, the recrystallization process can be described by the volume fraction of material recrystallized as a function of time as shown in Figure 2.1. This figure indicates that the process consists of three parts such as an increasing rate of recrystallization at nucleation region, linear region of growth and finally a decreasing rate of recrystallization region as grain growth nears completion.

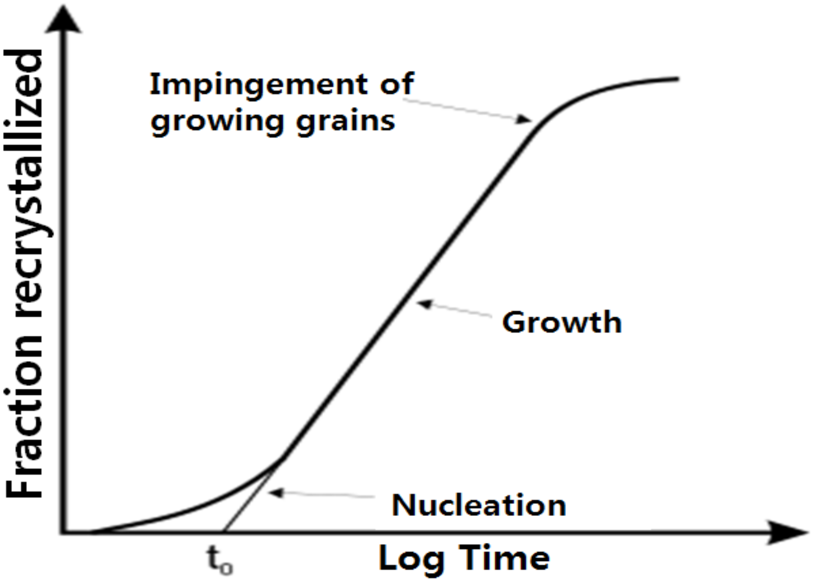


Figure 2.1 Typical recrystallization kinetics during isothermal annealing [20].

Research into the recrystallization behavior of materials has been performed over the past 50 years. Stanley and Mehl [21], and Anderson and Mehl [22] initially divided recrystallization into nucleation and growth contribution to the overall kinetics. Under consideration that recrystallization is a nucleation and growth phenomenon which is controlled by thermally activated processes with stored energy of deformation providing the driving force, the laws of recrystallization are established [20, 23, 24]. These are described as follows:

1. Deformation has to be sufficient to form a nucleus for the recrystallization and to provide a necessary driving force to maintain growth.
2. When recrystallization temperature decreases, the annealing time increases.
3. When strain increases, recrystallization temperature decreases. More rapid nucleation and growth occur at a lower temperature in a more highly deformed material.
4. The recrystallized grain size is related to the amount of deformation. The number of nuclei or the nucleation rate is affected by strain more than the growth rate. A higher strain can provide more nuclei per unit volume, resulting in a smaller final grain size.
5. When the amount of deformation is fixed, the recrystallization temperature can increase by a larger initial grain size and higher deformation temperature. A large grain size has relatively few grain boundaries that are preferred nucleation sites so the nucleation rate is lowered and the recrystallization is slower at higher temperature. In addition, more recovery occurs at high temperature of deformation and the stored energy is thus lower than for a similar strain at a lower deformation temperature [20].

These recrystallization laws still provide a useful guide to the overall behavior of recrystallization. This classical recrystallization background gives insight into materials and processing parameters in thin films. Thin films have high surface area to volume ratio, which leads to greater surface/interface energy minimization and the mechanical effects due to the attachment of films to substrates. To understand recrystallization phenomena at a much finer microstructural level, the classical nucleation and growth phenomena are reviewed in brief following the work of Christian [25].

2.1.1 Nucleation

In practice, homogeneous nucleation occurs only in extremely pure solids with very few structural defects. More usually, impurities or strained regions (phase boundaries) of the lattice cause nuclei formation with a much smaller free energy of activation than that of homogeneous nucleation [25]. The free energy for heterogeneous nucleation can be explained as a function of the contact angle as follows:

$$\Delta G_{het} = \Delta G_{hom} * f(\theta) \qquad (1)$$

$$f(\theta) = \frac{1}{4}(2 - 3\cos\theta + \cos^3\theta)$$

Recrystallization and nuclei are formed and become stable when the energy advantage gained by removing stored energy from the bulk is greater than the additional energy required to form the new grain boundary area shown in Figure 2.2. Nucleation depends on a reduction in the net surface energy needed to form a nucleus [25].

If the strain energy effects are neglected, the following general free energy of formation can be written

$$\Delta G^{s} = \left(\eta^{\beta} r^{3} / v^{\beta}\right)\left(g^{\beta} - g^{\alpha}\right) + r^{2}\left\{\eta^{\alpha\beta}\sigma^{\alpha\beta} + \eta^{\alpha S}\left(\sigma^{\beta S} - \sigma^{\alpha S}\right)\right\} \qquad (2)$$

Where, α phase is the deformed or strained state in contact with a surface S and ß is the strain free recrystallized grain of developing nucleus that is also in contact with S.

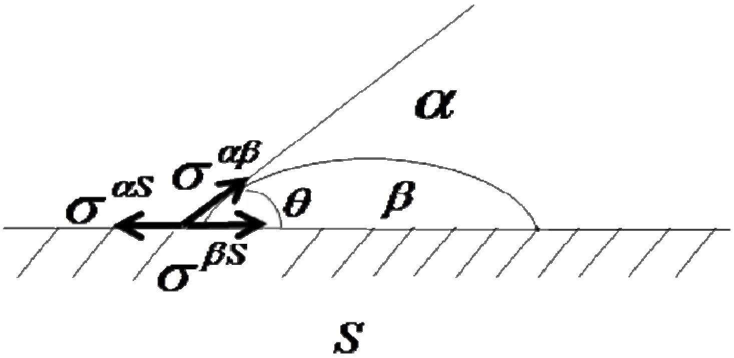


Figure 2.2 The formation of ß nucleus on a flat surface S.

If the surface energy $\sigma^{\alpha\beta}$, of the α-ß interface is isotropic, the ß nucleus will be bounded by spherical surfaces of radius r, except where it is in contact with S. The volume of ß is $\eta^{\beta} r^{3}$ and its surface area of contact with the α phase is $\eta^{\alpha\beta} r^{2}$, where $\eta^{\alpha\beta}$ and η^{β} are the shape factors. The area of contact with ß is equal to $\eta^{\alpha S} r^{2}$, the area of α-S interface destroyed when the ß is formed and $\sigma^{\alpha\beta}, \sigma^{\alpha S}, \sigma^{\beta S}$ are the surface energies per unit area of the respective surfaces. The free energy of formation to create a nucleus of critical size, r$_c$, when $\frac{\partial G^{S}}{\partial r} = 0$.

This gives the critical energy difference between strained and un-strained (recrystallized) lattice to be

$$\Delta G_c^S = \frac{4}{27}\frac{\left\{\eta^{\alpha\beta}\sigma^{\alpha\beta} + \eta^{\alpha S}\left(\sigma^{\beta S} - \sigma^{\alpha S}\right)\right\}^3 \left(v^{\beta}\right)^2}{\left(\eta^{\beta}\right)^2 \left(g^{\alpha} - g^{\beta}\right)^2} \tag{3}$$

2.1.2 Grain Growth

Grain growth may be categorized as normal grain growth or abnormal grain growth [20]. Normal grain growth leads to a uniformly coarsened microstructure that has a relatively narrow grain size range and shape with a time independent grain size distribution profile. On the other side, during abnormal grain growth, a few grains in the microstructure grow and consume the matrix of small grains and a bimodal grain size distribution develops, which is a discontinuous process as shown in Figure 2.3 [20, 27].

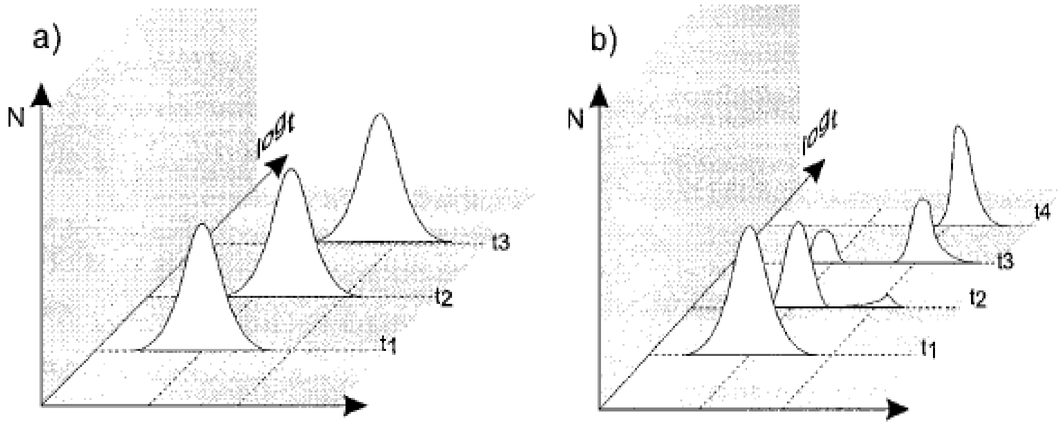


Figure 2.3 Schematic representation of grain size distribution during (a) Normal grain growth (continuous grain growth) and (b) Abnormal grain growth (discontinuous grain growth) [20, 27].

Many researchers have performed to verify the critical characteristics of grain growth and distribution of grain sizes in thin films. It has been shown that surface/ interface energy minimization governs the grain growth, for example, (111) in FCC metals at elevated deposition temperature of thin structures [1, 3, 4, 5] and grain growth in films often stagnates when the average grain diameters is 2 to 3 times the film thickness. Surface grooves are the apparent cause of stagnant columnar grain structure [3, 26].

The main factors to control grain growth in thin films are four elements: temperature, texture, precipitates and specimen size [3, 20], which is described as below:

1. Temperature: Grain growth includes high angle grain boundary migration and the kinetics of grain boundary motion are strongly temperature dependent. Because the driving force for grain growth is small, the rate of grain growth is significant only at very high temperature.

2. Texture: Strongly textured materials contain a lot of low angle boundaries of low energy so texture can be a reduced driving force for grain growth.

3. Precipitates: When precipitates occur in a thin film, the grain growth is inhibited by the pinning of grain boundaries. This process leads to a local energy increase and thus precipitates exert a drag force on grain boundary motion.

4. Specimen size: The grain growth rate is decreased when the grain size becomes greater than the sheet specimen thickness. The reason is that the columnar grains are curved only in one direction so the driving force is diminished.

Grain growth in thin films generally plays an important role in defining the microstructural characteristics of the films [1]. One of the important reasons that grain growth in thin films is different from the grain growth in bulk materials is the interface of the film with the substrate. The top surface of the film plays an important role in suppressing normal grain growth and promoting secondary or abnormal grain growth [1, 3]. Because the energy of the surfaces and/or interfaces is anisotropic, secondary grains generally have restricted crystallographic orientations. These orientations are affected by the atomic structure of the substrate interfaces as well as the environment of the top surface of the film. Therefore, secondary grain growth which is driven by surface and interface energy leads to an evolution in the preferred crystallographic orientation of a film as well as the average grain size [3].

2.2 Annealing Twins

Annealing twins are observed in a variety of recrystallized FCC metals and alloys. These twins are exhibited through four prominent morphologies which are schematically shown in Figure 2.4 [29].

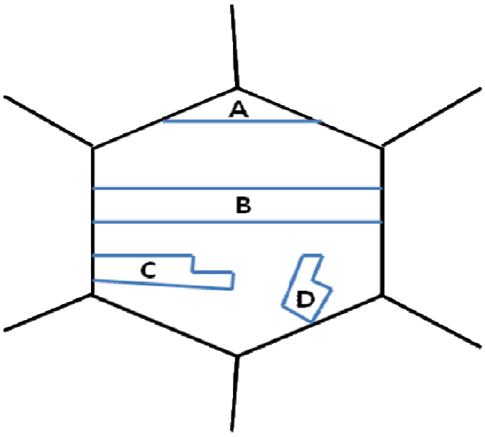


Figure 2.4 Schematic descriptions of various annealing twins morphologies observed in FCC metals [29].

In 1963, annealing twin nuclei which consist of stacking fault packets having complicated morphology were described by Dash and coworker [30]. In 1978, Meyers and Murr [31] introduced the “pop-out” model for formation of twins with stacking fault packets that pop out from ledges present on grain boundaries. Recently, Mahajan [29], Pande [32] and their colleagues have developed a unified model to explain the formation of annealing twins in mechanistic and the twin morphologies shown in Figure 2.4.

After Fullman and Fisher (1951), Figure 2.5 shows a standard theory of twin formation at a three-grain junction where grain G3 is growing at the expense of G1 and G2 [25, 34].

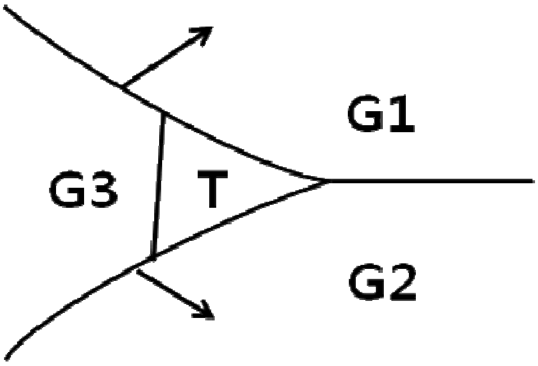


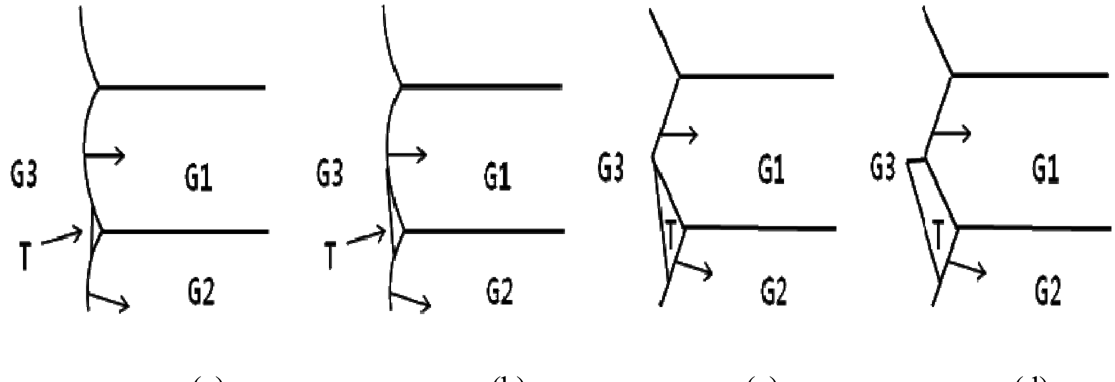
Figure 2.5 The formation of annealing twins at a three-grain junction [25].

According to their model, the concept of twin formation lies in the energy minimization consideration. During grain growth, when the energy of the boundary between grain G3 and T is low enough to reduce the total energy state even though the total grain boundary area is increasing, the twin will form. The occurrence of T can reduce the total interfacial free energy if

$$o^{G3T}\sigma^{G3T}+o^{TG1}\sigma^{TG1}+o^{TG2}\sigma^{TG2}<o^{TG1}\sigma^{G3G1}+o^{TG2}\sigma^{G3G2} \qquad (4)$$

Where o^{ij} are the surface areas between grain *i* and *j* and σ^{ij} are the specific free energies between grain *i* and *j*. When the inequality is sustained, a growth fault may occur at some point leading to the macroscopic regions for formation of T which is twinned with respect to grain G3. When the twin interface is fully coherent, the twin boundary energy will be low and thus the reasonable probability of total interfacial free energy inequality is valid [25]. From the data of Murr (1975), the coherent twin boundary energy is 24 mJm^{-2}, the incoherent twin boundary is 498 mJm^{-2}, and the high angle grain boundary energy is 625 mJm^{-2} for copper [35]. Therefore, there may be a reduction of total boundary energy by formation of a coherent twin boundary because the energy is significantly lower than that of general boundaries.

The Fullman-Fisher theory also applies to the situation leading to the formation of a terminated twin shown in Figure 2.6. If a twin T has a close orientation to that of grain G2, the T and G2 interface have low energy and the configuration with the twin will have a lower energy than the configuration without it. The coherent twin interface becomes tangential to the main migrating grain boundary, after which another interface must form. The number of twin lamellae per unit grain boundary area depends only on the number of new grain contacts which have been made during growth, thus the sequences of Figure 2.6 is a usual method of introducing a twin [25].



(a) (b) (c) (d)

Figure2.6 Formation of a twin with a coherent and a non coherent interface during grain boundary migration [25].

In general, the formation of annealing twins is associated with grain growth that occurs during annealing at relatively high temperature. Twins form during recovery, primary recrystallization or during grain growth following recrystallization [20]. There has been interest in improving the properties of alloys by controlling the grain boundary character (grain boundary engineering), and maximizing the number of low Σ boundaries such as Σ3 twins through thermo-mechanical processing [11, 12, 33]. The Σ indicates the reciprocal of the ratio of coincidence site lattice (CSL) volume to the primitive lattice volume. The CSL is the lattice formed by superposition of lattice sites from two crystals and used in the context of structure analysis to categorize grain boundary geometry [36]. For certain orientation relationships having a good fit between the grains, the coherent twin (Σ3) boundary, low angle grain boundaries (Σ1) and high mobility (Σ7) boundaries in FCC metals are good examples of boundaries with special properties [20].

Twin boundaries are often flat and extend across an entire grain. Annealing twins in FCC metals with low stacking fault energy are typically related to the parent through a specific misorientation (Δg) of 60° rotation about the <111> crystal direction. For a coherent twin boundary, the twinning plane must be aligned with the boundary plane separating the twin from the parent. This means that the FCC primary recrystallized twin boundary plane must be aligned with one of the {111} planes in each of the neighboring crystals [37].

2.3 Crystallographic Characterization Technique

2.3.1 Orientation Mapping

Monte Carlo Potts model can inherently mapped onto a two dimensional (2D) microstructure by a series of FORTRAN programming codes. These codes have been modified to create specific files that can be analyzed by using TSL OIM analysis 4.5 version software. Orientation imaging microscopy or electron backscattered diffraction (EBSD) indicates an automated measurement and storage of orientations to depict the microstructure according to a pre-defined pattern of coordinates on the sampling plane of the specimen [36]. Automated EBSD provides spatially specific measurement of crystallographic orientation information on regular grids. Each sampling grid is stored with its orientation, spatial coordinates and often a pattern quality index. These data then allow entirely quantitative and accessible map outputs as illustrated in Figure 2.7, for example, which contains an orientation map of random nucleation and grain growth of a Monte Carlo simulation.

The application of orientation mapping can be categorized depending on the information such as spatial distribution of texture components, misorientation and interfaces, orientation perturbations within grains, true grain size/shape distributions and deformation (i.e. pattern quality) maps [36]. Specifically, misorientation which means the orientation difference between two individual orientation measurements plays an important role to identify potential twin boundaries. By inspecting a misorientation between neighboring measurements on the grid assigned particular colors in a map, boundaries can be identified as low angle, high angle, coherent twin, incoherent twin, etc. [37].

In addition, orientation color key at Figure 2.7 (f) represents that the orientation of the specimen coordinate system can be projected into the crystal coordination system. In other words, the orientation color key shows different colors to the crystal orientations aligned with a given direction. The reference system is the crystal coordinate system and the orientation is defined by the axes of the specimen coordinate system, e.g. rolling direction (RD), transverse direction (TD) and normal direction (ND)

2. 3. 2 Euler Angles

Euler angles specify a triplet of rotations about the Z, X and Z directions. The Euler angles refer to three rotations which transform the specimen coordinate system onto the crystal coordinate system. The definition of Euler angles adopted here is that according to Bunge, described as follows [36, 38]:

1. φ_1 about the normal direction (ND), transforming the transverse direction (TD) into (TD$^{'}$) and the rolling direction (RD) into (RD$^{'}$).
2. Φ about the axis RD$^{'}$
3. φ_2 about ND$^{''}$

The orientation of the crystal is in the form of Bunge Euler angles (φ_1, Φ, φ_2). The angle $\varphi 1$ and φ_2 range from 0 to 2π and the angle Φ range from 0 to π. These range forms a bounded space referred to as Euler space [38]. Analytically, the three rotations are expressed as parameters $g = g$ ($\varphi 1$, Φ, $\varphi 2$)

$$g_{\varphi_1} = \begin{pmatrix} \cos\varphi_1 & \sin\varphi_1 & 0 \\ -\sin\varphi_1 & \cos\varphi_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, g_{\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix}, g_{\varphi_2} = \begin{pmatrix} \cos\varphi_2 & \sin\varphi_2 & 0 \\ -\sin\varphi_2 & \cos\varphi_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (5)$$

By multiplication of these three matrices in order $g = g_{\varphi_1} \bullet g_{\phi} \bullet g_{\varphi_2}$, the Euler angles can be obtained as following matrix:

$$\begin{pmatrix} \begin{matrix} \cos\varphi_1\cos\varphi_2 \\ -\sin\varphi_1\sin\varphi_2\cos\phi \end{matrix} & \begin{matrix} \sin\varphi_1\cos\varphi_2 \\ +\cos\varphi_1\sin\varphi_2\cos\phi \end{matrix} & \sin\varphi_2\sin\phi \\ \begin{matrix} -\cos\varphi_1\sin\varphi_2 \\ -\sin\varphi_1\cos\varphi_2\cos\phi \end{matrix} & \begin{matrix} -\sin\varphi_1\sin\varphi_2 \\ +\cos\varphi_1 co\,s\varphi_2\cos\phi \end{matrix} & \cos\varphi_2\sin\phi \\ \sin\varphi_1\sin\phi & -\cos\varphi_1\sin\phi & \cos\phi \end{pmatrix} \qquad (6)$$

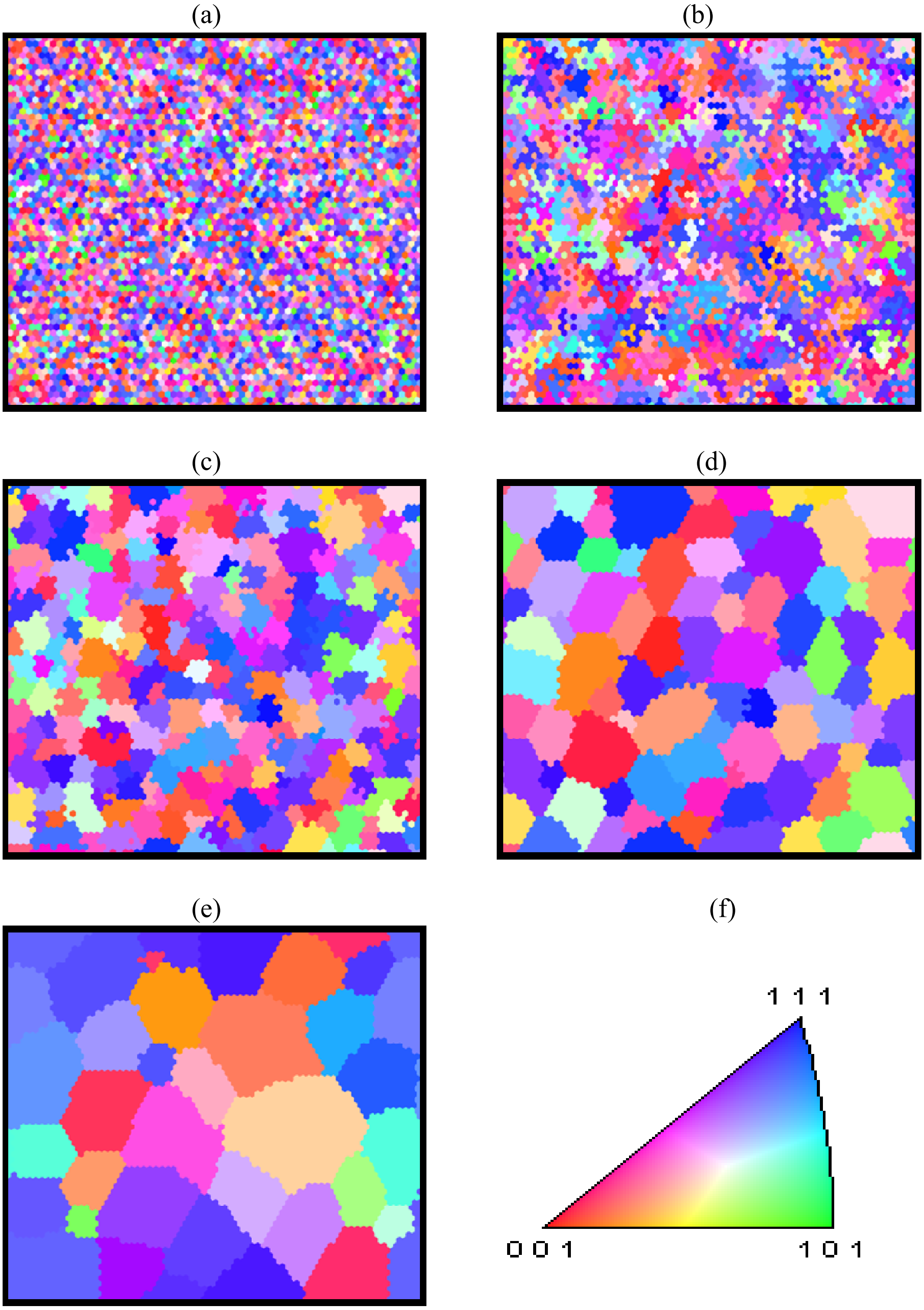


Figure 2.7 Random nucleation and grain growth simulation results (a) - (e) analyzed by using OIM software and (f) orientation color key.

CHAPTER 3

MICROSTRUCTURAL SIMULATION

In many circumstances, computer simulation has been applied to gain insight into a physical problem for obtaining new theoretical understanding and interpreting the differences between ideal and practical situations. In addition, computer simulation has played an important role in reducing a large expense of practical experiments in industry. Many techniques have been developed to simulate microstructure evolution in materials science. Because microstructure governs the mechanical, electrical and thermal properties of engineered materials, it is critical to understand microstructure. One of the simulation methods applied to microstructure evolution is the Monte Carlo (MC) methods that Anderson and colleagues developed for simulation of grain growth in two and three dimensions [20].

3.1 Monte Carlo Potts Model

In general, Monte Carlo refers to any method that utilized sequences of random numbers to perform statistical simulation. The genesis of the Monte Carlo Potts model lies in the development of the Ising model (1925) [6]. The Ising model represents magnetized materials as two states of spin-up and spin-down at every point on a discretized domain structure. Potts (1952) later generalized the Ising model with adding a concept of Q states for each particle in a system. Considering the Ising model, only two degenerate spin states (Q=2) are equivalent to a Potts model. The Potts model as it is called a Large-Q model has been extensively used to simulate behavior of materials such as recrystallization, grain growth and texture evolution. The Monte Carlo Potts model, modified from that developed by Hassold and Holm [8] has been used to simulate two dimensional microstructure evolution. This simulation method divides materials into a small area or particle which is assumed to be a nucleus or grain. This nucleus or grain has its own crystallographic orientation (*g*) expressed by Bunge Euler angles indicated by spin variables ranging from 1 to Q. The Q-states are assigned total orientations (*g*) which are randomly selected from an input file each having unique orientations and these orientations can be changed to that of an adjacent grain according to the effect of this change on the overall energy state.

The unit of simulation time is called Monte Carlo Step (MCS) and represents an integer time increment [8]. One iteration represents 1/N time increments and N iterations are required for each site in the lattice to have a chance to change its state. Therefore, 1 unit of system simulation time elapses after N iterations. The increase in the mean grain size in accordance with time reduces the total lengths of grain boundaries and thus the system energy through two algorithms of Conventional Potts Model (CPM) Algorithm and the N-Fold way of the ISING Model. For each MCS of domain growth in conventional Potts Model Algorithm, the following procedure is performed [8]:

CPM

1 Set the energy of each site in the system to zero at the beginning $E_i = 0$, $E_f = 0$

2 Pick a site i: $i \in \{1,……, N\}$ and assign an index number (orientation) randomly

3 Pick a spin value S_i: $S_i \in \{1,.......,Q\}$ at random

4 For each site i, if a randomly picked spin value of S_i is not equal to the spin of the neighbors spin (S_j), the energy state is E_i.

```
5    Randomly picked spin value of $S_i$ is not equal to the $S_j$, the energy state is $E_f$.
6    $\Delta E = E_f - E_i$
7    Flip (reorientation) site $i$ to spin $S_i$ with probability $P(\Delta E)$
8    Increment time by 1/N Monte Carlo steps (MCS)
```

Each MCS requires N iterations (lines 4 and 5) plus N iterations of some constant time operations (line 1-3 and 6-8) and typical Potts model applications may require as many as 10^6 - 10^8 MCS per simulation, so obviously an inefficient linear dependence of computing time per MCS exists in the conventional Monte Carlo algorithm. Hence, as grains increase in size, fewer and fewer iterations of the conventional Monte Carlo loop will result in spin flips actually occurring. As a result, for the purpose of speeding up the simulation time, the extensive simulation algorithm of N-fold way was applied [8]. When the grain growth was nearly complete, most sites in the lattice were surrounded by sites of similar orientation. Therefore, the probability of changing orientation is low and computational time is saved by using the N-fold way as opposed to the conventional Monte Carlo method. Initially, the processing time of the N-fold method is slower than Monte Carlo Potts algorithm. Because the N-fold way is set to very fine grained domain structure at the beginning, a high total system activity is required. When the domains have coarsened and the average domain radius (grain size) reaches a critical radius according to individual user assumptions, the N-fold way becomes much faster than the Monte Carlo Potts algorithm with decreasing the net system activity [8].

3.2 Energy Minimization Model

The concept of interaction between sites is based on an energy minimization model. The total system energy can be described by interacting neighboring spins with each other. When the spins are the same, no interaction energy is contributed whereas different neighboring spins lead to a contribution to the system energy [6]. In order to decrease the total interacting boundary area of a system, boundaries move toward their centers of curvature. High energy state grains have a tendency to reduce their boundary curvature and this curvature-driven diffusive coarsening of grains dominates the grain growth [8]. Randomly selected sites or grains with adjacent neighboring grains have their unique orientations indentified by Euler angles and these are used to help determine the proposed energy change of the position (ΔE) assuming that the orientation was changed to that of one of its neighbors. In this simulation, the system provided a driving force to reach a minimum energy state through re-orientation of each site with probability P (ΔE) given by

$$p\left(\Delta E\right)=\begin{cases} 1 & if\,\Delta E\leq 0 \\ \exp\left(-\Delta E/k_B T\right) & if\,\Delta E>0 \end{cases} \tag{7}$$

ΔE is the energy change, T is annealing temperature, and k_B is the Boltzmann constant. If the energy change is less than or equal to zero, the site would reorient to another orientation with the transition probability given as one. If the energy change is greater than zero, the reorientation changes with probability exp ($-\Delta E/K_BT$). According to the energy minimization competition, some nuclei are consumed while others coarsen and grow, resulting in a preferred orientation.

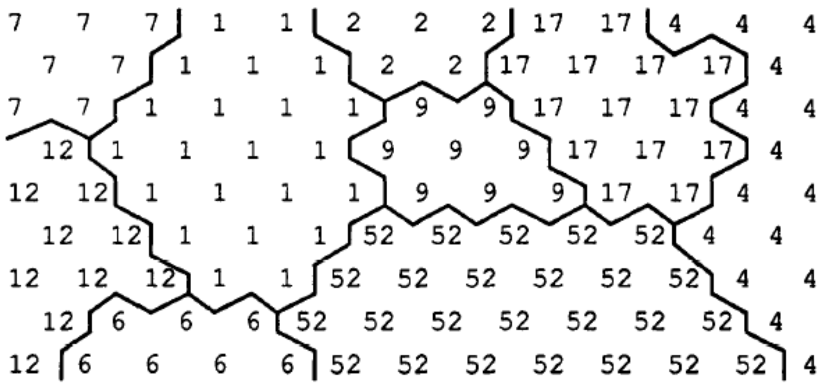


Figure 3.1 Diagram of the triangular 2D lattice, showing orientation numbers at each site and boundary drawn between sites with dissimilar orientations [7].

The Figure 3.1 shows the triangular 2D lattice with orientation numbers at each site and boundaries between sites with dissimilar orientations [7]. This two dimensional (2D) microstructural evolution of grain structure has been tracked by a change of each initially established orientation according to energy considerations suggested by Park *et al* [2]. The surface/interface, grain boundary and strain energies lead to the preferred final textures of thin films. These energetic factors are described in the following Hamiltonian

$$\mathrm{E} = \sum_{i=1}^{N}\left(\gamma_{\mathrm{surface}}(\mathrm{g}) + \gamma_{\mathrm{interface}}(\mathrm{g}) + \gamma_{\mathrm{strain}}(\mathrm{g}) + \frac{1}{2}\sum_{j=1}^{m}\gamma_{\mathrm{grain\ boundary}}(\Delta\mathrm{g})\right) \tag{8}$$

where, E is the total energy of a system, *N* is the total number of lattice sites, $\gamma_{\mathrm{surface}}(\mathrm{g})$ is the surface energy of a unit lattice as a function of orientation (g) according to normal direction (ND)(001) out of plane, $\gamma_{\mathrm{interface}}(\mathrm{g})$ is the interface energy as a function of orientation (g), $\gamma_{\mathrm{strain}}(\mathrm{g})$ is the bi-axial strain energy as a function of orientation (g), $\gamma_{\mathrm{grain\ boundary}}(\Delta\mathrm{g})$ is the grain boundary energy as a function of misorientation (Δg) between two grains of different index and *m* is 6, considering the number of nearest neighbors in a triangular (1,2) lattice (hexagonal array) indicated in Table 3.1. This governing Hamiltonian is composed of three individual elements of surface, interface and strain energy terms and grain boundary terms. These energetic considerations are seen in the software shown in appendices A and B.

Table 3.1 Listing of 2D lattice types with geometries and anisotropies [6].

Lattice Type	Wulff Shape	Coordination Number	Grain Growth
Square (4.1)	Square	4	Inhibited
Triangular	Hexagonal	6	Normal
Square(1,2)	Octagon	8	Normal
Triangular (1,2)	Dodgecagon	18	Normal

3.2.1 Surface/ Interface Energies

For the purpose of comparing surface energy $\gamma_{surface}(g)$ in this model, the broken-bond energy concept given by Sundquist [39] was used. The number of unsatisfied broken bonds will vary from plane to plane so the different atomic packing system has the various crystallographic planes and the surface energy is dependent on crystallographic orientation (g). The misorientation of an arbitrary grain with the neighboring grain orientations allows one to determine the angular difference of the surface normal from the reference. This angle is used for comparing surface energy by following a simple relationship:

$$\frac{\gamma_0\{\mathrm{hkl}\}}{\gamma_0\{\mathrm{h_0k_0l_0}\}} = \cos\theta_{hkl\text{ - }h_0k_0l_0} = \frac{A \bullet B}{|A||B|} \tag{9}$$

where $\gamma_0\{hkl\}$ is the surface energy at 0°K of the $\{hkl\}$ plane and θ is the angle between the $\{hkl\}$ plane and the reference plane. The reference surface normal unit vector (hkl) of each grain is (001) // ND since this is the high energy plane in an FCC structure. The initial value is 2.610 J/m^2 for (100) surface at 200°C with temperature dependence given by

$\frac{d\gamma_{hkl}}{dT} = -5.0\times10^{-4} J/(m^2\,{}^\circ C)$ [39, 40].

In addition, the interface energy is assumed to be identical to the surface energy. Since there is no change in barrier layer or stacking sequence, the interface energy for all films is similar. The interface energy can be expressed by

$$\gamma_{\text{surface}}\left(g\right)=\gamma_{\text{interface}}\left(g\right)=\int_{A_{hkl}}\gamma_{hkl}\cdot dA_{hkl} \tag{10}$$

3.2.2 Strain Energy

The strain energy is given by the following equation:

$$\gamma_{\text{strain}}\left(g\right)=h\cdot e^{2}\cdot\int_{A_{hkl}}M_{hkl}\cdot dA_{hkl}$$

$$M_{hkl}=C_{11}+C_{12}+K-\frac{2\left(C_{12}-K\right)^{2}}{C_{11}+2K}$$

$$K=\left(2C_{44}-C_{11}+C_{12}\right)\left(h^{2}k^{2}+k^{2}l^{2}+l^{2}h^{2}\right) \tag{11}$$

$$e=\int_{T_{dep}}^{T_{gg}}\left(\alpha_{s}-\alpha_{f}\right)dT\cong\Delta\alpha\cdot\Delta T$$

In the above equations, *h* is the thickness of the films, *e* is the elastic strain which is a function of the temperature difference, M_{hkl} is the biaxial modulus for the {*hkl*} oriented grain, *Δα* is the mismatch in coefficients of thermal expansion, and stiffness constants of C_{11} = 170.2 GPa, C_{12} = 123.2 GPa, C_{44} = 75.4 GPa at 25°C with temperature gradient effects included as

$\frac{dC_{11}}{dT}=-0.0353\,\text{GPa/°C}$, $\frac{dC_{12}}{dT}=-0.0153\,\text{GPa/°C}$, $\frac{dC_{44}}{dT}=-0.0277\,\text{GPa/°C}$ [41].

3.2.3 Grain Boundary Energy

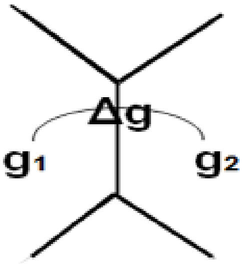
The grain boundary energy can be described by

$$\gamma_{\text{grain boundary}}\left(\Delta g\right)=h\left[L_{LAGB}\gamma_{LAGB}+L_{ICTB}\gamma_{ICTB}+L_{CTB}\gamma_{CTB}\right] \quad (12)$$

L_{LAGB}, L_{ICTB} and L_{CTB} are the lengths of the large angle grain boundary, incoherent twin boundary and coherent twin boundary and γ_{LAGB} , γ_{ICTB} and γ_{CTB} are the energies of large angle grain boundaries, incoherent twin boundaries and coherent twin boundaries. The large angle grain boundary energy is assumed to be 0.625 J/m^2 at 925°C, the incoherent twin boundary energy is 0.498 J/m^2 at 950°C and the coherent twin boundary energy is 0.024 J/m^2 at 800°C with temperature effects of

$$\frac{d\gamma_{LAGB}}{dT}=-1.0X10^{-4}J/m^2{}^\circ C \, , \frac{d\gamma_{ICTB}}{dT}=-1.0X10^{-4}J/m^2{}^\circ C \, , \frac{d\gamma_{CTB}}{dT}=-2.0X10^{-5}J/m^2{}^\circ C$$ [42].

Grain boundaries can be characterized by the misorientation (Δg) between two neighboring sites [43]. The grain boundary energy increases with increasing angle of misorientation to 15°. With the exception of twin boundaries, the grain boundaries with larger than 15° misorientation have similar grain boundary energies considered to be high angle grain boundaries (HAGB). By analyzing the misorientation (Δg) between neighboring sites, twins and twin boundaries can be identified. The next equation shows the misorientation Δg between two grains 1 and 2, where T_i is the 24 symmetry operators, superscript T denotes transpose.



$$\Delta g = g_1 \bullet g_2^T$$

$$\Delta g_{eq} = T_i \bullet \Delta g \qquad (13)$$

T_i − symmetry operator
(24 matrices)

Figure 3.2 Schematic of misorientation or the difference in orientation between two grains.

Geometrically, annealing twins can form and relate to the parent through specific misorientation (Δg) of 60° with <111> axis. For coherent twins, the twinning plane has to be aligned with a grain boundary plane position [37] and the local relationships between surface energy and grain boundary energy can be considered in the following way [20, 44]:

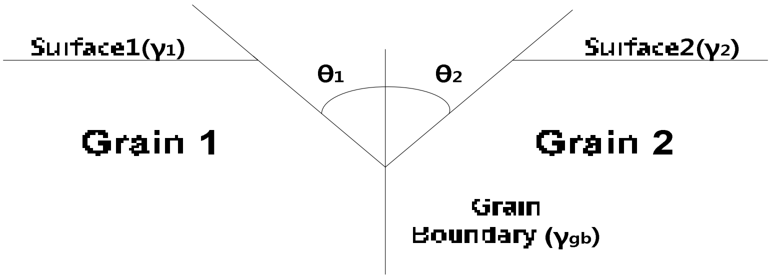


Figure 3.3 Schematic illustration of grain boundary energy interacting with free surface energies

$$\cos\theta 1 = \frac{\gamma 1}{\gamma gb}, \quad \theta_1 = \cos^{-1}\left(\frac{\gamma 1}{\gamma gb}\right) \qquad (14)$$

$$\cos\theta 2 = \frac{\gamma 2}{\gamma gb}, \quad \theta 2 = \cos^{-1}\left(\frac{\gamma 2}{\gamma gb}\right)$$

In this simulation, the grain boundary grooving phenomenon at Figure 3.3 was considered. Thermal grooving forms in order to achieve a capillary force balance [44]. Because grain boundary grooves can develop during annealing, Young's equation (14) was applied to our model for 300°C annealing condition and grain boundary area increased through the grooves as grain growth took place. The software to perform this operation is given in appendix C.

CHAPTER 4

SIMULATION RESULTS

For the Monte Carlo code used in this study, the orientations marked as Bunge Euler angles of grains were compared and changed for grain growth as a function of the energetics introduced in previous research [2, 40, 41] summarized by equations (10)-(12). Three different film thicknesses were set for the simulations; 100 nm, 500 nm and 800 nm. The deposition temperature was 25°C and the annealing temperature was 300°C. In this simulation, as Monte Carlo steps (MCS) increased, the texture evolved as growth conditions favored a lowering of system energy.

4.1 1st case (100nm film thickness at 300°C annealing temperature)

Previous experimental results, given in Table 4.1 [2], indicate that the 100nm-thick sputtered film has a {111} out of plane fiber texture. This result was compared with this simulation result shown in Figure 4.1. Figure 4.1 shows the simulation of microstructure evolution at 100 nm film thickness and 300°C annealing temperature conditions. Figure 4.1.f contains the orientation color key that is used throughout the paper for all orientation images shown. The final stage of this simulation (Fig. 4.1.e) indicates that the texture evolves to a dominant {111} orientation (blue color) associated with <111> // ND (normal direction) with permissible misorientation range of 15°.

Table 4.1 Experimental results of texture according to different film thickness in Cu films [2].

Case	Thickness	Texture
1	100nm-thick sputtered films	{111} fiber
2	200nm-thick sputtered films	{111} fiber +twins of {111}
3	500nm-thick sputtered films	{001} fiber, very large grain size
4	800nm-thick sputtered films	{001}+twins variants of {001}

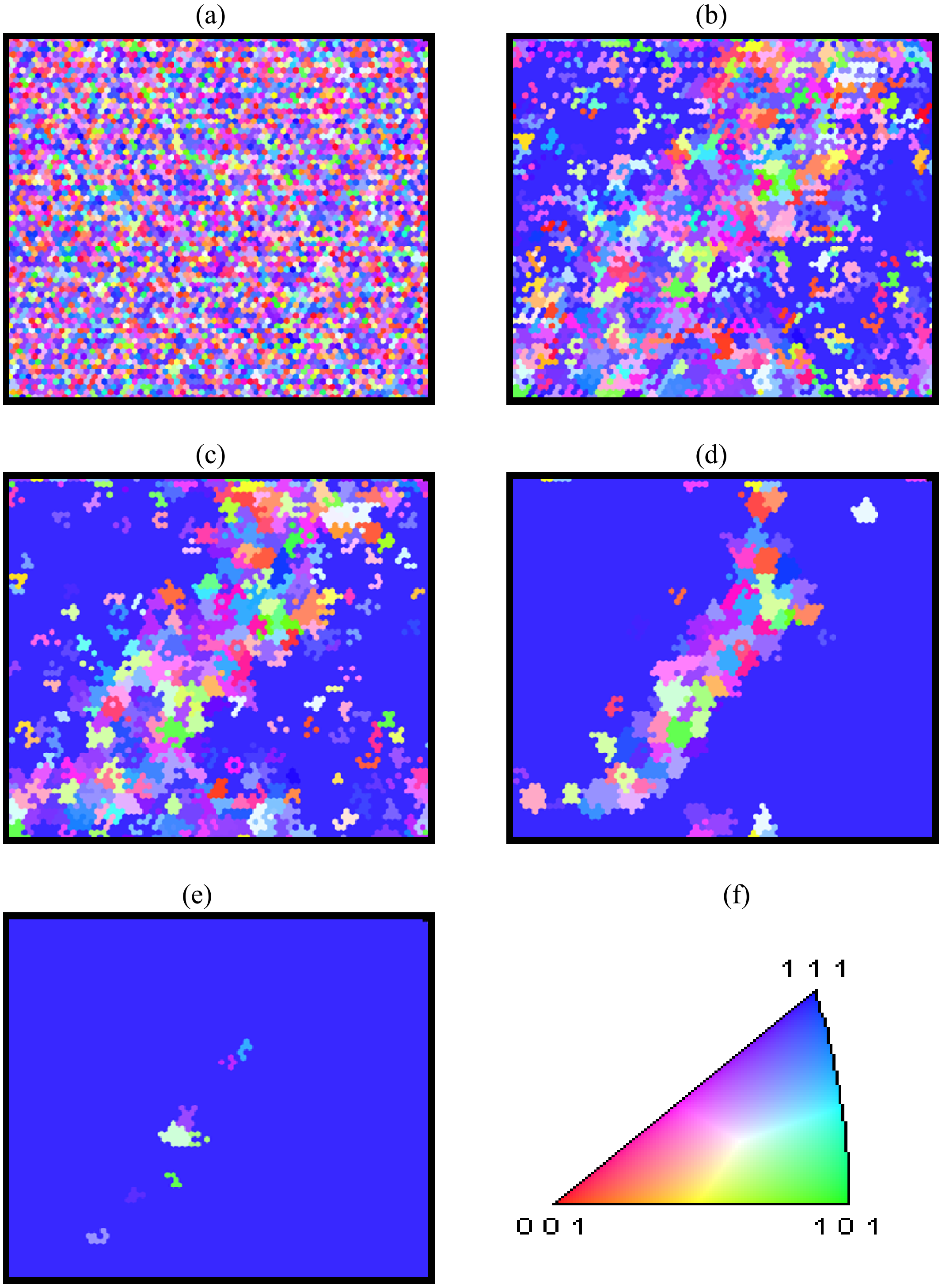


Figure 4.1 Orientation maps showing a story board presentation of simulated grain growth for 100 nm and 300°C (Figures a-e are in order of increasing MCS), f) shows the orientation color key.

Fig 4.2 and 4.3 show {111}, {101} and {001} texture component development respectively as a function of Monte Carlo Steps (MCS). These texture components are analyzed according to color gradients indicated by a degree of misorientation between 0° and 15° from the ideal component. Blue is {111}, green is {101} and red is {001}. The final stage shows that the texture is composed of almost entirely {111} orientations in a thin film of 100 nm thickness.

This agrees with well known experimental observation. This phenomenon has been studied by many researchers. For thin films, the {111} orientation is favored because of the surface energy minimization effect on the film [3, 4]. In general, thin Cu films have preferred {111} texture as a result of the close-packed surface that has the lowest surface free energy [5, 19].

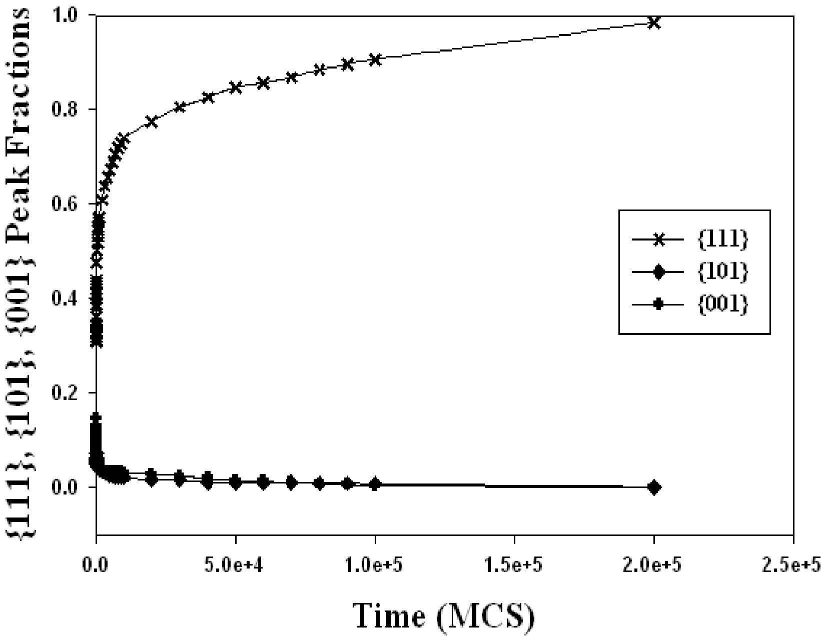


Figure 4.2 Predicted texture components as a function of Monte Carlo time steps for 100 nm and 300°

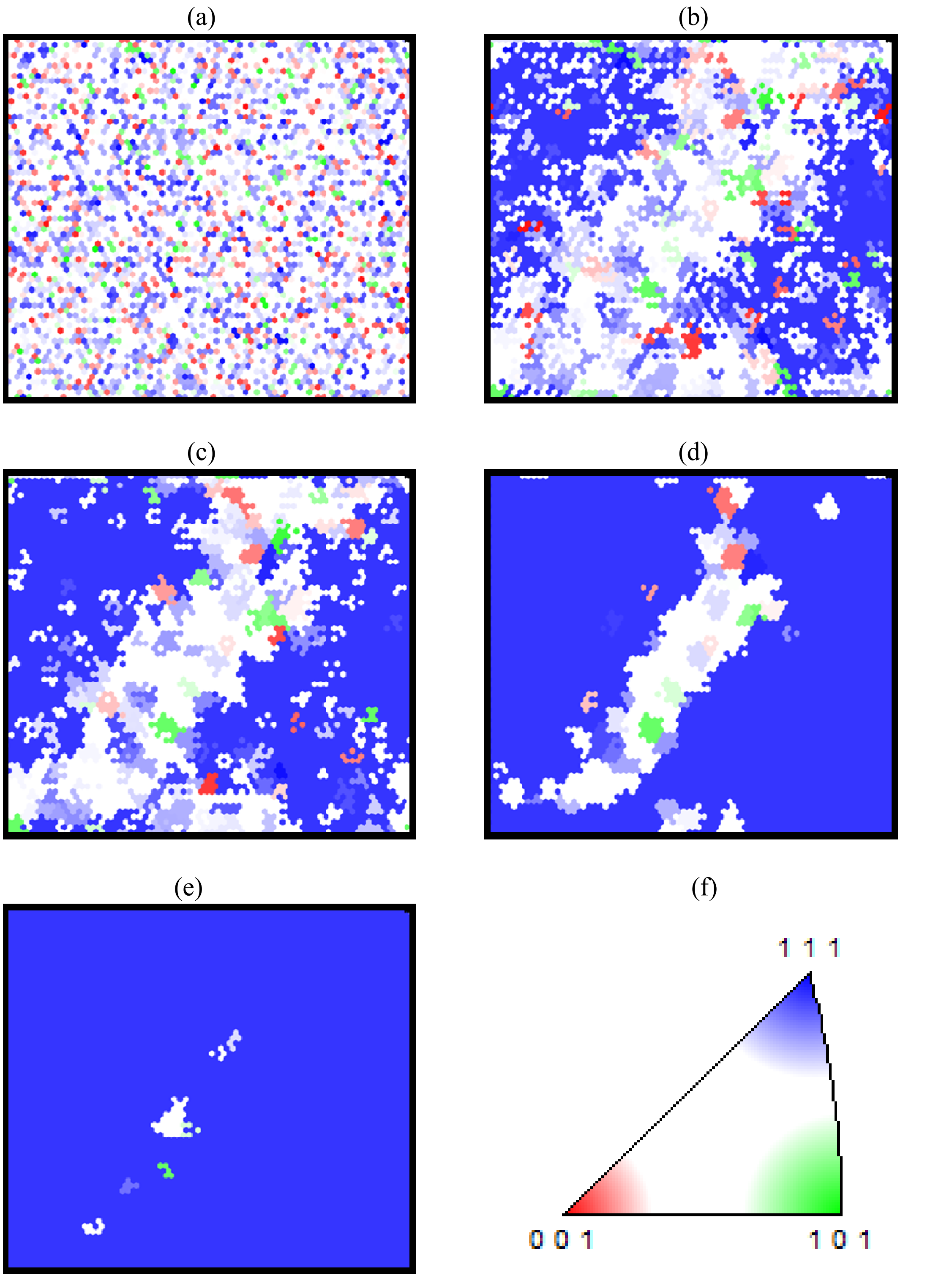


Figure 4.3 The same structures as shown in Fig. 4.2 with colors that identify {111}, {101} and{001} fractions for 100nm and 300°C

4.2 2nd case (500nm film thickness at 300°C annealing temperature)

The next simulation result, shown in Figure 4.4, was based on the conditions of 500 nm thick films with 300°C annealing temperature. Figure 4.4 indicates that when the {111} grains grow, the {511} twins of {111} grains start to appear as they assist in reducing the total system energy. These twins are about 16° away from {001} grains. These twins appear automatically in the evolving structure through grain boundary energy considerations. Even though surface and/or strain energy are not minimized by a {511} orientation, the total energy considerations tend to favor the twin component because of the favorable boundary energy between the {111} and {511} components. This result compares well with observations of structure evolution for 200 nm films (see Table 1).

In addition, Figure 4.5 and 4.6 indicate evolution of twin grains having orientation of {511}//ND with 27 percent of the boundaries considered to be Σ3 boundaries according to CSL theory. These twins were traced by the geometrical concept that the misorientation (Δg) of twin has the 60° rotation about the <111> crystal direction with energetic relations between surface energy and grain boundary energy at equation (14). Figure 4.5 shows the fraction changing of {001}, {101}, {111} and {511} grains for 500nm at 300°C annealing temperature and especially, Figure 4.6 indicates the changing of {511} twins marked by Σ3 of twins of {111} grains as a function of Monte Carlo steps (MCS).

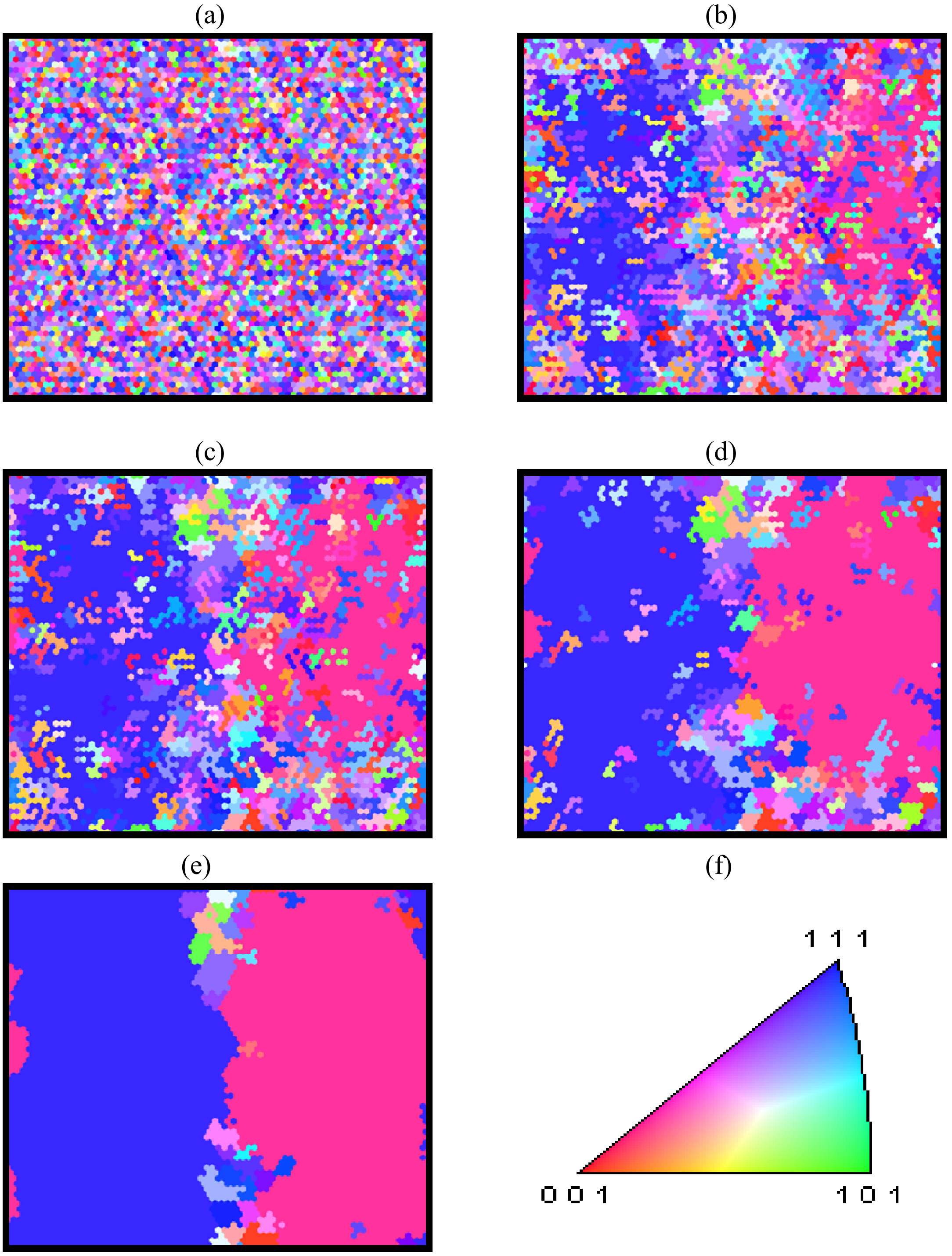


Figure 4.4 Orientation maps showing a story board presentation of simulated grain growth for 500 nm and 300°C (Figures a-e are in order of increasing MCS), f) shows the orientation color key.

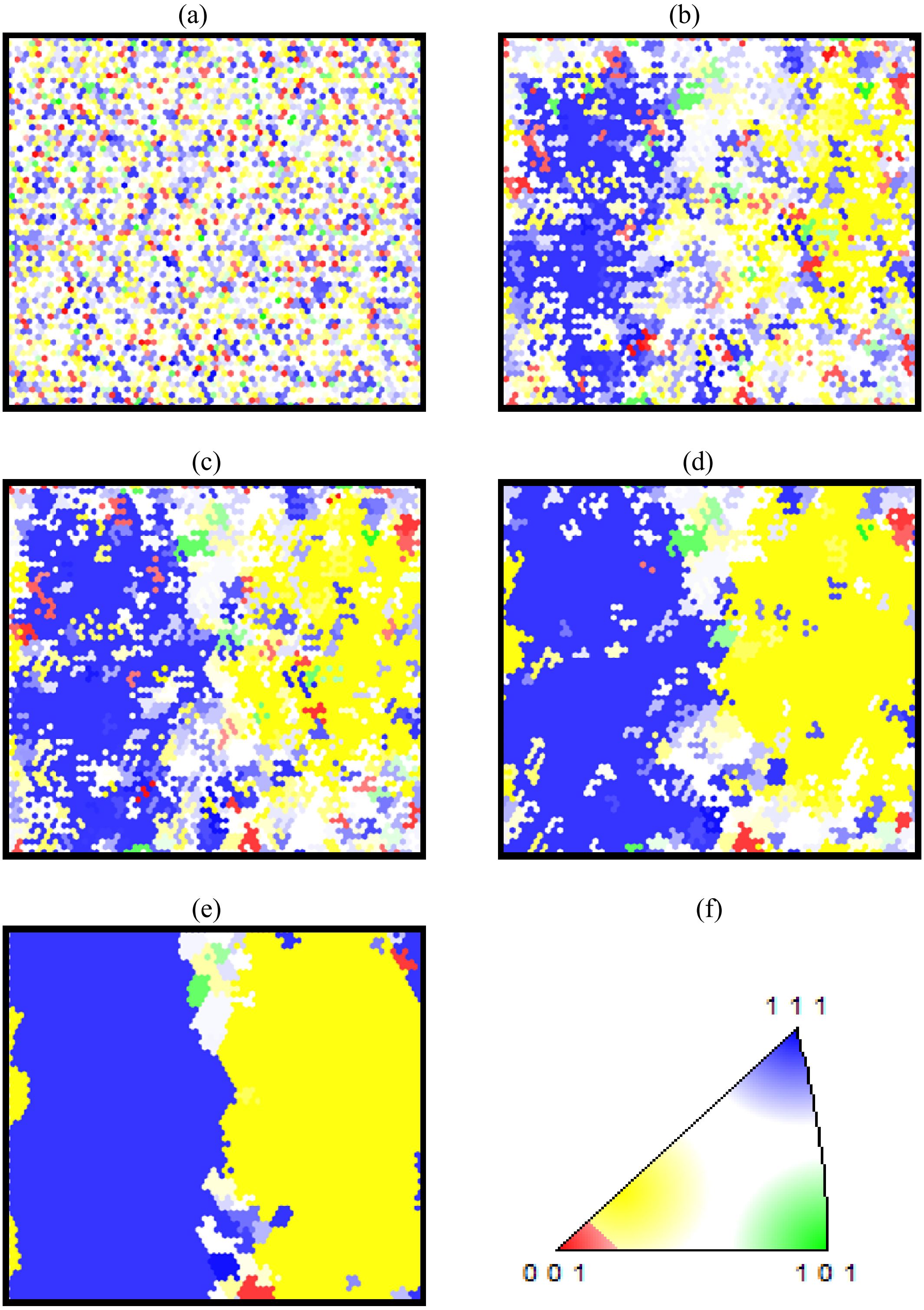
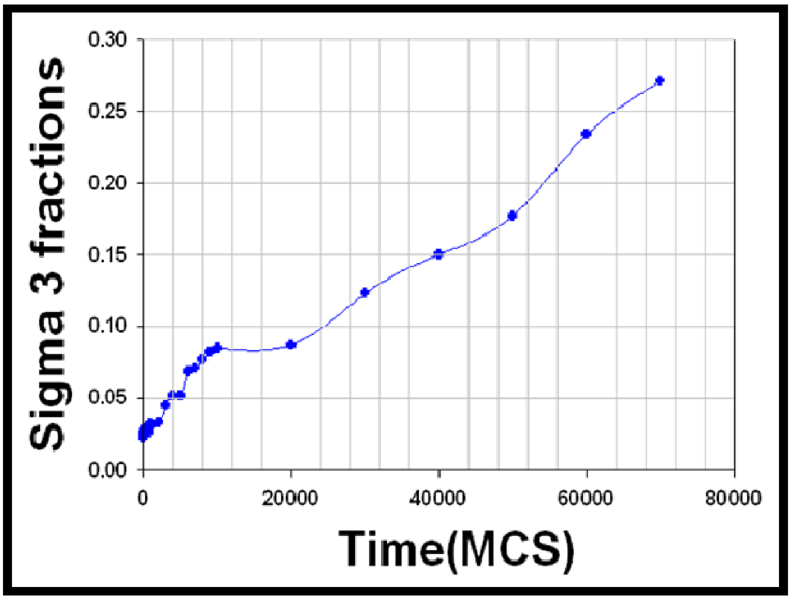


Figure 4.5 The same structures as shown in Fig. 4.4 with colors that identify {111}, {101} {001} and {511} fractions for 500nm and 300°C



MCS	Σ3 Fractions
1	0.022
300	0.029
900	0.031
9,000	0.085
70,000	0.271

Figure 4.6 The changing of {511} twins indicated by Σ3 of twins of {111} grains according to Monte Carlo steps (MCS).

4.3 3rd case (800nm film thickness at 300°C annealing temperature)

Simulation results for 800 nm thick films show a dominant {001} orientation at the final step of the MC simulation (Figure 4.7). The initial textures and grain assignments for the 100 nm and 800 nm simulations were similar but the final textures are distinct. The only difference in simulated conditions was film thickness. This 800 nm film thickness case of simulation indicates the preferred texture is {001}. Figure 4.8 and 4.9 indicate how the fractions of {111}, {101} and {001} evolved with MC time steps. Figure 4.8 and 4.9 show that the texture rapidly changed to {001}. {111} and {101} fractions quickly went to zero at the expense of the {001} grains. No twins of {100} grains were observed in the simulations as there is apparently no condition where the energy would be minimized by creation of such a structure. The results that preferred texture formation at thinner 100 nm films was {111} and thicker 800 nm films was {001} can be explained by lowering the energy of the system during processing.

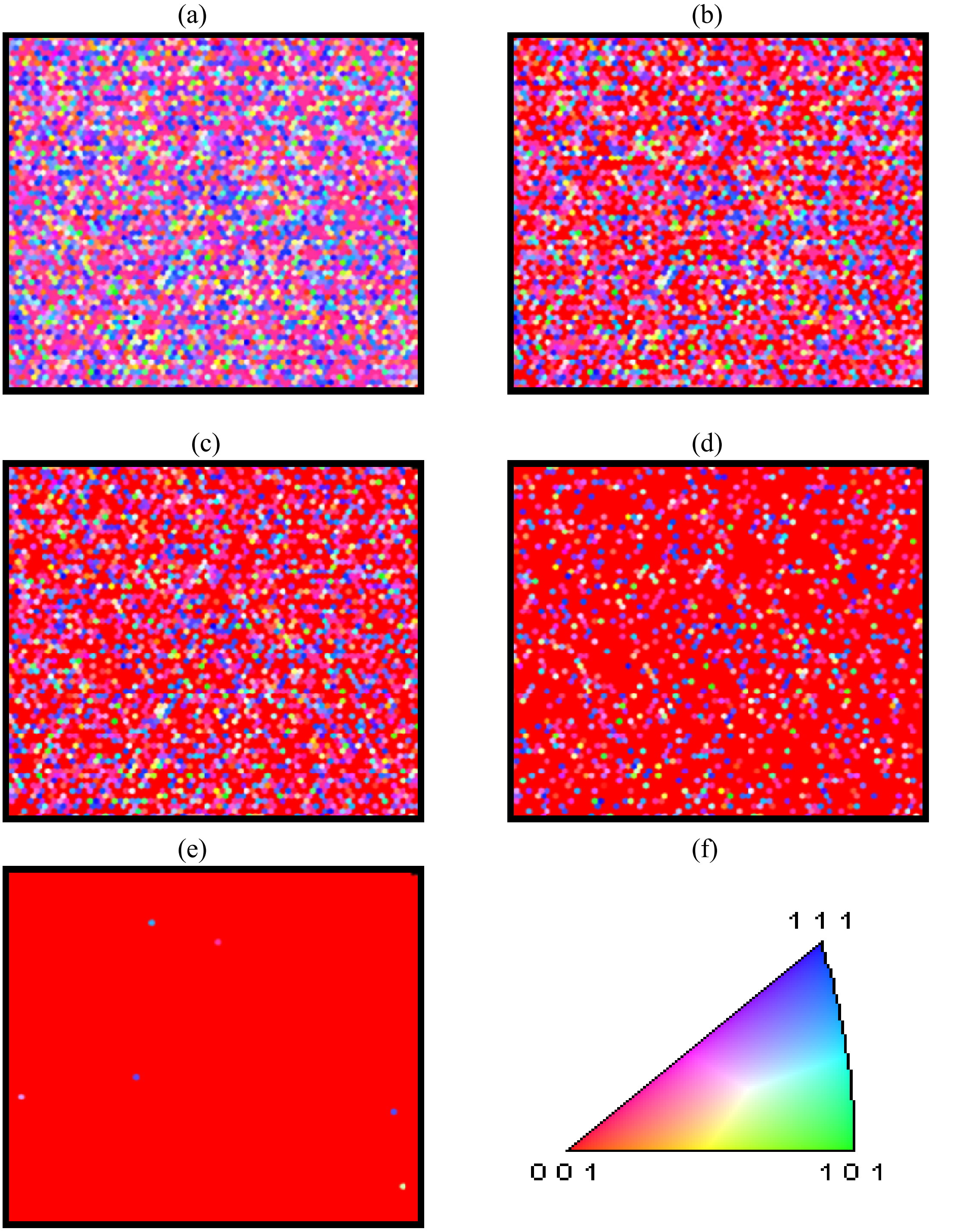


Figure 4.7 Orientation maps showing a story board presentation of simulated grain growth for 800 nm and 300°C (Figures a-e are in order of increasing MCS), f) shows the orientation color key.

Thompson and co-workers established the concept of energy competition between surface/interface and strain energy (as a function of film thickness and temperature) almost two decades ago [3, 4]. The determination of which grains will grow is a result of the competition between strain energy minimization and surface/interface energy minimization. When the change of the surface/interface energy is larger than the change of strain energy during grain growth, {111} out of plane texture dominates. Otherwise, {100} texture minimizes the strain energy density and dominates growth.

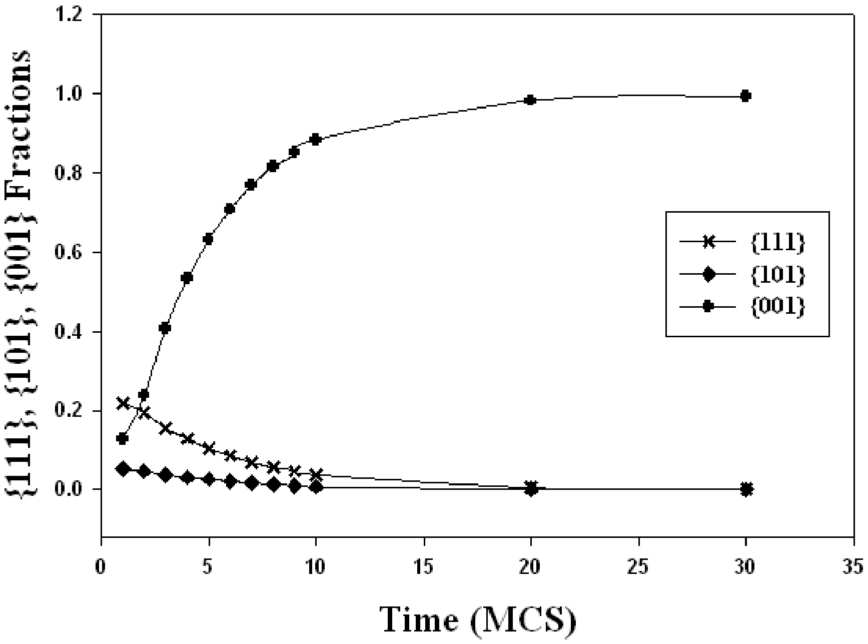


Figure 4.8 Predicted texture components as a function of Monte Carlo time steps for 800 nm and 300°C.

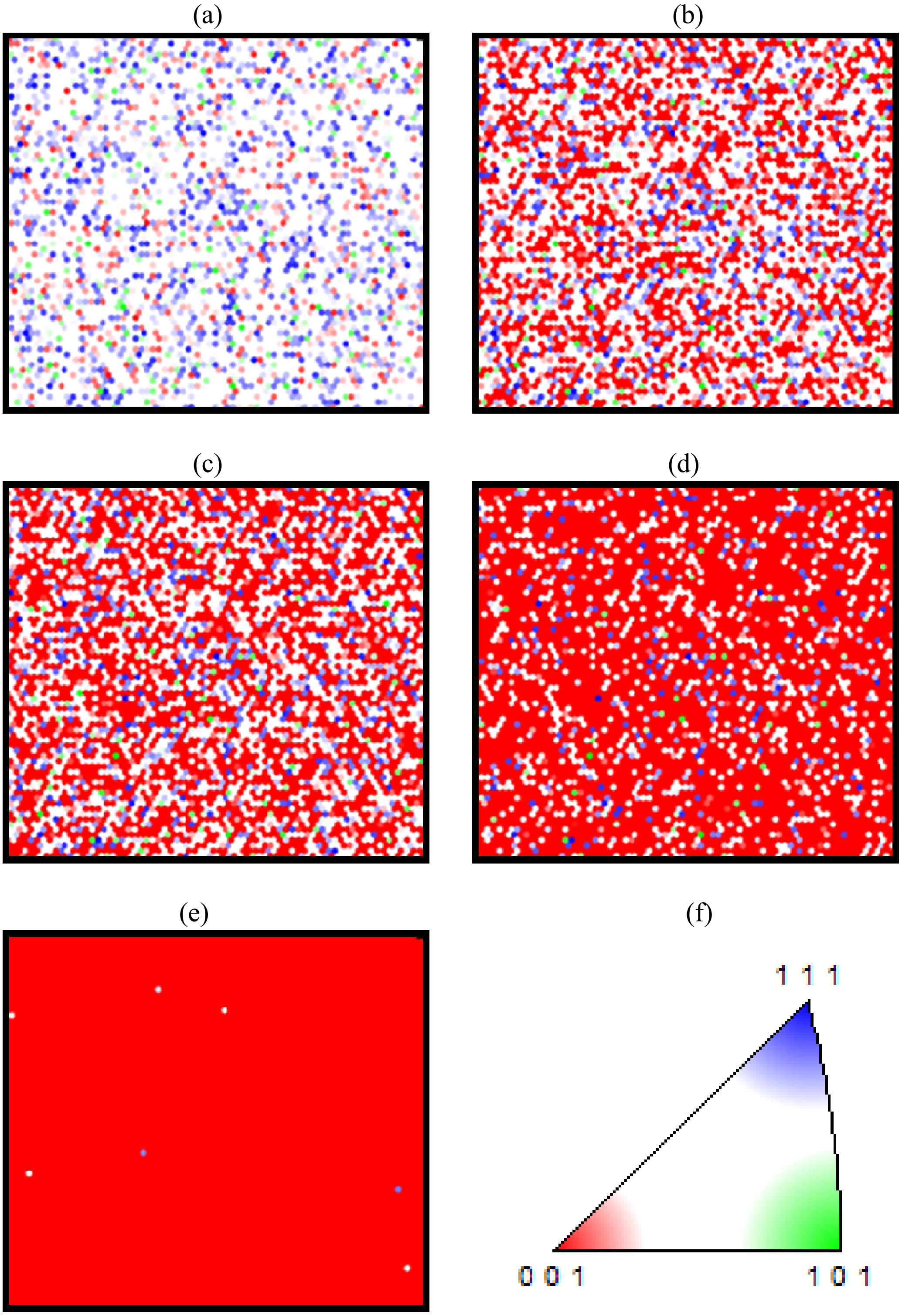
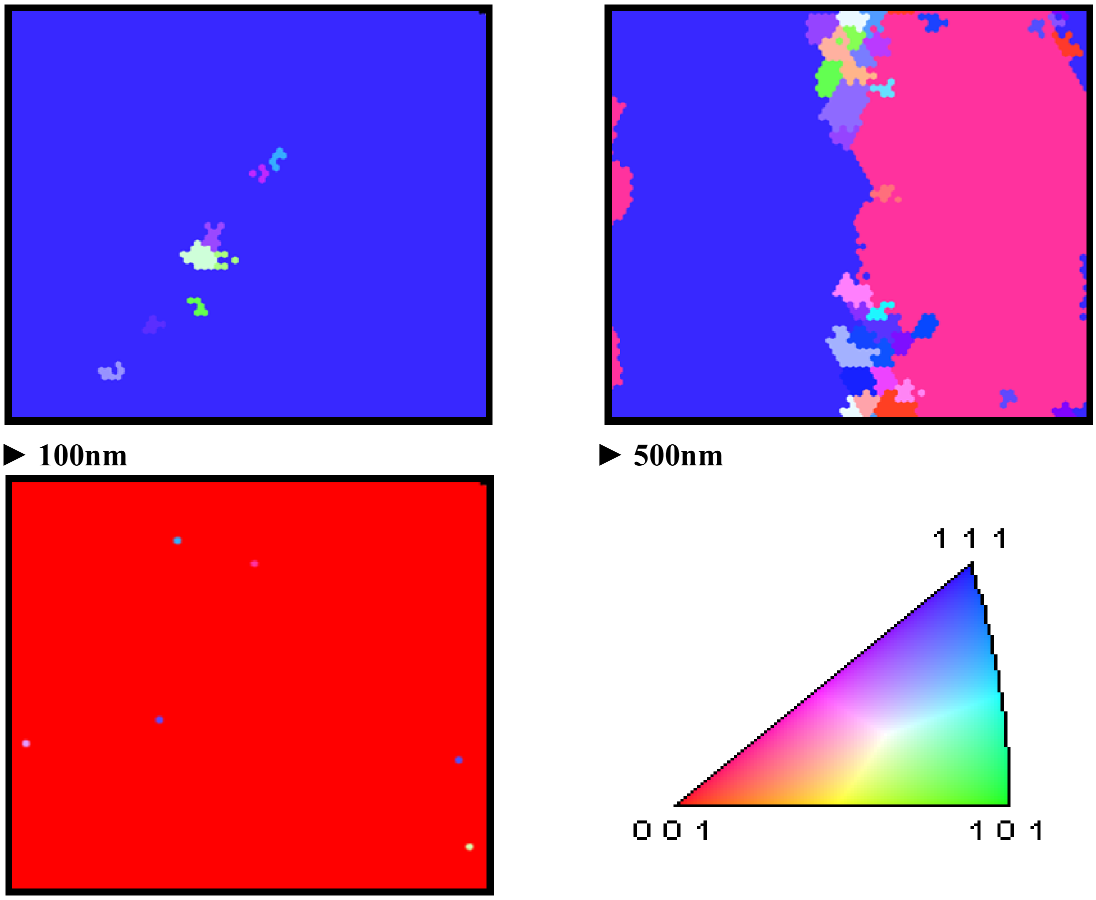


Figure 4.9 The same structures as shown in Fig. 4.7 with colors that identify {111}, {101} and{001} fractions for 800nm and 300°C

CHATER 5

DISCUSSION

This Monte Carlo simulation method with energetic considerations based on surface/interface, grain boundary, and strain energies showed a structure evolution in sputtered thin films as a function of film thickness. Figure 5.1 shows a summary of the final simulation results including the resulting fractions of {111}, {511} and {001} textures for films of 100 nm, 500 nm and 800 nm thickness. In 100 nm films, the experimental and simulation results show the preferred texture is {111} with surface/interface energy minimization dominating structure evolution [3, 4]. The 500 nm film thickness case of Figure 5.1 shows that {111} grains with twins of {511} tend to form a lower total system energy when all effects (surface energy, strain energy, and grain boundary energy) are considered. In thicker films such as the result shown for 800 nm, {001} grains grow rapidly during annealing because of strain energy minimization being the dominant factor in structure evolution [3, 4]. According to C. V. Thompson [3], surface energy dominates growth in the 100 nm films resulting in {111} texture. Strain energy dominates growth in the 800 nm films and {100} texture forms as seen in Figure 5.2 [2].



► **100nm**

► **500nm**

► **800nm**

Figure 5.1 Orientation maps showing a story board presentation of 100 nm, 500 nm and 800 nm at 300°C

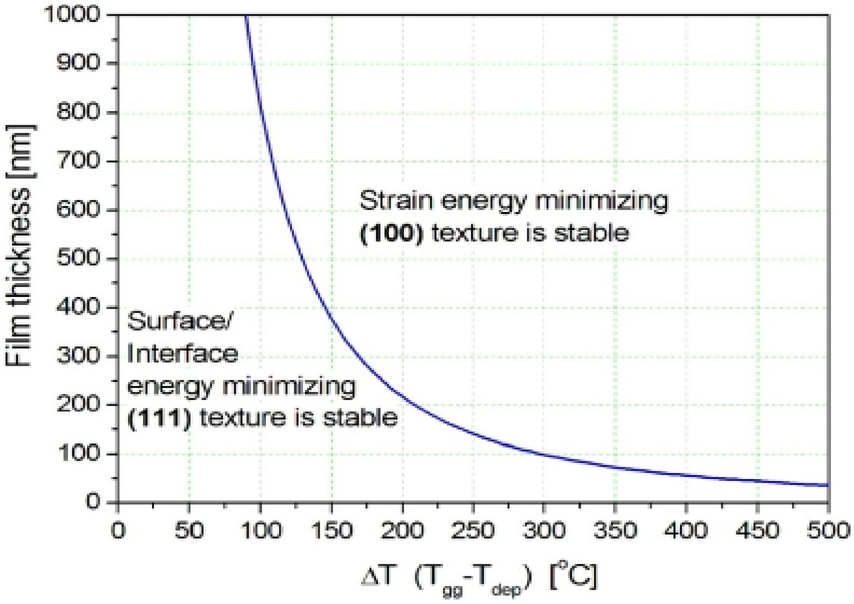


Figure 5.2 Texture map calculated with surface/interface and elastic strain energy for Cu-film resulting from grain growth at T_{gg} in Cu films of thickness h, deposited at T_{dep} [2].

There are no results that show a low energy configuration for the thicker films where the twins of {001} grains are present in the simulations. It is hypothesized that during the process of {111} grains growing, the {511} grains which are the twin grains of {111} start to be present for the purpose of reducing the total system energy. If a grain forms that is about 15º away from {111} and a twin develops in this structure, it would provide a nucleus for growth of {100} grains. This happens infrequently so the observed grains are very large (nuclei spaced far apart). In addition, experimental results show that {211} and other texture components that are twin related to either {111} or {001} textures exist in the thicker films (800 nm). The results obtained in this work show no such structure. In a previous paper [45], twin formation occurred by the brute force method. Twins grains were arbitrarily inserted into the structure at pre-selected conditions to achieve the desired structure evolution. While properly simulating evolved structures, this approach lacks any predictive capability.

A realistic model that is capable of accurately predicting structure evolution requires a mechanistic description of twin formation. Many researchers indicate formation of twins in terms of three factors such as grain boundary energy, grain boundary mobility and role of the dislocation arrangement. Specifically, according to growth accidents and nucleation of twins by stacking faults or fault packets, many researchers suggest a twin formation mechanism. Gleiter [46] in 1969 proposed a model of annealing twin formation as a result of growth accidents leading to stacking faults. Pande *et al*. [32] carried out a systematic study of annealing twin formation in nickel with the assumption that twin formation is caused by grain boundary migration and their density depends on the driving force for migration. Mahajan *et al*. [29] suggested that Shockley partial dislocation loops nucleate on consecutive {111} planes by growth accident related to grain boundary migration. In addition, previous research [13-18, 33] has shown experimental evidence of various twinning fractions caused by different kinds of conditions such as film thickness, stress state, annealing conditions and fabrication processes. These factors would have to be included in the models to predict formation and evolution of twins in thin films.

The experimental results presented above are for high purity sputtered Cu films. It is well known that electro- deposition of Cu films results in a wide variety of structures subsequent to post-deposition annealing. These can range from randomly oriented, twin-rich structures to highly oriented {111} fiber textures even for films on the order of 1 μm thickness. The effects of bath chemistry on the energetics of the system and the boundary mobility would have to be included in any models of structure evolution for plated films. This will increase the complexity of the models dramatically. As mentioned previously, there are several factors that lead to varied mechanisms of structure evolution in thin films, whether sputtered or plated. Different factors such as the film thickness, the substrate materials (and stacking sequence), stress state, film deposition parameters, plating bath chemistry, annealing temperature and time, and geometrical constraints would all have to be considered to accurately predict structure evolution.

CHAPTER 6

CONCLUSION

Structure evolution in thin Cu films, including twin formation and texture evolution, were simulated by a Monte Carlo Potts model based on surface/interface, grain boundary and strain energy factors. Three different film thicknesses were simulated and result in either {111} or {001} out of plane texture depending on the energy minimization competition between surface/interface and strain energy. For the 100 nm films, the process of grain growth proceeds as observed experimentally with {111} grains growing first and gradually consuming other types of orientations. For 500 nm films, a {511} twin component was observed that tended to minimize the total system energy when grain boundaries were considered as lower energy coherent twins. For thicker films at 800 nm, the {001} grains grow rapidly during annealing due to the strain energy minimization criterion. Monte Carlo Potts modeling which includes energy minimization concepts and considers twin grain evolution can reasonably predict evolution of grain structures and crystallographic textures in thin films, but the present model fails to accurately capture effects of twin grain nucleation and growth. Specific criteria for introducing twin grains into the structure must be developed in order to obtain the best results for thicker films where surface energy no longer dominates structure evolution.

CHAPTER 7

FUTURE WORK

Microstructure evolution is not truly a two dimensional process in real thin films. The best approach to model real microstructure evolution requires a three dimensional simulation circumstance. Future work should adopt a three dimension framework. In addition:

1. This simulation predicts the preferred {111} grains at 100nm and {001} grains at 800nm cases with reasonable accuracy but there is no grain size index. Experimentally, {001} has a larger grain than {111} grain. However, there is no standard index to compare grain sizes with cases. The grain size limitation function should be included in this model.

2. Simulations of twinning behavior are not energetic or mechanistic in the present model. From twin's nucleation to growth, specific criteria for introducing twin grains into the structure and mechanism need to be established in the model.

3. In sputtered films, Monte Carlo simulations predicted structure evolution with reasonable accuracy but in plated films, this simulation is inaccurate. An understanding of electrochemistry and the effect of trace amounts of organic molecules on grain boundary mobility is required to improve the predictive capability of this model.

There are many factors that to affect structure evolution such as film thickness, substrate materials, stress state, film deposition parameters, plating bath chemistry, annealing temperature and time, and geometrical constraints. These would all have to be considered to accurately predict structure evolution.

REFERENCES

[1]. C. V. Thompson, R. Carel, J. Mech. Phys. Solids, 44, 657 (1996)

[2]. N. J. Park, D.P. Field. Scripta Materialia 54, 999 (2006)

[3]. C. V. Thompson, Annu. Rev. Mater. Sci. 20, 245 (1990)

[4]. C. V. Thompson, R. Carel, Mater. Sci. Eng. B32, 211 (1995)

[5]. M.T. Perez-Prado and J.J. Vlassak, Scripta Materialia 47, 817 (2002)

[6]. A. D. Rollett and P. Manohar " Chapter 4 of The Monte Carlo Method", Wiley-VCH, New York, 2004,

[7]. A. D. Rollet,. Progress in Mat. Sci. 42, 79 (1997)

[8]. G. N. Hassold and E. A. Holm, Computers in Physics, 7, 97 (1993)

[9]. E. A. Holm, C.C. Battaile JOM; 53, 9 (2001)

[10]. Y. Saito and M. Enomoto, ISIJ international, 32, 267 (1992)

[11]. M. Winning, Scripta Mater. 54, 987 (2006)

[12]. M. Winning, Phys.stat.sol. (a) 201, 2867 (2004)

[13]. D. P. Field, R. C. Eames, T.M. Lillo, Scripta Materialia 54, 2006 pp.983-986.

[14]. E. E. Danaf, S. R. Kalidindi, R. D. Doherty, Metall. Mater. Trans. 30A, 1223 (1999)

[15]. A. Rohatgi, K. S. Vecchio, G.T. GrayⅢ, Metall. Mater. Trans. 32A, 135 (2001)

[16]. K. Vanstreels, S. H. Brongersma, Zs. Tokei, L. Carbonell, W. De Ceuninck, J. D'Haen, M. D'Olieslaeger, J. Mater. Res, 23, 642 (2008)

[17]. B. B. Rath, M. A. Imam, C. S. Pande, Mater. Phys. Mech. 1, 61 (2000)

[18]. N. J. Park, D. P. Field, M. M. Nowell and P. R. Besser. J. Elect. Matls. 34, 1500 (2005)

[19]. D. Walther, M.E. Gross, K. Evans-Lutterodt , W.L. Brown, M. Oh, S. Merchant, P. Naresh In: Mater Res Soc Symp Proc 612, D10.1 (2000)

[20]. F. J. Humphreys and M. Hatherly, Recrystallization and related annealing phenomena, 2nd edition,Pergamon, NY, (2004) pp.218-220, pp 261- 267, pp 334-378, pp. 509-514

[21]. J. K. Stanley, R. F. Mehl, Trans. Metall. Soc. AIME 161, (1942) 140.

[22]. W. A. Anderson, R. F. Mehl, Trans Metall. Soc. AIME 135, (1945) 416.

[23]. R. F. Mehl, ASM Metals Handnook, ASM, Metals Park, Ohio, 259.

[24]. J. E. Burke, D. Turnbull, Prog. Met. Phys., 3, (1952) 220.

[25]. J. W. Christian, The Theory of Transformations in Metals and Alloys, Pergamon, 2002 pp.452, pp854-856

[26]. M. Ohring, Materials Science of Thin Films, 2nd edition, Academic Press, 2002 pp 378-383, pp 521-535

[27]. B. Stockhert and J. Duyster, Journal of Structural Geology 21, 1477-1490 (1999)

[28]. J. M. E. Harper, C. Cabral, P. C. Andricacos, L. Gignac. I. C. Noyan, K. P. Rodbell, and C. K. Hu, Journal of Applied Physics.86, 2516(1999).

[29]. S. Mahajan, C. D. Pande, M. A. Imam, and B. B. Rath, Acta Mater. 45, 2663 (1997)

[30]. S. Dash, N. Brown, Acta Metall. 11, 1067 (1963)

[31]. M. Meyers, L. E. Murr, Acta Metall. 26, 951(1978)

[32]. C. S. Pande, M.A. Imam and B.B. Rath, Met. Trans. A, 21A,2891 (1990)

[33]. D. P. Field, L. T. Bradford, M. M. Nowell, T. M. Lillo, Acta materialia 55, 4233(2007)

[34]. R. L. Fullman, J. C. Fisher, Journal of Applied Physics. 22, 1350(1951)

[35]. LE Murr, Interfacial phenomena in metals and alloys. London: Addison-Wesley;1975

[36]. V. Randle and O. Engler, Introduction to Texture Analysis Macrotexture, Microtexture and Orientation Mapping: Taylor and Francis; 2000 pp25-40, pp 248- 262

[37]. S. I. Wright and R. J. Larsen, Journal of Microscopy, 205, 245 (2002)

[38]. A. Morawiec, Orientaions and Rotations Computations in Crystallographic Textures. New York: Springer;2004

[39]. B.E. Sundquist Acta Metall. 12, 67 (1964)

[40]. Mclean M, Acta Metall, 19, 387 (1971)

[41]. WC Overton and J. Gaffney, Phys Rev. 98, 969 (1955)

[42]. LE Murr, Interfacial phenomena in metals and alloys. London: Addison-Wesley;1975

[43]. T. Read and W. Shockley, Phys. Rev. 78, 275 (1950)

[44]. R.W. Balluffi, S. M. Allen, W. C. Carter, Kinetics of Materials, Wiley-Interscience, New York (2005) p. 342

[45]. C.J. Chung, D.P. Field, N.J. Park, R.G. Johnson, Thin Solid Films, 517, 1977 (2009)

[46]. Gleiter H. Acta Metall 17, 1421 (1969)

APPENDICES

These appendices primarily consist of programming codes modified by FORTRAN that have been changed from the previous E. A. Holm (1992) and A. D. Rollet (1995) Monte Carlo codes.

APPENDIX A – The main subroutine controls all other subroutines. Especially, MCGROWTH1, MCPROBS1 and TWFIND subroutines were modified to simulate microstructure evolution with annealing twins according to energetic considerations

MAIN.F

```
c       FINITE   :   Main program.
c       ----------------------------------------------------------
c       1. copyright 1992 Elizabeth A. Holm
c       2. modified by a.d.rollett, oct, nov 1995
c        3. further modified (minor stuff) summer 03 to deal with
c           Fe-3%Si microstructures from MPIE
c        4. further modifed (minor stuff) by ChangKyu Yoon to focus on
c           microstructure evolution with annealing twins considering energetic, 2008


c       Simulates two-dimensional recrystallization and grain growth via
c       Monte Carlo and n-fold way Potts model algorithms, combined to
c       minimize cpu time. Growth occurs at any temperature on a tri(1,2) lattice.
c
c       To choose simulation lattice type, uncomment the appropriate
c       lines in the common file common.f and in the prep subroutine.
c
c       Inert, stationary, second-phase particles may be incorporated
c       into the matrix.   The particles are whiskers one site wide with
c       length and volume fraction input by the user.
c
        program rxgrgr2d
        include 'common.f'
        integer way,site,expon,count,flips,time,iseed
c       character*1 one,ten
        character*1 one,ten,hundred
        integer inccs(mn),nnrexgns
        real volf
c       nnrexgns==no of rex grains;   volf==vol frac rex
        real amaxenrg,amaxmob,eavg,mavg,misavg
        integer iamax,iamaxngb,ngb
```

```
c     IAMAX,AMAXENERG,AMAXMOB & IAMAXNGB are the area,Energy,Mobility and gb
sites for the largest grain
c     EAVG, MAVG are the <E> and <M>; NGB is a av. no. of g.b. perim. sites
c     MISAVG is the av. misorientation
        real cubefrac
c     CUBEFRAC == fraction of area within a certain distance of cube1
        integer kcount,klimit
c     KCOUNT keeps track of iterations through outer loop
c     KLIMIT    sets the max no of iterations
        real tmpa(-1:q)
c     TMPA is used for swapping arrays
        integer type(-1:q)
c     TYPE keeps track of the type of orientation; here 0==general, 1==cube
        integer ngstop
c     NGSTOP used to calculate number of grains at end of each simulation
        character*1 decimal,hun,thou,tenthou
c     for use with KCOUNT
        character*6 altkey,swpkey

        logical time_to_stop,proceed
        real pi,pi2
        real ran1
        real oldcubefrac,tmp,fdiff,pfdiff
        integer i,iswap1,iswap2,itmp,pltw
c       real PB
c       read user-determined run parameters:
c       'keyword' identifies the run (six characters or less)
c       'tstat' is number of Monte Carlo Steps before first measurement
c       (double precision real)
c       'tmax' is the maximum number of Monte Carlo Steps (double precision real)
c       'temp' is the temperature (real)
c       'length' is the length of linear second phase particles (integer)
c       'linevol' is the volume fraction of second phase particles (real)
c       'nucvol' is the volume fraction of recrystallized grain nuclei (real)
c       'trials' is the number of independent trials to be performed (integer)
c       'mobratio' is the ratio of mobility of a special boundary class
c       which can be greater or lesser than unity; if mobratio>1,
c       we must rescale the general mobility
c       'eratio' is the ratio of boundary energy of a special class of boundary
c       if eratio>1, we must rescale the general energy level
c       mobility affects the transition probability only, whereas
c       the energy ratio affects the delta(E) also
c     'rtime0' is the radius of the special grain at time=0
c       'ISTOP' is the Q value above which the central (S=1) grain has special properties
c       'STORED is the value of the stored energy
```

```
      pi=3.1415926536
      pi2=pi/2.
      read(*,*) keyword
      altkey=keyword
      read(*,*) tstat
      read(*,*) tmax
      print*,'TSTAT= ',tstat
      print*,'TMAX= ',tmax
      read(*,*) temp
      read(*,*) length
      read(*,*) linevol
      read(*,*) nucvol
      read(*,*) trials
      read(*,*) mobratio
      read(*,*) eratio
      read(*,*) rtime0
      read(*,*) istop
      read(*,*) stored


      write(*,*) 'temp = ',temp
      write(*,*) 'length = ',length
      write(*,*) 'linevol = ',linevol
      write(*,*) 'nucvol = ',nucvol
      write(*,*) 'trials = ',trials
      write(*,*) 'mobratio = ',mobratio
      write(*,*) 'eratio = ',eratio
      write(*,*) 'rtime0 = ',rtime0
      write(*,*) 'istop = ',istop
      write(*,*) 'stored = ',stored


      if(istop.lt.(q2+1)) stop 'error: ISTOP too small'
c     if(istop.gt.q) stop 'error: ISTOP>Q! '
      do i=q2+1,q
         if(i .le. istop) then
            type(i)=1
         else
            type(i)=0
         end if
      end do
c  sets up the TYPE array


      if(mobratio .gt. 1.0) then
            bulkratio=1./mobratio
```

```
                    mobratio=1.
          else
                    bulkratio=1.
          end if

          if(eratio .gt. 1.0) then
                    bulke=1./eratio
                    eratio=1.
          else
                    bulke=1.
          end if

          if(temp .ge. tcrit) then
                    write(*,*) 'warning:    temperature greater than critical'
                    write(*,*) 'system will disorder!'
          end if

c         initialize the random number generator using the UNIX system
c         time as a seed

c         iseed=time()
          kcount=1
          klimit=1      !    set this to a large number to allow many iterations
          iswap1=0
          iswap2=0
          oldcubefrac=0.0
c   !!!!!!!!!!!!!!    4000 series labels denote the simulated anneal edits
c   return to 4010 to loop back
4010              iseed=secnds(0.0)
           iisave=-1*iseed
c    edited for Dec alpha, iii96    adr
c         call srand(iseed)
          tmp=ran1(iisave)
c         call ran1(iisave)
c         different initialization for RAN1 - see MISC.f
c
          do 123, i=1,6
          write(*,*) 'testing random numbers:',ran1(iisave)
123       continue
c



c         the main loop of independent trials
c
          trial=1
```

```
c   run with only ONE trial in this version
c
c         do 900 trial=1,trials
c
c         initialize variables:
c         't' is the simulation time in Monte Carlo Steps (MCS)
c         'tmeas' is the time at which next set of measurements will be taken
c         'rbar' is the average grain radius
c         'way' indicates which algorithm (i.e. classical vs. n-fold way)
c         is currently being used
c         'count' and 'expon' identify the time at which a photo was taken
c         (time = count*10^expon)
c         'trlno' identifies the trial in progress
c         'flips' counts n-fold way steps between probability table resortings
c
          t=0.d0
          tmeas=tstat
          rbar=1.
          way=0
          pltw=0
          expon=int(log10(tmeas))
          count=int(int(tmeas)/10.**expon)
          trlno=char(48+trial)
          flips=0
c
c    start new file for size statistics in file data.keyword.trlno
c
          if(kcount .le. 1) then
                   open(21,file='fraction.cube')
                   open(22,file=keyword//'.amax.hist')
c    this open used to be inside size.f, but now needs to be in main, and once only!
c
c
c            open(10,file=keyword//'.'//'data.'//trlno,recl=132)
             open(10,file=keyword//'.'//trlno//'.data')
                          write(21,*) "      time(MCS) rbar abar stdeva",
c        &                        " grain activ no_rex_grains, VfRex, CubeFrac"
       &                 " grain activ iamax eavg mavg ngb Misavg CubeFrac"
                          write(10,*) "      time(MCS) rbar abar stdeva",
c        &                        " grain activ no_rex_grains, VfRex, CubeFrac"
       &                 " grain activ iamax eavg mavg ngb Misavg CubeFrac"
                          write(*,*) "      time(MCS) rbar abar stdeva",
c        &                        " grain activ no_rex_grains, VfRex, CubeFrac"
       &                 " grain activ iamax eavg mavg ngb Misavg CubeFrac"
c                        close(10)
c                        close(21)
```

```
        else
c                 open(10,file=keyword//'.data.'//trlno,access='append')
                  open(10,file=keyword//'.'//trlno//'.data.',access='append')
                  write(10,*)
                  write(10,*) 'Iteration ',kcount
                  write(10,*) 'old, new cube frac. ',oldcubefrac,cubefrac
                  close(10)
        end if
c   only need to open the data file once, assuming that we want all iterations
c
c       prepare a list of the neighbors of each lattice site,
c       initialize the time increment for the classical growth routine,
c       initialize the grain structure, place second phase particles
c       in the matrix, and nucleate recrystallized grains
c
        if(kcount .le. 1) then
                  call prep
        else
                  call init
        end if
c   only if we are in the first time through, call PREP
c   otherwise, call INIT to reset the grain structure!
c
                  if(kcount .ge. 0 .and. kcount .lt. 10) then
                            decimal=char(48+kcount)
                            altkey='1aaaa'//decimal
                  elseif(kcount.ge.10.and.kcount.lt.100) then
                            decimal=char(48+mod(kcount,10))
                            hun=char(48+(kcount/10))
                            altkey='1aaa'//hun//decimal
                  elseif(kcount.ge.100.and.kcount.lt.1000) then
                            decimal=char(48+(mod(kcount,10)))
                            hun=char(48+(mod((kcount/10),10)))
                            thou=char(48+(kcount/100))
                            altkey='1aa'//thou//hun//decimal
                  elseif(kcount.ge.1000.and.kcount.lt.10000) then
                            decimal=char(48+(mod(kcount,10)))
                            hun=char(48+(mod((kcount/10),10)))
                            thou=char(48+(mod((kcount/100),10)))
                            tenthou=char(48+(kcount/1000))
                            altkey='1'//tenthou//thou//hun//decimal
                  elseif(kcount.ge.10000) then
                            stop 'too high an iteration count!'
                  endif
c                 write(*,*) 'made up file name ',altkey
                  swpkey=keyword
```

```
            keyword=altkey
c           call tex_out(one,ten,cubefrac)
c      print*,'MAIN: TMAX= ',tmax
c           call micro_out(one,ten)
            keyword=swpkey
c   all this to capture the structure at the beginning of the simulation
c
c      if the grain size indicates that the n-fold way would be more
c      efficient, change the growth type indicator 'way'
c      and initialize the n-fold way lists
c
c           if(temp .gt. 0.) then
c        write(*,*) 'before QUENCH'
c                 call quench
c           end if
       write(*,*) 'before LISTS'
            call lists
       write(*,*) 'before INDEXX'
            call indexx(mn,pis,table,index)
         write(*,*) 'before BINNING'
            call binning
       write(*,*) 'before SIZE'
            call size(nnrexgns,volf,iamax,amaxenrg,amaxmob,iamaxngb)
c
            if((way.eq.0).and.(rbar.ge.crossr)) then
                  way=1
                  write(*,*) 'rbar=',rbar
            end if
        write(*,*) 'Using [0=classical, 1=n-fold] way=', way
c
c      perform recrystallization and growth:
c
c      for growth steps - at early times, use the classical
c      Monte Carlo algorithm 'mcgrowth'; at late times, use the n-fold
c      way growth algorithm 'ngrowth'
c
c      note:   nucleation of recrystallized grains is performed all at
c      once before growth commences in the 'nucleate' subroutine
c
c      statement 900 loops back to statement 10
c
10     continue
c
c      write(*,*) 'after 10; flips=',flips,'    time= ',t,' tmax= ',tmax
       if(way.eq.0) then
            call mcgrowth1
```

```
      else
      if(pltw.eq.0) then
            call twfind
      pltw=1
      go to 100
      else
            call ngrowth
            flips=flips+1
      end if

      end if

c
c     increment simulation time
c
      t=t+delt
c

c     if enough n-fold way flips have been performed,
c     resort the n-fold way probability tables
c
c     write(*,*) 'value of flips:',flips,'    value of mn:',mn
      if(flips.ge.(mn/10)) then
        write(*,*) 'before INDEXX/flips'
            call indexx(mn,pis,table,index)
        write(*,*) 'before BINNING/flips'
            call binning
            flips=0
      end if
c
c     if it is time, measure
c     the # of grains 'grain'
c     the mean grain area 'abar'
c     the mean grain radius 'rbar'
c     the standard deviation of the grain area 'stdeva'
c     the the total system activity 'activ'
c     the grain size distribution 'areas(size)'
c     and topological distribution 'nngrns(neighbors)'
c
c     output files are:
c     data.keyword.trlno - grain growth data as a function of time
c           for trial trlno
c     area.xy.keyword.trlno - grain size distribution at time
c           t=x*10^y for trial trlno
c     nns.xy.keyword.trlno - distribution of the number of neighbor
c           grains at time t=x*10^y for trial trlno
```

```
c          ph.xy.keyword - a dump (photo) of the grain structure at time
c                    t=x*10^y, taken only in the first trial
c          info.keyword - a description of the simulation, written from
c                    subroutine prep
c    micro_out    writes out *.ang files which can be analyzed in
c    the standard TSL OIM software to investigate g.b. character
c    tex_out writes out *.wts files for characterizing texture
c    with the popLA package
c
c          in the first trial, measurements and photos are taken in
c          logrithmic increments - in later trials, size measurements
c          are taken in logrithmic increments, and no photos are taken
c
c          if the temp>0, the system is quenched to eliminate singlet grains
c          before measurements are taken; the original system is restored
c          after measurements are done
c
c          write(*,*) 'value of t:',t,'    value of tmeas:',tmeas,'    value of temp:',temp
           if(t.ge.tmeas) then
                    if(temp.gt.0.) then
             write(*,*) 'before SIZE, 1st call'
                             call size(nnrexgns,volf,iamax,amaxenrg,amaxmob,iamaxngb)

             write(*,*) 'before quench/size'
                             call quench
             write(*,*) 'before LISTS/size'
                             call lists
             write(*,*) 'before INDEXX/size'
                             call indexx(mn,pis,table,index)
             write(*,*) 'before BINNING/size'
                             call binning
                    end if
             write(*,*) 'before SIZE, 2nd call'
                    call size(nnrexgns,volf,iamax,amaxenrg,amaxmob,iamaxngb)

                    call gbanalyse(eavg,mavg,ngb,misavg)
c     gets the average energy and mobility of the grain boundaries
c          find the current time as count*10^expon and convert 'count'
c          and 'expon' to character variables 'ten' and 'one' respectively
c
                    expon=int(log10(t))
                    count=int(t/10.**expon)
                    ten=char(48+count)
c                   one=char(48+expon)
                    one=char(48+mod(expon,10))
                    hundred=char(48+(expon/10))
```

```
c   this assumes that we do not go over 99 in time exponent!
c
c         append size statistics to file data.keyword.trlno
c
            write(*,*) 'before TEX_OUT '
                    call tex_out(one,ten,hundred,cubefrac)
c   1 iv 01: moved tex_out here to obtain the cubefraction
c
c                   call exch_out(one,ten,hundred)
c   xii 01; write out the exchange info to a file
c
                    open(10,file=keyword//'.'//trlno//'.data.',access='append')

                    write(10,7001) t,rbar,abar
       &                 ,stdeva,grain,activ,iamax,eavg,mavg,ngb,misavg,cubefrac
 7001                    format(4(1x,g9.3),1x,i8,(1x,g9.3),1x,i8,2(1x,g9.3)
         1                    ,1x,i8,4(1x,g9.3))

                    write(*,7001) t,rbar,abar
       &                 ,stdeva,grain,activ,iamax,eavg,mavg,ngb,misavg,cubefrac
                         close(10)
c
c         output the grain size distribution data to area.xy.keyword.trlno
c         and the topological distribution data to nns.xy.keyword.trlno
c
                    open(11,file=keyword//'.'//hundred//one//ten//
        1                     '.'//trlno//'.area')
                    write(11,*) t,rbar,abar,stdeva,grain
                    do 97 site=1,mn
                        if(areas(site).ne.0) then
                            write(11,"(1x,i7,1x,i10,1x,f10.4)")
        1                                 site,areas(site),areas(site)/abar
                        end if
97                  continue
                    close(11)

100       continue
                    if(trial.eq.1) then
                              call micro_out(one,ten,hundred)
                              open(17,file=keyword//'.ph.'//hundred//one//ten)
                              write(17,*) m,n
                              write(17,*) '''',keyword,'''',t,rbar,temp,q
                              write(17,*) length,linevol,nucvol
                              write(17,*) (spins(site),site=1,mn)
                              close(17)
                    endif
```

```
c
c     if the grain size indicates that the n-fold way would be more
c     efficient, change the growth type indicator 'way'
c     and initialize the n-fold way lists
c
              if((way.eq.0).and.(rbar.ge.crossr)) then
                      call lists
                      call indexx(mn,pis,table,index)
                      call binning
                      way=1
                        write(*,*) 'changing to N-fold way!'
                    end if
c
c     determine the time at which the next set of measurements will
c     be taken
c
              if(count.eq.9) then
                      expon=expon+1
                      count=1
              else
                      count=count+1
              end if
              tmeas=float(count)*(10.**expon)
c  edited to include effect of "hundred", 14 sep 03, adr

      end if

c     if the maximum time has been reached OR if the grain radius
c     exceeds one-third the matrix width, take a picture and go to the
c     next trial, otherwise perform another growth step
c
c     write(*,*) 'check: time, tmax, ',t, tmax
c
      ngstop=max0(100,(400-kcount))
c  this algorithm slowly decreases the number of grains at which
c  each coarsening process is stopped
c
c
      time_to_stop=.false.
      proceed=.false.
      inquire(file='stop_now',exist=time_to_stop)
      if(.not.time_to_stop) proceed=.true.

      if(t.lt.tmax.and.activ.gt.0.1.and.grain.gt.40.and.proceed) then
              go to 10
      else
```

```
                    if(kcount.lt.klimit.and.cubefrac.lt.0.3) then
c   no need to continue to iterate if we achieved a high cube fraction!
c
                    write(*,*) 'old, new cube fractions ',oldcubefrac,cubefrac
c                   open(21,file='fraction.cube',access='append')
c                   write(21,7001) t,rbar,abar,stdeva,grain,activ
c                   write(21,7001) t,rbar,abar
c        &                        ,stdeva,grain,activ,nnrexgns,volf,cubefrac
                    write(21,7001) t,rbar,abar
        &                ,stdeva,grain,activ,iamax,eavg,mavg,ngb,cubefrac
c                   close(21)
                              fdiff = oldcubefrac-cubefrac
                              pfdiff=amin1(1.0001,(exp(-50.*fdiff)))
c   obtain a probability of keeping the swap; Boltzman factor if
c   the cube fraction decreased.
c
c                             if(cubefrac.lt.oldcubefrac.and.kcount.gt.1) then
                              if(ran1(iisave).gt.pfdiff.and.kcount.gt.1) then
                              write(*,*) 'Reversing the swap!'
c   we have to reverse the shuffle of grains
                                        itmp=type(iswap1)
                                        type(iswap1)=type(iswap2)
                                        type(iswap2)=itmp
c
                                        tmpa(1)=tayf(iswap1)
                                        tayf(iswap1)=tayf(iswap2)
                                        tayf(iswap2)=tmpa(1)
c
                                        do 4100, i=1,4
                                                  tmpa(i)=aquat(i,iswap1)
4100                          continue
                                        do 4110, i=1,4
                                                  aquat(i,iswap1)=aquat(i,iswap2)
                                                  aquat(i,iswap2)=tmpa(i)
4110                          continue
c
                                        do 4120,i=-1,q
4120                                    tmpa(i)=qtable(i,iswap1)
                                        do 4130,i=-1,q
                                                  qtable(i,iswap1)=qtable(i,iswap2)
4130                                    qtable(i,iswap2)=tmpa(i)
                                        do 4140,i=-1,q
4140                                    tmpa(i)=qtable(iswap1,i)
                                        do 4150,i=-1,q
                                                  qtable(iswap1,i)=qtable(iswap2,i)
4150                                    qtable(iswap2,i)=tmpa(i)
```

```
c
                        do 4220,i=-1,q
4220                    tmpa(i)=mtable(i,iswap1)
                        do 4230,i=-1,q
                              mtable(i,iswap1)=mtable(i,iswap2)
4230                    mtable(i,iswap2)=tmpa(i)
                        do 4240,i=-1,q
4240                    tmpa(i)=mtable(iswap1,i)
                        do 4250,i=-1,q
                              mtable(iswap1,i)=mtable(iswap2,i)
4250                    mtable(iswap2,i)=tmpa(i)

c   *** vii 03; decided that these arrays are not required
                  else
                        oldcubefrac=cubefrac
c   only replace the old cube fraction by the new
c   if we are NOT swapping back
                        expon=int(log10(t))
                        count=int(int(t)/10.**expon)
                        ten=char(48+count)
                        one=char(48+expon)
                        if(kcount.ge.0.and.kcount.lt.10) then
                              decimal=char(48+kcount)
                              altkey='aaaaa'//decimal
                              open(17,file=altkey//'.ph.'//hundred//ten//one)
                        elseif(kcount.ge.10.and.kcount.lt.100) then
                              decimal=char(48+mod(kcount,10))
                              hun=char(48+(kcount/10))
                              altkey='aaaa'//hun//decimal
                              open(17,file=altkey//'.ph.'//hundred//ten//one)
                        elseif(kcount.ge.100.and.kcount.lt.1000) then
                              decimal=char(48+(mod(kcount,10)))
                              hun=char(48+(mod((kcount/10),10)))
                              thou=char(48+(kcount/100))
                              altkey='aaa'//thou//hun//decimal
                              open(17,file=altkey//'.ph.'//hundred//ten//one)
                        elseif(kcount.ge.1000.and.kcount.lt.10000) then
                              decimal=char(48+(mod(kcount,10)))
                              hun=char(48+(mod((kcount/10),10)))
                              thou=char(48+(mod((kcount/100),10)))
                              tenthou=char(48+(kcount/1000))
                              altkey='a'//tenthou//thou//hun//decimal
                              open(17,file=altkey//'.ph.'//hundred//ten//one)
                        elseif(kcount.ge.10000) then
                              stop 'too high an iteration count!'
                        endif
                        write(*,*) 'made up file name ',altkey
                        write(17,*) m,n
                        write(17,*) '"',keyword,'"',t,rbar,temp
                        write(17,*) length,linevol,nucvol
                        write(17,*) (spins(site),site=1,mn)
                        close(17)
                        swpkey=keyword
                        keyword=altkey
                        call tex_out(one,ten,hundred,cubefrac)
                        call micro_out(one,ten,hundred)
                        keyword=swpkey
c
                  endif
c  now we shuffle a pair of grains
                  iswap1=iran(q2)+q2
4500        iswap2=iran(q2)+q2
                  write(*,*) '1st, 2nd pick for swap = ',iswap1,iswap2
c  put both on the rex range
c                 if((iswap2.eq.iswap1).or.(type(iswap1).eq.type(iswap2))) goto 4500
c  and go around again if they represent no shuffle, or if they are of
c  the same type of orientation, e.g. both cube
                  if(iswap2.eq.iswap1) goto 4500
c  and now to swap
            write(*,*) 'Performing swap of grains ',iswap1,' and ',iswap2
                        itmp=type(iswap2)
                        type(iswap2)=type(iswap1)
                        type(iswap1)=itmp

                        tmpa(1)=tayf(iswap2)
                        tayf(iswap2)=tayf(iswap1)
                        tayf(iswap1)=tmpa(1)

                        do 4600, i=1,4
                              tmpa(i)=aquat(i,iswap2)
4600              continue
                        do 4610, i=1,4
                              aquat(i,iswap2)=aquat(i,iswap1)
                              aquat(i,iswap1)=tmpa(i)
4610              continue

                        do 4620,i=-1,q
4620                    tmpa(i)=qtable(i,iswap2)
                        do 4630,i=-1,q
                              qtable(i,iswap2)=qtable(i,iswap1)
4630                    qtable(i,iswap1)=tmpa(i)
                        do 4640,i=-1,q
```

```
4640                            tmpa(i)=qtable(iswap2,i)
                                do 4650,i=-1,q
                                        qtable(iswap2,i)=qtable(iswap1,i)
4650                            qtable(iswap1,i)=tmpa(i)
c
                                do 4720,i=-1,q
4720                            tmpa(i)=mtable(i,iswap2)
                                do 4730,i=-1,q
                                        mtable(i,iswap2)=mtable(i,iswap1)
4730                            mtable(i,iswap1)=tmpa(i)
                                do 4740,i=-1,q
4740                            tmpa(i)=mtable(iswap2,i)
                                do 4750,i=-1,q
                                        mtable(iswap2,i)=mtable(iswap1,i)
4750                            mtable(iswap1,i)=tmpa(i)

c   *** vii 03; these arrays not needed
                                kcount=kcount+1
                                write(*,*) 'swap completed, iteration ',kcount
                                goto 4010
c
                else
                        write(*,*) 'MAIN, exiting: time, activ= ',t,activ

                endif
        end if
900     continue
        close(10)
        close(21)
        close(22)
        stop
        end
```

APPENDIX B – This subroutine consists of energetic considerations such as surface/interface, grain boundary and strain energies to check the probability a given spin flip for MCGROWTH1 subroutine as a function of thickness and temperature. This subroutine shows the case of 100nm at 300°C annealing conditions.

MCPROBS1.F

```
c         MCPROBS1.F:    Determines the probability of a given
c         spin flip for the 'MCGROWTH1' subroutine
c         ------------------------------------------------------------
c         copyright 1994 Elizabeth A. Holm
c
          function mcprobs(site,newspin)

          include 'common.f'

          integer i,w,neigh,site,newspin,oldspin,ng,spsite,spspin
          real eprob,qr(4),qq(4),d(3,3)
          real d1,d2,d3,th,dd1,dd2,dd3
          double precision c1,c,c2,s1,s,s2,pi
          double precision h,k1,l,rnorm,max_t
          integer Tdep,Tanneal,j
          real ENgb,ENlagb,ENictb,ENctb,ENsur,de,de2,ENStr,ENinter
          real ENgb2,ENlagb2,ENictb2,ENctb2,ENsur2,ENstr2,ENinter2
          real oldbulk,newbulk
          real p(3,3),DET,DeltaG,EqdeltaG
          real c11,c12,c44,Mhkl,G,e
c         rad=57.29578
c         real stored
c         data stored / 2.5 /    !    now in common, vii 00    adr


          oldspin=spins(site)

c         newspin=oldspin
c         oldbulk=tayf(oldspin)*stored
c         if(oldspin .gt. q2) oldbulk=0.
c         newbulk=1.        ! from EAH
c         newbulk=stored
c         newbulk=tayf(newspin)*stored
c         if(newspin .gt. q2) newbulk=0.



          do 5000 c=1,4
```

```
                  qr(c)=aquat(c,oldspin)
5000        continue
          call q2eulB(d1,d2,d3,qr)
c         print*,'debug1=',d1,d2,d3

c         d1=d1*pi/180.
c         d2=d2*pi/180.
c         d3=d3*pi/180.

          c1=DCOS(1d0*d1)
          c=DCOS(1d0*d2)
          c2=DCOS(1d0*d3)
          s1=DSIN(1d0*d1)
          s=DSIN(1d0*d2)
          s2=DSIN(1d0*d3)

          d(1,1)=c1*c2-s1*s2*c
          d(1,2)=s1*c2-c1*s2*c
          d(1,3)=s2*s
          d(2,1)=-c1*s2-s1*c2*c
          d(2,2)=-s1*s2-c1*c2*c
          d(2,3)=c2*s
          d(3,1)=s1*s
          d(3,2)=-c1*s
          d(3,3)=c
c         direction checking according to ND
          rnorm=dsqrt((s2*s)**2+(s*c2)**2+c**2)
          h= (s2*s)/rnorm
          k1= (s*c2)/rnorm
          l= (c)/rnorm

c         p(1,1)= d(3,3)*d(2,2)-d(3,2)*d(2,3)
c         p(1,2)=-(d(3,3)*d(1,2)-d(3,2)*d(1,3))
c         p(1,3)= d(2,3)*d(1,2)-d(2,2)*d(1,3)
c         p(2,1)=-(d(3,3)*d(2,1)-d(3,1)*d(2,3))
c         p(2,2)= d(3,3)*d(1,1)-d(3,1)*d(1,3)
c         p(2,3)=-(d(2,3)*d(1,1)-d(2,1)*d(1,3))
c         p(3,1)= d(3,2)*d(2,1)-d(3,1)*d(2,2)
c         p(3,2)=-(d(3,2)*d(1,1)-d(3,1)*d(1,2))
c         p(3,3)= d(2,2)*d(1,1)-d(2,1)*d(1,2)
c         DET =d(1,1)*(d(3,3)*d(2,2)-d(3,2)*d(2,3))-d(2,1)*(d(3,3)*d(1,2)-
c         d(3,2)*d(1,3))+d(3,1)*(d(2,3)*d(1,2)-d(2,2)*d(1,3))
c         DeltaG=d(3,3)*1/DET*p(3,3)
```

```
      th=100/1000000000.0
      Tdep=25.0
      Tanneal=300.0

      max_t=h
      max_t=max(max_t,k1)
      max_t=max(max_t,l)

      ENsur=(2.710+(((-5.0)/10000)*(Tanneal-Tdep)))*max_t
      ENinter=ENsur

      ENlagb = 0.7175+((-1/10000)*(Tanneal-Tdep))
      ENictb = 0.593+((-1/10000)*(Tanneal-Tdep))
      ENctb = 0.04+((-2/100000)*(Tanneal-Tdep))
      ENgb=(ENlagb+ENictb+ENctb)*th

      e=13.9*273/1000000
      Mhkl=(c11+c12+G-2*(((c12-G)**2)/(c11-2*G)))*(102*10000000/9.8)
      G=(2*c44-c11+c12)*(h**2*k1**2+k1**2*l**2+l**2*h**2)
      c11=(171.0825+(-0.0353)*(Tanneal-Tdep))
      c12=(123.5825+(-0.0153)*(Tanneal-Tdep))
      c44=(76.0925+(-0.0277)*(Tanneal-Tdep))
      ENstr= G*e**2*Mhkl*th
      Mhkl=(-1)*Mhkl
c     ENstr=(-1)*ENstr
      de=2*ENsur+ENstr
c     de=(-1)*de

      do 15, j=1,6

      spspin=spins(neighs(site,j))
            do 5010 c=1,4
            qq(c)=aquat(c,spspin)

5010    continue
      call q2eulB(dd1,dd2,dd3,qq)

c     dd1=dd1*pi/180.
c     dd2=dd2*pi/180.
c     dd3=dd3*pi/180.
```

```
      c1=DCOS(1d0*dd1)
      c=DCOS(1d0*dd2)
      c2=DCOS(1d0*dd3)
      s1=DSIN(1d0*dd1)
      s=DSIN(1d0*dd2)
      s2=DSIN(1d0*dd3)

      d(1,1)=c1*c2-s1*s2*c
      d(1,2)=s1*c2-c1*s2*c
      d(1,3)=s2*s
      d(2,1)=-c1*s2-s1*c2*c
      d(2,2)=-s1*s2-c1*c2*c
      d(2,3)=c2*s
      d(3,1)=s1*s
      d(3,2)=-c1*s
      d(3,3)=c

      rnorm=dsqrt((s2*s)**2+(s*c2)**2+c**2)
      h= (s2*s)/rnorm
      k1= (s*c2)/rnorm
      l= (c)/rnorm

c     p(1,1)= d(3,3)*d(2,2)-d(3,2)*d(2,3)
c     p(1,2)=-(d(3,3)*d(1,2)-d(3,2)*d(1,3))
c     p(1,3)= d(2,3)*d(1,2)-d(2,2)*d(1,3)
c     p(2,1)=-(d(3,3)*d(2,1)-d(3,1)*d(2,3))
c     p(2,2)= d(3,3)*d(1,1)-d(3,1)*d(1,3)
c     p(2,3)=-(d(2,3)*d(1,1)-d(2,1)*d(1,3))
c     p(3,1)= d(3,2)*d(2,1)-d(3,1)*d(2,2)
c     p(3,2)=-(d(3,2)*d(1,1)-d(3,1)*d(1,2))
c     p(3,3)= d(2,2)*d(1,1)-d(2,1)*d(1,2)
c     DET =d(1,1)*(d(3,3)*d(2,2)-d(3,2)*d(2,3))-d(2,1)*(d(3,3)*d(1,2)-
d(3,2)*d(1,3))+d(3,1)*(d(2,3)*d(1,2)-d(2,2)*d(1,3))
c     DeltaG=d(3,3)*1/DET*p(3,3)

      th=100/1000000000.0
      Tdep=25.0
      Tanneal=300.0

      max_t=h
      max_t=max(max_t,k1)
      max_t=max(max_t,l)
```

```
        ENsur2=(2.710+(((-5.0)/10000)*(Tanneal-Tdep)))*max_t
        ENinter2=ENsur2
        ENlagb2 = 0.7175+((-1/10000)*(Tanneal-Tdep))
        ENictb2 = 0.593+((-1/10000)*(Tanneal-Tdep))
        ENctb2 = 0.04+((-2/100000)*(Tanneal-Tdep))
        ENgb2=(ENlagb+ENictb+ENctb)*th

        e=13.9*273/1000000
        G=(2*c44-c11+c12)*(h**2*k1**2+k1**2*l**2+l**2*h**2)
        c11=(171.0825+(-0.0353)*(Tanneal-Tdep))
        c12=(123.5825+(-0.0153)*(Tanneal-Tdep))
        c44=(76.0925+(-0.0277)*(Tanneal-Tdep))
        Mhkl=(c11+c12+G-2*(((c12-G)**2)/(c11-2*G)))*(102*10000000/9.8)

        ENstr2= G*e**2*Mhkl*th
        Mhkl=(-1)*Mhkl
c       ENstr=(-1)*ENstr

        de2=2*ENsur2+ENstr2
c       de2=(-1)*de2

        if (de .gt. de2) then
                de=de2
                newspin = spspin
        end if

c       print*, 'energy = ',ENsur, oldspin, newspin
15      continue

c
c       mobility=amin1(1.0,mtable(oldspin,newspin))
c
c       eprob=amin1(1.0,qtable(oldspin,newspin))
c       eprob=qtable(oldspin,newspin)
c
c       find the probability of a flip with change in energy = de
c       Rule: 'oldspin' flips to 'newspin' with probability
c
c               P={1 for dE<=0} and P={exp(dE/kT) for dE>0}
c
c       but now, if either the energy varies or the mobility, then the
c       probability may be less than one
c
```

```
c     mcprobs=amin1(1.0,mobility*eprob)
      mcprobs=0.
      if(de.le.0.) then
      mcprobs=1.
      end if
c    mcprobs =0.
c   if(de.le.0.)then
c      mcprobs =1
c   else
      if(de.gt.0.)then
c     mcprobs =exp(-1*de/Taneal/(8.617E-5))
c     end if
c     mcprobs=exp(-1*de/Tanneal/(8.617E-5))
      mcprobs=exp(-1*de/Tanneal/(8.617E-5)/273.15)
c     mcprobs=exp(-1*de/Tanneal/(1.3806504E-23)/273.15)
c* amin1(1.0,mobility*eprob)
c     mcprobs=amin1(1.0,mobility*eprob*exp(-1*de/Taneal/(8.617E-5)))
c     mcprobs=amin1(1.0,mobility*eprob)
      end if
c
c           if(mcprobs .gt. 1.0) stop 'error in MCPROBS!!!'
c
c     mobility is based on the piece of boundary that moves, even if
c     this is a wildcard flip.
c
c     else
c           if(temp .gt. 0.) then
c                 mcprobs=mobility*eprob
c   &             *exp(-1.*de/temp/eprob)
      if(mcprobs.gt.1.0) stop 'error in MCPROBS!!!'
c     end if
c     modify probability by mobility
c
c   2 XII 00, adr
c  ++++ see discussion above for this use of variable temperature depending on the
c  magnitude of the grain boundary energy.
c
c           end if
c     end if
c
c     return to main
c
      return
      end
```

APPENDIX C – This subroutine shows the possibilities to find twins by 60° rotation about the <111> crystal direction with 24 symmetry operator at cubic system.

TWFIND.F

```
c     1.get g(orientation)
c     Euler angles specify a triple of rotations (transformations) about the Z,X and Z
c     directions.
c     Euler angles are presented according to Bunge's definition
c
c   aij = |cos(p1)*cos(p2)-sin(p1)*sin(p2)*cos(p)    sin(p1)*cos(p2)+cos(p1)*sin(p2)*cos(p)    sin(p2)*sin(p)|
c         |-cos(p1)*sin(p2)-sin(p1)*cos(p2)*cos(p)   -sin(p1)*sin(p2)+cos(p1)*cos(p2)*cos(p) cos(p2)*sin(p)|
c         |                sin(p1)*sin(p)                              -cos(p1)*sin(p)                 cos(p)|

c   G = |cos(p1)*cos(p2)-sin(p1)*sin(p2)*cos(p)      sin(p1)*cos(p2)+cos(p1)*sin(p2)*cos(p)    sin(p2)*sin(p)|
c         |-cos(p1)*sin(p2)-sin(p1)*cos(p2)*cos(p)   -sin(p1)*sin(p2)+cos(p1)*cos(p2)*cos(p)   cos(p2)*sin(p)|
c         |                sin(p1)*sin(p)                              -cos(p1)*sin(p)                 cos(p)     |

        subroutine twfind
        include 'common.f'

        real d1,d2,d3,dd1,dd2,dd3
        real Ga,inverseG1,d(3,3),a(3,3),p(3,3),DET
        real G2,DeltaG,EqdeltaG,b(3,3),neigh,qr(4),qq(4)
        real hi(24),de,th
        real ENgb,ENlagb,ENictb,ENctb
c       real neiang1,neiang2,neiang3,twang1,twang2,twang3
        double precision c1,c,c2,s1,s,s2,pi
        double precision r1,r,r2,t1,t2,t3
        integer site,ng,i,j,y,l,pltw,v
        integer twspin,neispin,oldspin,spspin
        integer Ta,Tanneal
        integer n1,n2,n3,n4,n5,n6
c
c   G1 = |d(1,1) d(1,2) d(1,3)|
c          |d(2,1) d(2,2) d(2,3)|
c          |d(3,1) d(3,2) d(3,3)|
        pi=3.14159
        do 5000 c=1,4
                qr(c)=aquat(c,oldspin)
5000       continue
c       print*,'debug=',
        call q2eulB(d1,d2,d3,qr)
c       print*,'debug1=',d1,d2,d3
```

```
c       d1=d1*pi/180.
c       d2=d2*pi/180.
c       d3=d3*pi/180.
        c1=DCOS(1d0*d1)
        c=DCOS(1d0*d2)
        c2=DCOS(1d0*d3)
        s1=DSIN(1d0*d1)
        s=DSIN(1d0*d2)
        s2=DSIN(1d0*d3)

        d(1,1)=c1*c2-s1*s2*c
        d(1,2)=s1*c2-c1*s2*c
        d(1,3)=s2*s
        d(2,1)=-c1*s2-s1*c2*c
        d(2,2)=-s1*s2-c1*c2*c
        d(2,3)=c2*s
        d(3,1)=s1*s
        d(3,2)=-c1*s
        d(3,3)=c


        do 15, j=1,nbors

        spspin=spins(neighs(site,j))
                do 5010 c=1,4
                qq(c)=aquat(c,spspin)

5010       continue
        call q2eulB(dd1,dd2,dd3,qq)
c       print*,'debug3= ',dd1,dd2,dd3

c       dd1=dd1*pi/180.
c       dd2=dd2*pi/180.
c       dd3=dd3*pi/180.

        r1=DCOS(1d0*dd1)
        r=DCOS(1d0*dd2)
        r2=DCOS(1d0*dd3)
        t1=DSIN(1d0*dd1)
        t2=DSIN(1d0*dd2)
        t3=DSIN(1d0*dd3)

        a(1,1)=r1*r2-t1*t3*r
        a(1,2)=t1*r2-r1*t3*r
        a(1,3)=t3*t2
        a(2,1)=-r1*t3-t1*r2*r
```

```
          a(2,2)=-t1*t3-r1*r2*r
          a(2,3)=r2*t2
          a(3,1)=t1*t2
          a(3,2)=-r1*t2
          a(3,3)=r

          p(1,1)= a(3,3)*a(2,2)-a(3,2)*a(2,3)
          p(1,2)=-(a(3,3)*a(1,2)-a(3,2)*a(1,3))
          p(1,3)= a(2,3)*a(1,2)-a(2,2)*a(1,3)
          p(2,1)=-(a(3,3)*a(2,1)-a(3,1)*a(2,3))
          p(2,2)= a(3,3)*a(1,1)-a(3,1)*a(1,3)
          p(2,3)=-(a(2,3)*a(1,1)-a(2,1)*a(1,3))
          p(3,1)= a(3,2)*a(2,1)-a(3,1)*a(2,2)
          p(3,2)=-(a(3,2)*a(1,1)-a(3,1)*a(1,2))
          p(3,3)= a(2,2)*a(1,1)-a(2,1)*a(1,2)

          DET =a(1,1)*(a(3,3)*a(2,2)-a(3,2)*a(2,3))-a(2,1)*(a(3,3)*a(1,2)-
a(3,2)*a(1,3))+a(3,1)*(a(2,3)*a(1,2)-a(2,2)*a(1,3))



          do 300, v=1,24
          call symoperator(hi)
          b(3,3)= hi(v)* d(3,3) * 1/DET * p(3,3)
300         continue
15        continue
          pltw=0
          EqdeltaG=b(3,3)

          if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq.
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. 1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. 1/sqrt(3.)))then
          pltw=1
          end if


          if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq. -
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. -1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. -1/sqrt(3.)))then
          pltw=1
          end if


          if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq. -
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. 1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. 1/sqrt(3.))) then
          pltw=1
          end if
```

```
        if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq.
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. -1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. 1/sqrt(3.))) then
        pltw=1
        end if


        if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq.
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. 1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. -1/sqrt(3.))) then
        pltw=1
        end if
        if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq. -
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. -1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. 1/sqrt(3.)))then
        pltw=1
        end if


        if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq. -
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. 1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. -1/sqrt(3.))) then
        pltw=1
        end if


        if ((((b(1,1)+b(2,2)+b(3,3)-1)/2) .eq. cos(60.*pi/180)).and. ((b(2,3)-b(3,2)) .eq.
1/sqrt(3.)) .and. ((b(3,1)-b(1,3)) .eq. -1/sqrt(3.)) .and. ((b(1,2)-b(2,1)) .eq. -1/sqrt(3.))) then
        pltw=1
        end if


        return
        end
```