Interfacial Structure of Delta Phase in Inconel 718 and the Selection of Precipitate Habit Planes

by

Qiang Liang

Dissertation submitted to the faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Materials Engineering Science

APPROVED:

Prof. W. T. Reynolds Jr., Chairman

Prof. D. Farkas

Prof. S. B. Desu

Prof. W. A. Curtin Jr. (MSE/ESM)

Prof. S. L. Kampe

August, 1996

Blacksburg, Virginia

Keywords: Precipitate morphology, Interfacial structure, Habit plane, Conjugate plane, Coincident sites, Matching

Interfacial Structure of Delta Phase in Inconel 718 and the Selection of Precipitate Habit Planes

by

Qiang Liang

Committee Chairman: Prof. W. T. Reynolds Jr.

Materials Engineering Science Program

(ABSTRACT)

We investigated the structure and defects associated with interphase boundaries between a γ (fcc) matrix and plate-shaped precipitates of the δ (orthorhombic) phase in Inconel 718. Based upon transmission electron microscopy (TEM) observations, the average habit plane was confirmed to be $(111)_\gamma$ which is consistent with previous reports. A parallel array of misfit dislocations with Burgers vector $\mathbf{b}=\frac{1}{6}[11\bar{2}]_\gamma$ (designated M1) are always observed lying along the $[1\bar{1}0]$ direction. Another array of misfit dislocations appears in some regions of the interface with Burgers vector $\mathbf{b}=\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ (designated M2). These dislocations also lie along the $[1\bar{1}0]$ direction. Irregular ledges were identified on the interface and are believed to contribute to the thickening of δ plates. Dislocations in the matrix were also characterized. Most matrix dislocations have a $\frac{1}{2}[\bar{1}01]_\gamma$ Burgers vector. The growth ledges in the habit plane of a single δ plate have a variety of effective Burgers vectors. A geometric matching approach based upon near-coincident sites was employed to explain the interfaces structure of interphase boundaries in Inconel 718, as well as fcc/bcc in Ni–45wt% Cr. In both cases, the conjugate plane is the plane with the highest areal density of near-coincident sites over a small region while the average habit plane is determined by the continuity of near-coincidence sites over a large area. The M1 interfacial dislocations in the γ/δ interface accommodate misfit in the habit plane whereas M2 dislocations do not and are probably a by-product of the dissociation of matrix dislocations. In the fcc/bcc system, the habit plane is not parallel to the conjugate plane and the partial dislocations associated with matrix

stacking faults improve matching in the habit plane even though their Burgers vector lies out of this plane.

ACKNOWLEDGEMENTS

I would like to call Blacksburg as my second hometown. Since it is the place where Virginia Tech is located, where I was lucky enough to know my advisor, Prof. W. T. Reynolds Jr. He has not only taught and helped me in my research field, but also influenced me by his attitude to science, and kindness of getting along with people. Without his trust, I could not have become a Ph.D candidate in the phase transformation group, and this work could not have been accomplished. During my stay in the group, I was also pleased to work with Drs. Gang Chen and J. K. Chen. I owe the thanks for their tremendous helps in transmission electron microscopy and computer operation. Especially, the valuable discussions with J. K. Chen on interfacial structures and matching are greatly appreciated.

The conscientious reviews of this manuscript by Prof. D. Farkas, Prof. S. B. Desu, Prof. W. A. Curtin Jr, and Prof. S. L. Kampe, are gratefully acknowledged. Prof. J. M. Howe, and Dr. M. M. Tsai of University of Virginia for help with high resolution transmission electron microscopy is deeply appreciated. Prof. G. Q. Lu, Prof. G. V. Gibbs, Drs. Baoping He, and Tingkai Li are thanked for their friendship and encouragement. The financial support of National Science Foundation through grant DMR93–03518 is also gratefully acknowledged.

I would also like to give my acknowledgements to my friends and fellow students who gave me their friendship during my years at Virginia Tech. They are: Prof. Reynolds' wife, Mariko Reynolds, Ms. Jan Doran, Ms. Amy Hill, Mr. G. Levon Fattal, Mr. Jud Marte, Mr. Mike Stawovy, Mr. Dave Teter, Dr. Prabodh L. Ratnaparkhi, Ms. Heidi. D. Allison, Ms. Becky. Herrmann and Ms. Batsirai Mutasa.

Mr. R. L. Kennedy of TELEDYNE/ALLVAC/VASC Co. for providing the sample material is appreciated.

Finally, I wish to thank my family in China who gave me their every kind of support over the years. Especial thanks is given to my wife Wei for her love and understanding. The award of this accomplishment belongs to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

It is well-known that the morphology of precipitates plays an important role in determining a material's properties. According to a general theory of precipitate morphology [1, 2], the structure of the interface boundary enclosing a precipitate is important to understanding precipitate shape, In real alloy systems, precipitates can adopt a rod-shape [3, 4, 5], needle-shape [6, 7, 8], plate-shape [9, 10, 11, 12, 13, 14, 15], or lath-shape [16, 17, 18, 19]. The better understanding of the formation mechanism for these different shaped precipitates would enable us to control precipitates morphology in the future.

Generally, precipitation transformations can be divided into two categories: martensitic (massive) [20], or diffusional [21]. And the interfacial structure of plate-shaped precipitates has been viewed as one of two types: glissile, or sessile. The former is a good example for martensitic transformation, in which precipitate growth is taken to occur through the glide of interface dislocations. On the other hand, precipitates with sessile interface are usually referred to diffusional transformation, in which atomic rearrangements to change the boundary crystal structure and redistribute solute and solvent atoms is accomplished by a biased random walk of individual atoms [22]. Two important parameters, precipitate elongation direction and habit plane (broad face direction), are always used to analyze the validity of a transformation model, although interfacial defects are also mentioned in some cases.

The phenomenological theory of martensite crystallography (PTMC), based on the lat-

tice correspondence, was built to explain the precipitate interfacial structure and growth of martensitic transformations [23, 24, 25, 26]. It requires a homogeneous strain. PTMC can be expressed as [27, 28]:

$$\mathbf{X_2} = \mathbf{A}\mathbf{X_1} = \mathbf{R}\mathbf{S_2}\mathbf{S_1^{-1}}\mathbf{X_1} \tag{1.1}$$

The transformation matrix, **A**, is described in terms of three components: a rotation matrix, **R**, which is obtained from the orientation relationship, and $\mathbf{S_1}$ and $\mathbf{S_2}$ whose columns are the basis vectors of lattice 1 and 2, respectively. The quantity $\mathbf{S_2}\mathbf{S_1^{-1}}$ is referred to as the "lattice correspondence".

Other theories, although their starting point are different from PTMC, they finally fall into the same category, such as the invariant line theory [29, 30], and O-lattice theory [28, 31, 32, 33]. This shows all these models give an unique transformation matrix, **A**, to describe the precipitate interface and growth process. Therefore, a lattice correspondence is always involved and a homogeneous strain is required. For the most studied fcc-bct martensitic transformation in ferrous alloys, a lattice correspondence is generally assumed as Bain correspondence [34, 35, 27].

The PTMC successfully predicts the growth direction of some lath- or plate-shaped precipitates systems by diffusion-controlled precipitation reactions [5, 36, 16], such as in Cu–0.33wt% Cr and Ni–45wt% Cr systems, thus it has even been suggested that the PTMC is also applicable to diffusion controlled precipitation reactions which are necessarily nonmartensitic[37].

Precipitate boundaries with a sessile dislocation structure are generally believed to be displaced by a growth ledge mechanism. No phenomenological theory is available to describe the crystallography of such a transformation and a lattice correspondence may be present, but is not required by the transformation mechanism. Transmission electron microscopy

(TEM) is currently the only available tool for studying the structure of the boundary between the precipitates and the parent matrix phase. These interface defects are the link between growth mechanism and precipitate shape that has been missing from the general theory of precipitate morphology[38]. Meanwhile, computer simulations has also been employed to analyze the precipitate interphase boundary, especially in Al, and Cu alloys [11, 39, 40, 41, 42].

Aaronson proposed a general theory to describe the precipitate shape in terms of the the boundary mobility during diffusional precipitation transformation [1]. Better matching boundary direction is believed to have higher mobility, and thus grow out leaving behind facets composed of lower mobility boundaries. The good matching boundary direction can be achieved by uncoordinated motion of individual atoms jumps.

Chen et al [43] also reported a convinced result by employing atomic matching rather than lattice correspondence to predict the habit plane of fcc/bcc in Ni–45wt% Cr system.

However, something is still unclear for the understanding of good boundary matching directions. Such as, interfacial dislocation are generally believed to compensate the misfit on interface in diffusional transformation theory, but how they exactly function; the purpose of stacking faults in Chen's experimental results [19]; the relationship between conjugate direction and best matching direction; and the relationship among conjugate plane, habit plane, and best locally matching plane. It is the goal of this dissertation to propose a method for better predicting the precipitate growth direction and habit plane between two arbitrary orientated crystals, based upon the atomic matching approach and the experimental observed defects and atomic structures on interface.

The discussions are made in the view of: interfacial structure and related defects (Chapter 2), and the derivation of habit plane and best matching direction (Chapter 3). Each of two chapters is prepared as a paper for the open literature, thus each contains introduction,

procedure, results, discussion, and conclusions sections as required by technical papers. References to the chapters are listed as follows: Chapter 2 is Reference [44], Chapter 3 is Reference [45]. Appendix A is the Appendix for Reference [45].

In Chapter 2, the interphase boundary structure and related defects for thirty δ precipitates in Inconel 718 are examined and identified. The orientation relationship and habit plane are found to be consistent with previous results [46] for orthorhombic δ in fcc γ matrix. The importance of the ledges associated with the precipitate growth is discussed. Lattice correspondence model employed is failed to explain the boundary structure.

Chapter 3 adopts a geometric atomically matching approach to determine the interphase boundary orientations, based upon of the near coincident sites (NCS). The method does not involve the lattice correspondence selection. Experimental observed interfacial structures of δ/γ in Inconel 718 and fcc/bcc in Ni–45%wt Cr are the strong evidence for the method. Relationship among conjugate plane, habit plane, and best locally matching plane determined by the near-coincident site calculation is discussed.

Appendix A gives a brief description of each program used in Chapter 3. These programs are included with the dissertation as: a program (*ncs718elds.c*) produces near-coincident sites satisfied coincidence criterion in δ/γ system; a program (*ncsfbelds.c*) produces near-coincident sites in fcc/bcc; a program (*plane.c*) produces an array of near-coincident sites within the mismatch assigned over a large area; a program (*bmd.c*) produces the best matching direction of all near-coincident sites over the simulation block.

Chapter 2

Interfacial Structure of Plate-Shaped Delta Precipitates in Inconel 718

2.1 Introduction

Precipitate morphology has an important influence on a material's properties. According to a general theory of precipitate morphology [1, 2], a precipitate's shape is controlled by the anisotropic mobility of its boundaries. The mobility of a precipitate's boundaries depends upon their structure. Thus, interphase boundary structure is important to understanding precipitate shape.

The structure of the boundary between a precipitate and its parent matrix determines the mechanism by which the precipitate grows. Historically, the interfacial structure of plate-shaped precipitates has been viewed as one of two types: glissile, which leads to a lattice correspondence and a homogenous strain [20]; or sessile, which leads to a reconstructive transformation mechanism. Growth with a lattice correspondence is taken to occur through the glide of interface dislocations, whereas reconstructive, or diffusional growth, is considered to occur by means of a biased random walk of individual atoms [21, 22].

Over the years, the migration mechanisms of boundaries between fcc and bcc phases have received considerable attention [47, 16, 48, 19, 49, 18]. For this pair of phases, the habit plane of the precipitate phase does not correspond to the conjugate planes of the orientation relationship (the plane in the precipitate and the plane in the matrix which

are parallel to each other). A lattice correspondence is often assumed to occur between lath-shaped precipitates and their parent matrix, but recent evidence suggests lath-shaped precipitates can evolve by a reconstructive mechanism without a lattice correspondence [43]. In many fcc/bcc systems, precipitates grow preferentially along a low misfit direction, and the two phases match well across the precipitate habit plane.

This investigation was initiated to see if these trends extend to cases in which the habit plane coincides with the conjugate plane of the orientation relationship. Although this connection has been established in a number of systems (e.g. γ' in Al–Ag [50, 51], and α_2 in Ti–Al [11, 52, 53]), the conjugate planes of the matrix and precipitate in these cases have nearly identical atomic arrangements, and misfit accommodation is not needed in the habit plane.

The interface boundary between (δ) precipitates (Ni_3Nb) in the Ni-based alloy Inconel 718, on the other hand, represents more than just a change in stacking sequence. The matrix phase is a disordered, fcc solid solution, designated as γ. Well-formed plates of the ordered δ phase (an orthorhombic phase with a DO_a structure) precipitate at aging temperatures above $920°C$. The habit plane of the δ plates is parallel to the conjugate plane of the orientation relationship and contains an array of interfacial dislocations. Thus, these plates provide a useful vehicle to investigate how interfacial matching influences precipitate shape.

2.2 Procedure

2.2.1 Sample Preparation

The sample material used in this investigation was cut from forged and annealed bar stock 30 mm in diameter. The chemical analysis provided by the manufacturer[1] is shown

[1]TELEDYNE/ALLVAC/VASCO, Monroe, NC 28110

in Table 2.1.

Table 2.1: Chemical Composition (wt%) of commercial superalloy Inconel 718

Cr	Fe	Ni	Mo	Nb	Ti	Al	C	Mn	Cu	W	Si
17.99	17.83	54	2.87	5.12	0.99	0.49	0.028	0.11	0.07	0.05	0.07

Portions of the bar stock were first cold rolled to 0.5 mm sheet, encapsulated in quartz tubes, and evacuated to a pressure of $< 10^{-4}$ torr. The sheets then were solutionized at 1190°C for three hours before quenching in iced-brine. Based on published precipitation kinetics [54], the solutionized samples were encapsulated and evacuated again, and isothermally aged at 930°C for 50, 66, 74, and 100 hours to produce the plate-shaped δ precipitates from the γ matrix. The γ' and γ'' phases often found in Inconel 718 (In718) were not observed at this aging temperature.

Thin foils were prepared by mechanically grinding the aged samples to approximately 95 μm, punching 3 mm diameter disks, and jet polishing with a South Bay single jet polisher (model 550–C) in a solution of 10% perchloric acid, 70% 1-Butanol, and 20% Ethanol [55, 56], at a temperature between -20°C and -30°C and a potential between 150 and 200 V. The thin foils were studied using a Philips EM 420 transmission electron microscope (TEM) operated at 120 kV and a JEOL 4000EX high resolution electron microscope (HREM) operated at 400 kV.

The precipitate habit plane and defect direction were identified using trace analysis in at least three non-coplanar zones. Interfacial defects and dislocations were characterized using the $\mathbf{g} \cdot \mathbf{b} = 0$ invisibility criterion [57, 58, 59, 48] under two-beam conditions with multiple $\mathbf{g}$ vectors. Interfacial ledges were identified by the displacement of thickness fringes obtained

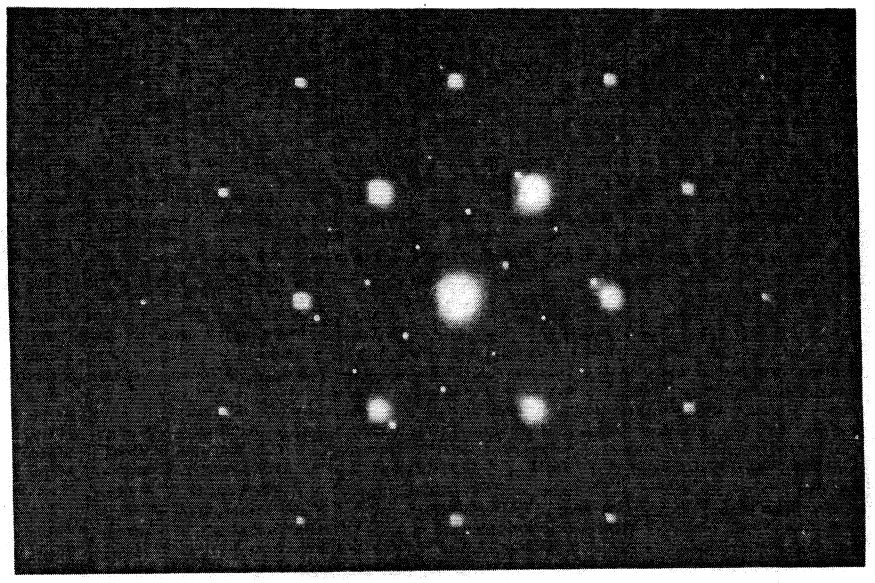
from a precipitate reflection not duplicated in the matrix with the sample tilted slightly away from the Bragg condition [60, 61]. The weak-beam dark field (WBDF) technique was used to resolve the fine details of the interfacial structure [62, 63].

2.3 Results

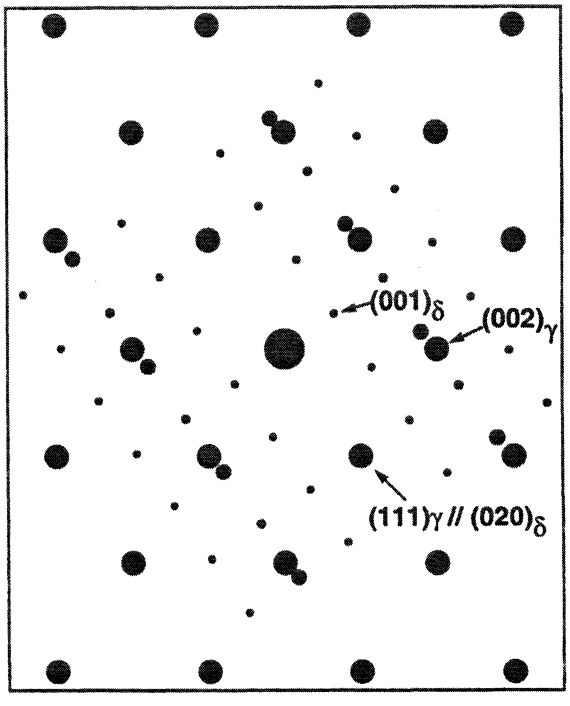
Selected area diffraction patterns (SADP) of a δ precipitate and the adjacent γ matrix are shown in Figure 2.1(a) and Figure 2.2(a). The corresponding indices are shown in the calculated diffraction patterns of Figure 2.1(b) and Figure 2.2(b). The extra reflections in Figures 2.1(a) and 2.2(a) at $(010)_\delta$ and $(100)_\delta$ are attributed to double diffraction. The orientation relationship (OR) obtained from diffraction patterns is: $(111)_\gamma \parallel (010)_\delta$, $[1\bar{1}0]_\gamma \parallel [100]_\delta$. The average habit plane determined by the trace analysis on thirty precipitates is $(1\ 1.05\ 1.03)_\gamma$. The largest deviation from this value is about 5.2°. The OR and average habit plane are both consistent with the data of previous workers [46, 55, 64, 65, 66].

Three types of interfacial features were identified in the $(111)_\gamma$ habit plane of the δ precipitate plates: an array of linear features with weak contrast, a second array of linear features parallel to the first but with stronger contrast, and ledges. Arrays of dislocations in the matrix adjacent to δ plates that appeared to be associated with the growth of the plates were also characterized.

Weak features were observed in the habit planes of all the precipitates studied. They lie parallel to the conjugate direction of the OR, $[1\bar{1}0]_\gamma$. Labeled $M1$ in Figure 2.3a, the weak features are spaced uniformly with an average separation of 7.0 nm. The strong features (labeled $M2$ in Figure 2.3b) also lie parallel to $[1\bar{1}0]_\gamma$, but they only occur in some regions of the interphase boundary. The spacing of the strong features is somewhat less regular than that of the weak features and averages 23.5 nm. Neither the weak nor the strong

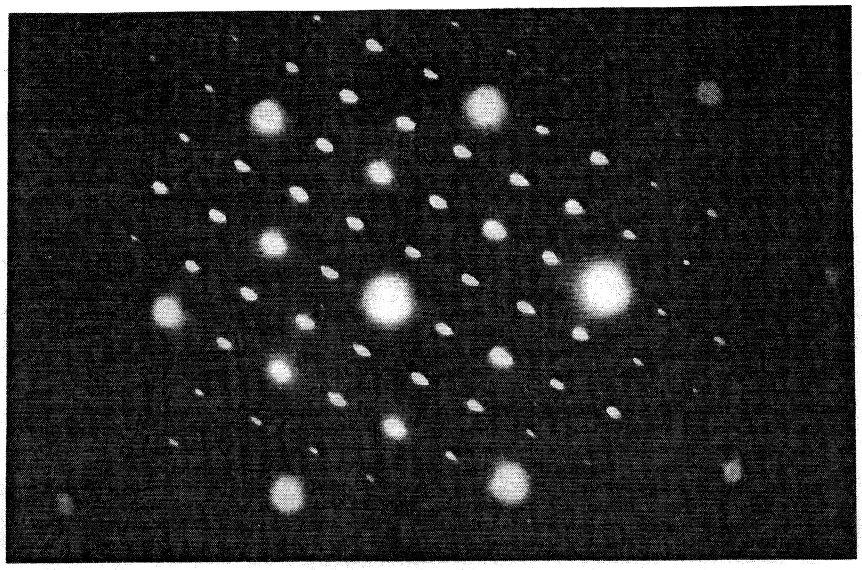


(a)

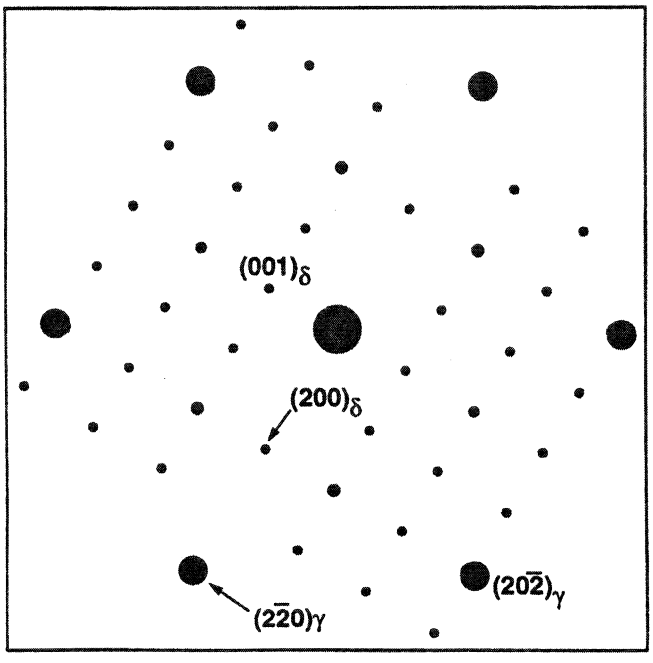


(b)

Figure 2.1: (a) Selected area diffraction patterns of a δ precipitate plate and the adjacent γ matrix and (b) indices corresponding to (a). The parallel zone axes of the patterns are $[1\bar{1}0]_\gamma$ and $[100]_\delta$



(a)



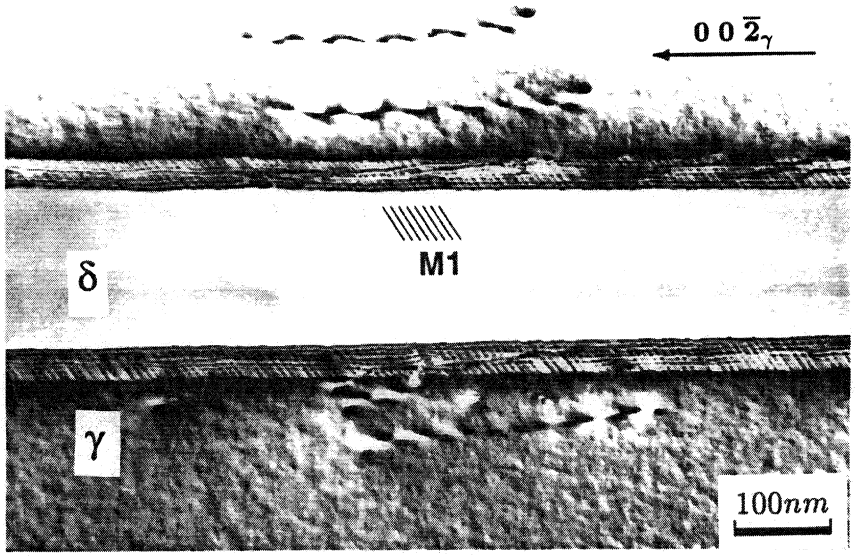
(b)

Figure 2.2: (a) Selected area diffraction patterns of a δ precipitate plate and the adjacent γ matrix and (b) indices corresponding to (a). The parallel zone axes of the patterns are $[111]_\gamma$ and $[020]_\delta$

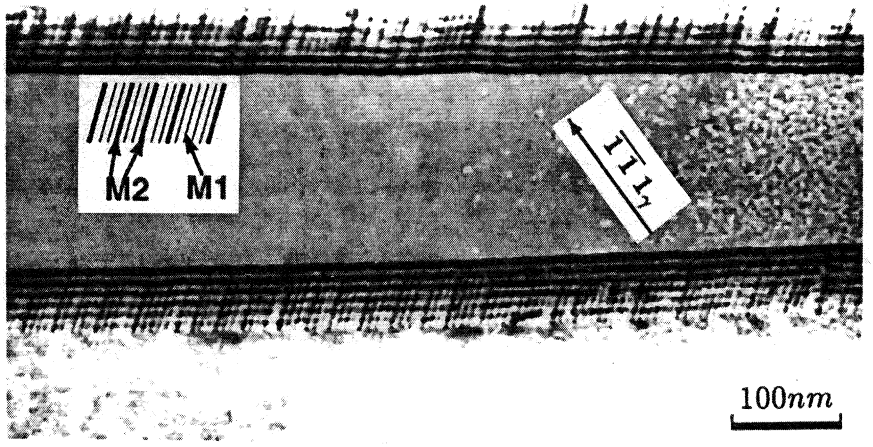
features step the interphase boundary as can be seen in the edge-on view of the habit plane in Figure 2.3c, so they are identified as misfit dislocations rather than ledges. Contrast experiments using the $\mathbf{g} \cdot \mathbf{b} = 0$ invisibility criterion were used to identify the weak features as misfit dislocations with a $[1\,1\,\bar{2}]_\gamma$ Burgers vector (Figure 2.4) and the strong features as misfit dislocations with $\mathbf{b} = [2\,\bar{1}\,\bar{1}]_\gamma$ (Figure 2.5).

Interfacial features identified as ledges are shown in Figure 2.6. The ledges exhibit strong contrast, are irregularly spaced, have a curved or wavy appearance, and do not seem to have a preferred orientation. They appear most frequently near the ends of the δ plates where the interphase boundary is not exactly planar. The ledges also could be distinguished from dislocations by the way they displace thickness fringes in the interface (Figure 2.6b) and misfit dislocation contrast (Figure 2.6c and d). Similar characteristics have been reported for ledges on precipitate plates in Zr–2.5wt %Nb [67], Cu–Zn [68], and Al–4wt %Cu [60]. The effective Burgers vector of the ledges varies from ledge to ledge on a single δ plate. Some have a Burgers vector lying in the habit plane ($[10\bar{1}]_\gamma$, $[\bar{1}10]_\gamma$, or $[01\bar{1}]_\gamma$) and others have a Burgers vector lying out of the habit plane ($[111]_\gamma$ or $[\bar{1}11]_\gamma$).

Several planar arrays of glissile dislocations are generally found in the γ matrix close to the precipitates. The dislocations in the most common array have a $[\bar{1}01]_\gamma$ Burgers vector, and the array lies parallel to the $(111)_\gamma$ habit plane a distance of a hundred nanometers or so from the δ precipitate (labeled *A* in Figure 2.7). The remaining dislocation arrays lie at an angle to the precipitate habit plane. Of these, the most common are $[\bar{1}01]_\gamma$ dislocations lying in the $(1\bar{1}1)_\gamma$ plane (labeled *B* in Figure 2.8). Less common are $[0\bar{1}1]_\gamma$ lying in the $(111)_\gamma$ plane (labeled *C* in Figure 2.8).



(a)

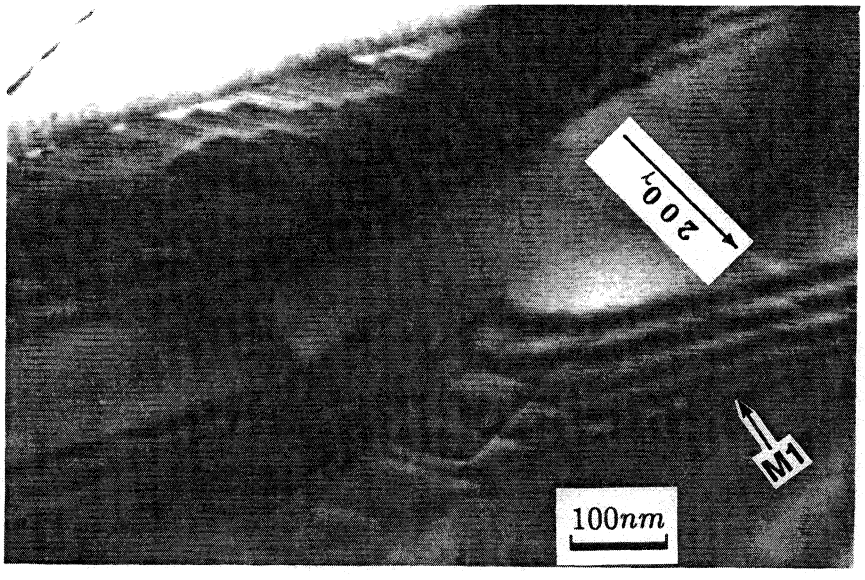


(b)

Figure 2.3: Two-beam bright field micrographs of the $(111)_\gamma$ habit plane using (a) $\mathbf{g} = 00\bar{2}_\gamma$ and (b) $\mathbf{g} = \bar{1}\bar{1}1_\gamma$. (c) Edge-on view ($[1\bar{1}0]_\gamma$ zone axis) of the same interface. "M1" and "M2" represents weak and strong contrast misfit dislocation, respectively.

(c)

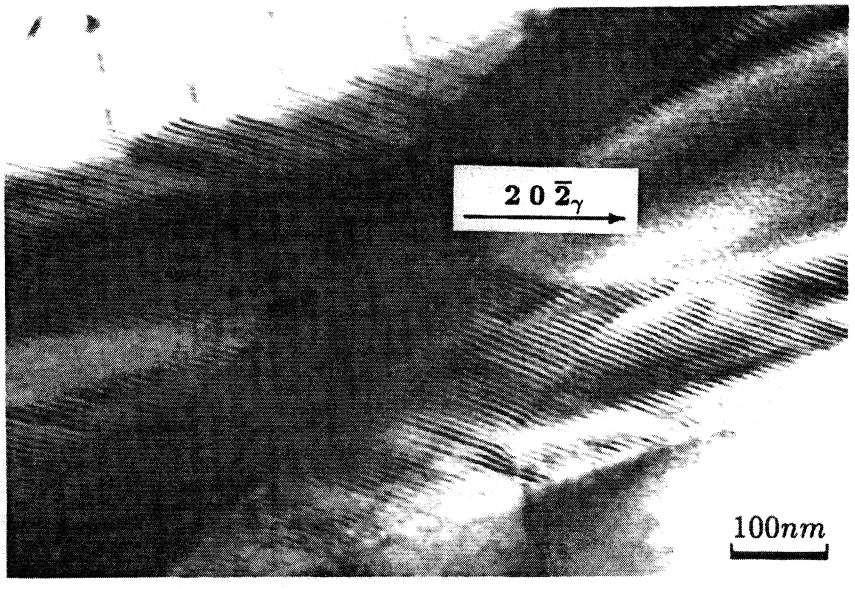
Figure 2.3 cont.



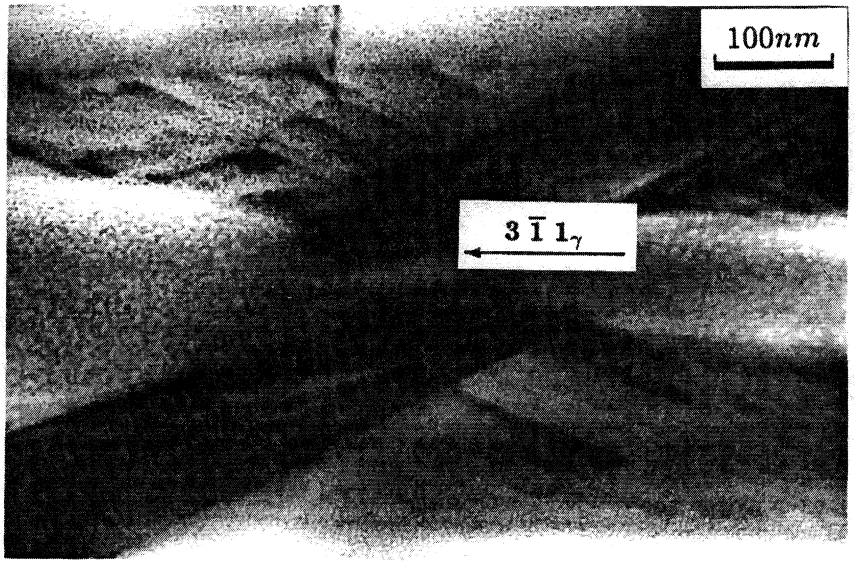
(a)

(b)

Figure 2.4: Two-beam bright field micrographs of $M1$-type misfit dislocations in the $(111)_\gamma$ habit plane using (a) $\mathbf{g} = 200_\gamma$, (b) $\mathbf{g} = \bar{2}0\bar{2}_\gamma$, (c) $\mathbf{g} = 20\bar{2}_\gamma$, (d) $\mathbf{g} = 3\bar{1}1_\gamma$; the dislocations are invisible in (d).

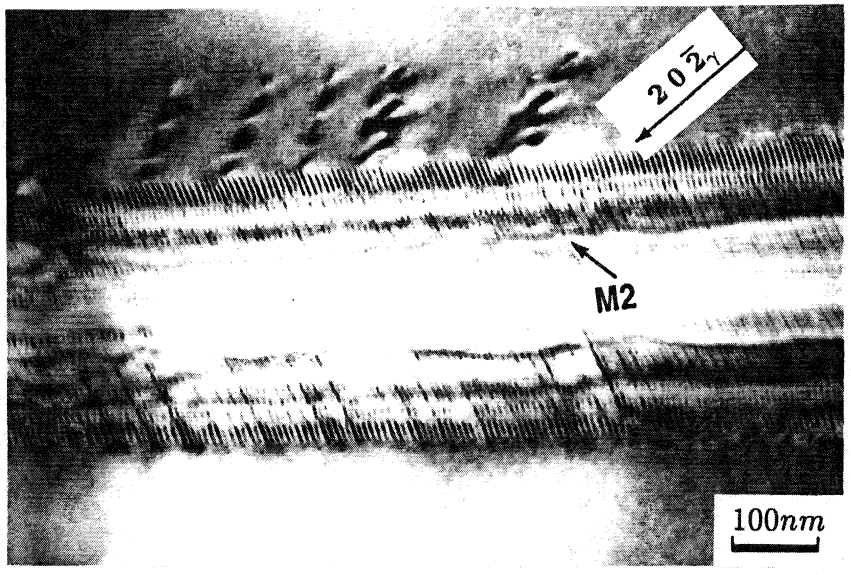


(c)

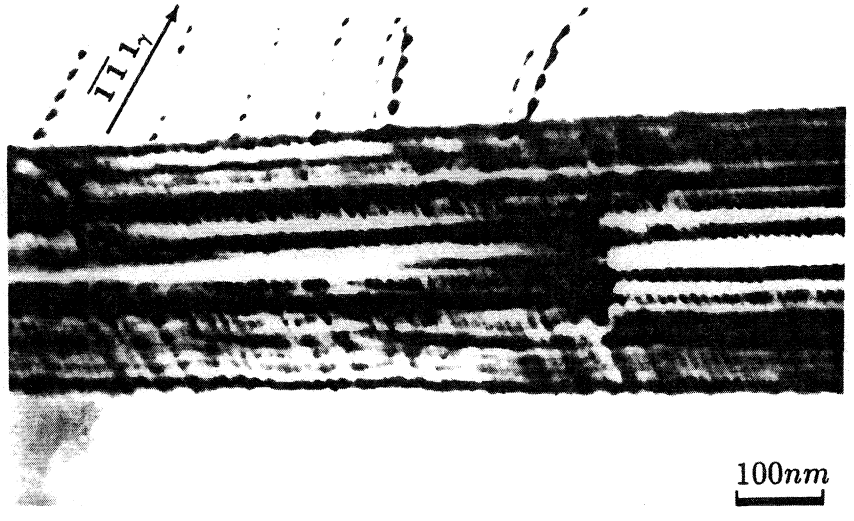


(d)

Figure 2.4 cont.

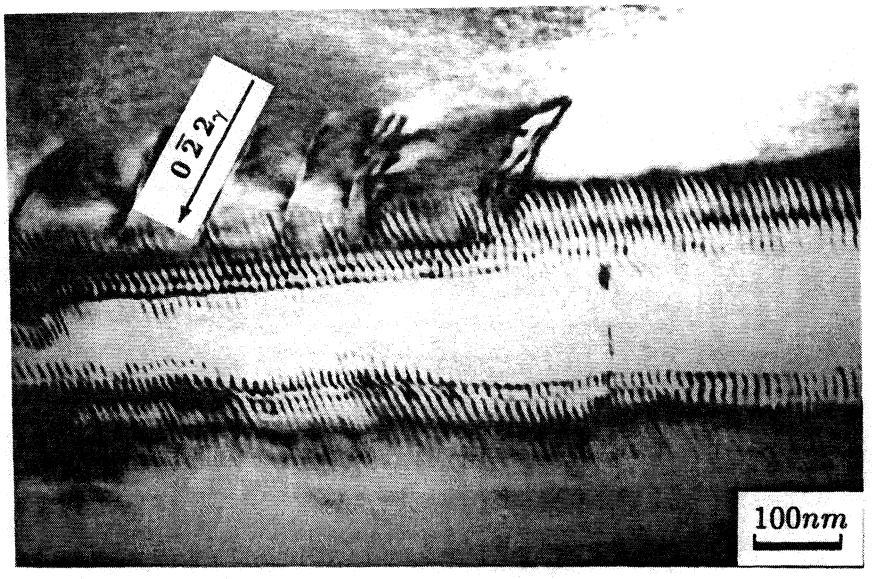


(a)



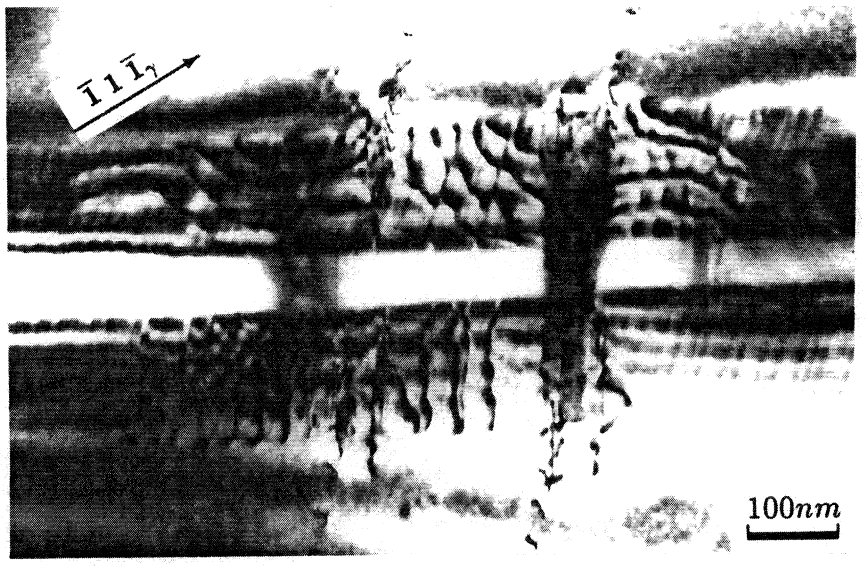
(b)

Figure 2.5: Two-beam bright field micrographs for the $M2$ type interfacial misfit dislocations of the $(111)_\gamma$ habit plane using (a) $\mathbf{g} = 20\bar{2}_\gamma$, (b) $\mathbf{g} = \bar{1}\bar{1}1_\gamma$, and (c) $\mathbf{g} = 0\bar{2}2_\gamma$; the dislocations in the upper right portion of the interface in (c) are invisible (the remaining dislocation contrast is from $M1$ dislocations).

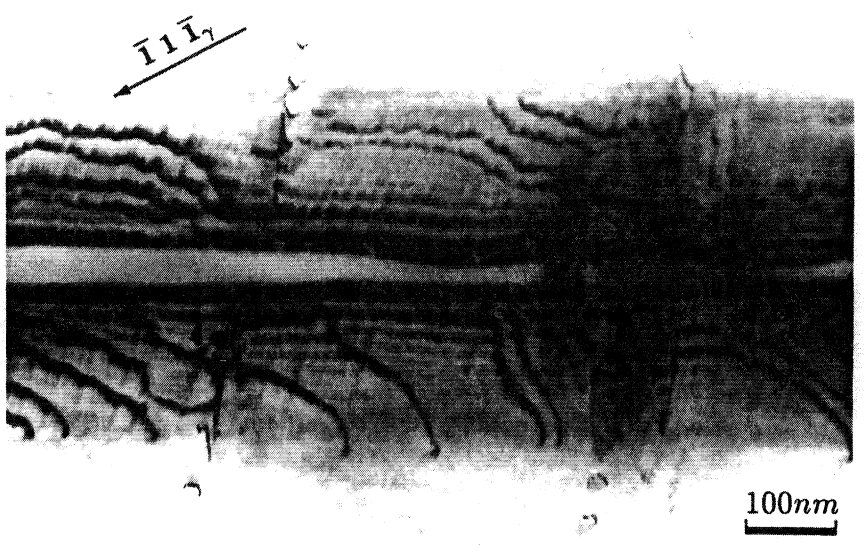


(c)

Figure 2.5 cont.

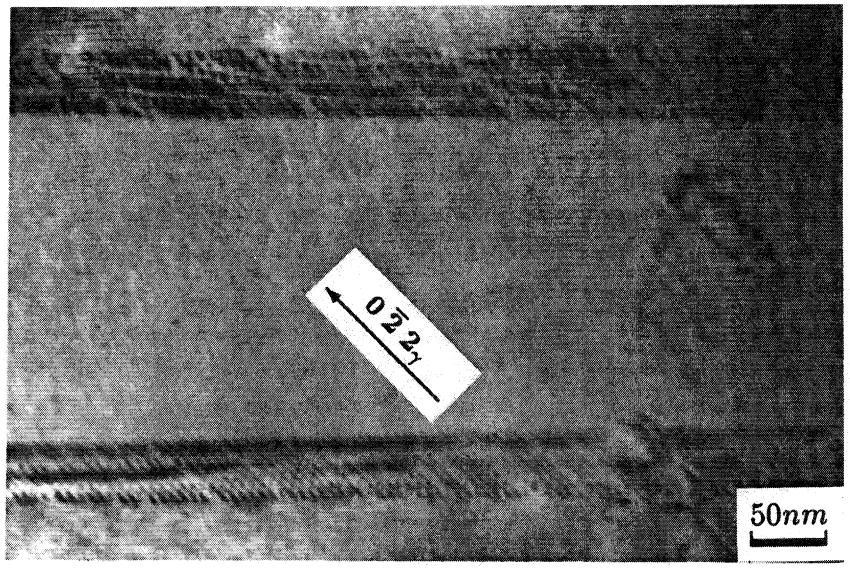


(a)

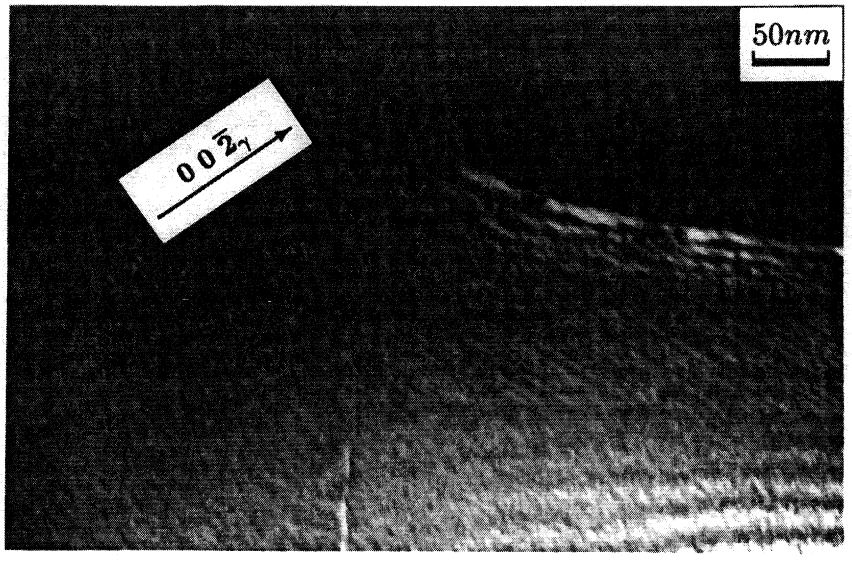


(b)

Figure 2.6: Two-beam bright field micrographs for the ledges in the interface using (a) $\mathbf{g} = \bar{1}1\bar{1}_\gamma$, (b) $\mathbf{g} = \bar{1}1\bar{1}_\gamma$ and (c) $\mathbf{g} = 0\bar{2}2_\gamma$. (d) WBDF micrograph using $\mathbf{g} = 00\bar{2}_\gamma$.

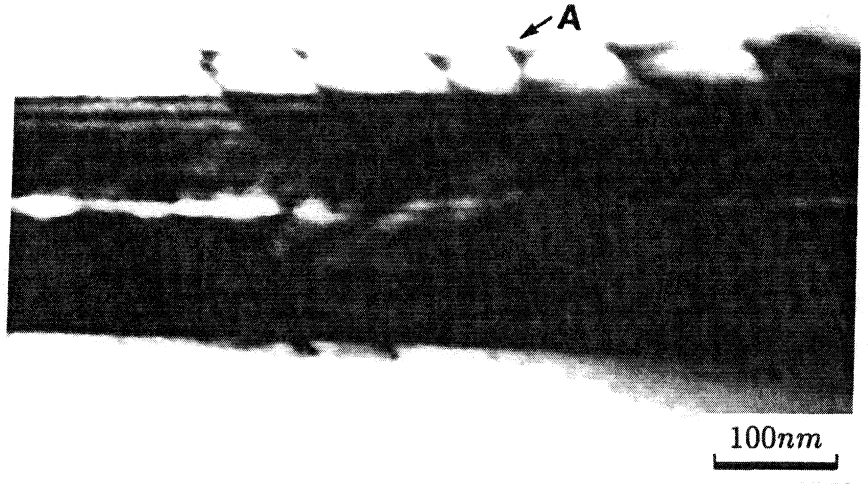


(c)

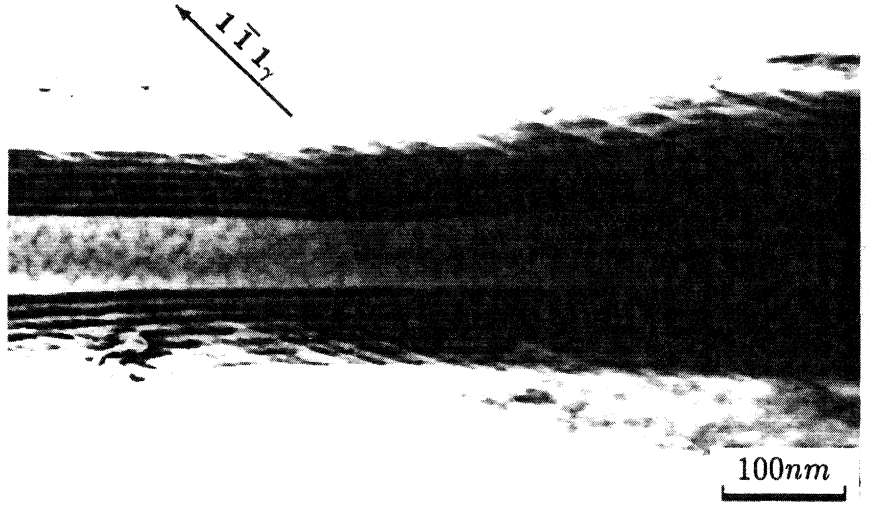


(d)

Figure 2.6 cont.

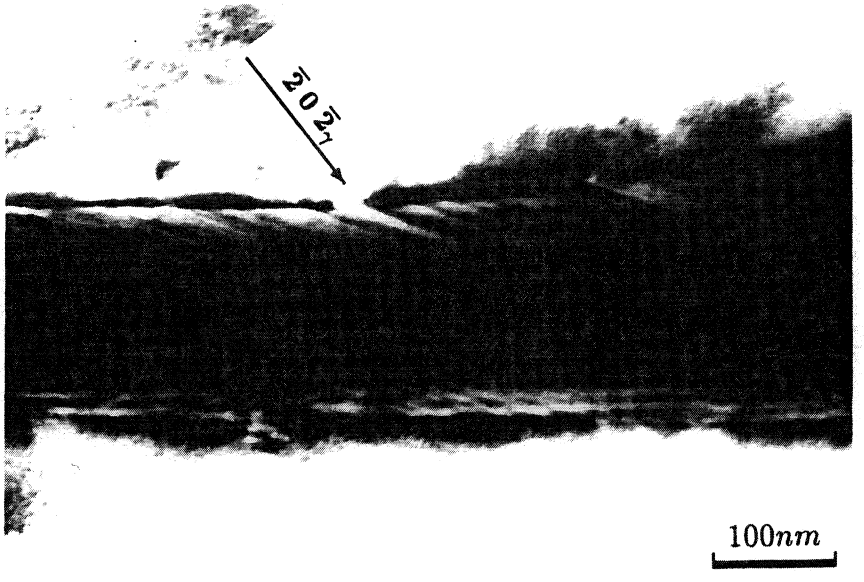


(a)



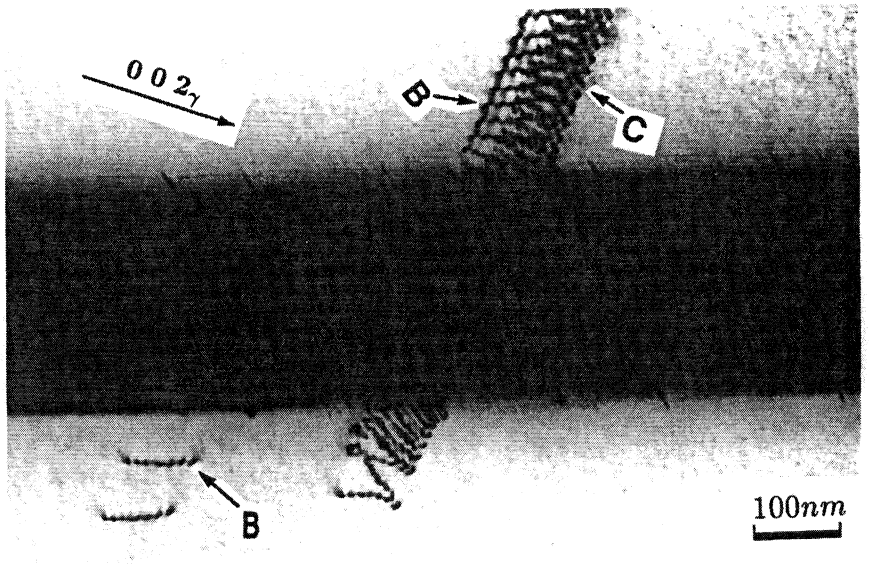
(b)

Figure 2.7: Dislocations in the matrix adjacent to a δ plate; (a) bright field, (b) two-beam bright field using $\mathbf{g} = 1\bar{1}1_\gamma$, (c) two-beam bright field using $\mathbf{g} = \bar{2}0\bar{2}_\gamma$. Dislocations labeled A are parallel to the precipitate's habit plane and have a $[\bar{1}01]_\gamma$ Burgers vector.

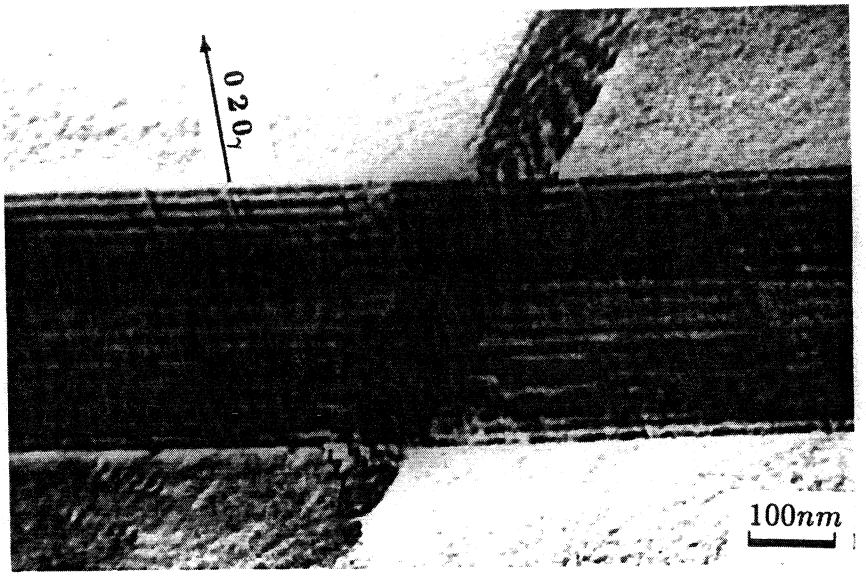


(c)

Figure 2.7 cont.

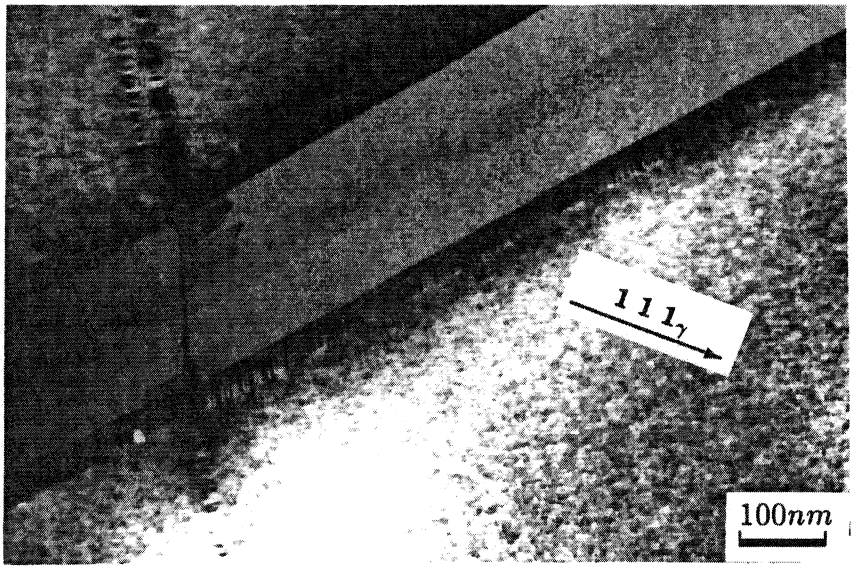


(a)

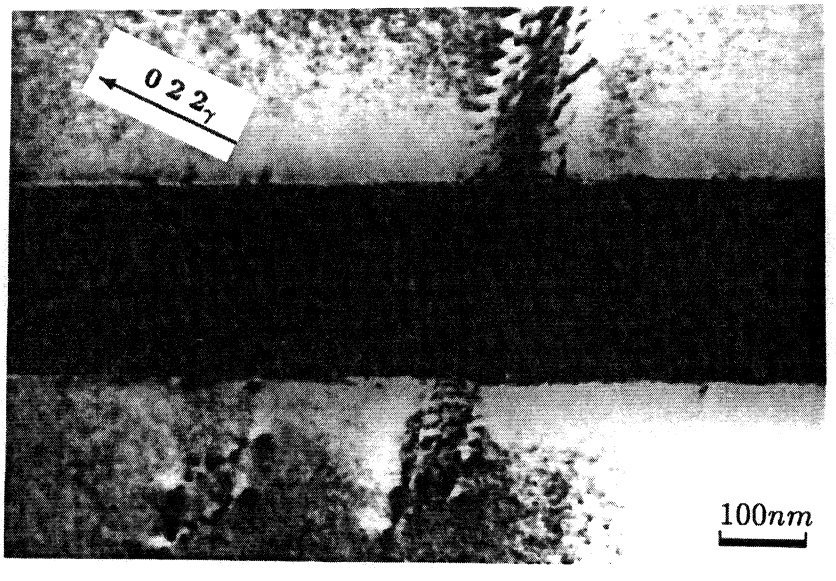


(b)

Figure 2.8: Two-beam bright field micrographs for the nearby dislocations in the matrix using (a) $\mathbf{g} = 002_\gamma$. (b) $\mathbf{g} = 020_\gamma$. (c) $\mathbf{g} = 111_\gamma$. (d) $\mathbf{g} = 022_\gamma$. Dislocations labeled B and C lie in $(1\bar{1}1)_\gamma$ and $(1\bar{1}1)_\gamma$ planes, respectively.



(c)



(d)

Figure 2.8 cont.

2.4 Discussion

There are some common features between the interfacial structure and crystallography of δ plates in Inconel 718 and precipitates in similar systems. Nakagawa and Weatherly [69] observed two orientation relationships between Ni_3Nb and a Ni_3Al matrix, one of which is the same as the OR for δ plates in Inconel 718. For this OR, however, they reported two arrays of orthogonal dislocations in the interface. Each set was identified as an array misfit dislocations in an edge orientation. Garmong and Rhodes [70] reported similar interfacial structure, as well as ledges and boundary facets. Annarumma and Turpin [71] investigated the interfacial structure of lamella in a directionally solidified Ni/Ni_3Nb eutectic, and also observed misfit dislocations, but the array was less regular than that seen here.

Interfacial misfit can be interpreted in one of two ways depending upon whether a lattice correspondence does or does not accompany δ precipitation. If a lattice correspondence is present, the misfit in the interface can be calculated from the homogenous deformation associated with the correspondence [72]. The misfit is

$$\mathbf{Misfit} = (\mathbf{A} - \mathbf{I})\mathbf{X} \qquad (\mathbf{I} = \text{Identity Matrix}) \tag{2.1}$$

where $\mathbf{A}$ is the transformation matrix and $\mathbf{X}$ is the direction in the lattice for which the misfit it is sought. If the following lattice correspondence is assumed: $(1,-1,0)_\gamma \rightarrow (1,0,0)_\delta$; $(0,0,2)_\gamma \rightarrow (0,1,1)_\delta$; and $(-1/2,-1/2,1)_\gamma \rightarrow (0,0,1)_\delta$, the misfit strain along $[1\bar{1}0]_\gamma$ and $[11\bar{2}]_\gamma$ (the smallest and largest misfitting directions in the habit plane) are 0.27% and 2.29%, respectively. In order to accommodate this misfit with the observed $[11\bar{2}]_\gamma$ and $[2\bar{1}\bar{1}]_\gamma$ misfit dislocations, the strong contrast and weak contrast misfit dislocations would need to be spaced 47.3 nm and 6.9 nm, respectively[2]. These spacings do not agree well with

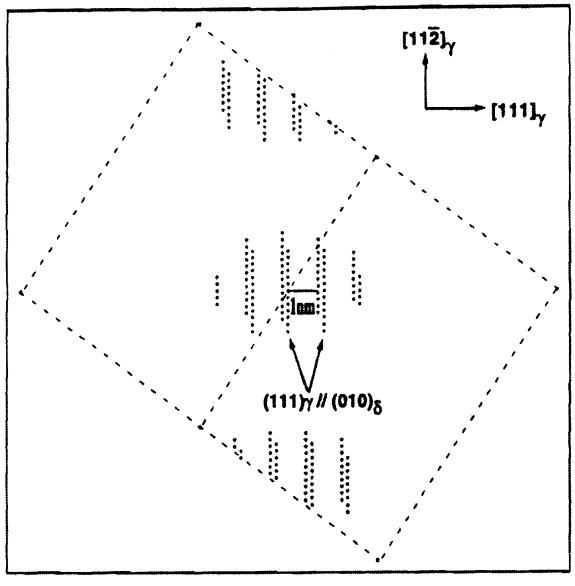
[2]These spacings are calculated using an assumed magnitude of $a/\sqrt{6}$ for the two types of misfit disloca-

the experimental values. In addition, the growth ledges should all have the same effective Burgers vector if a lattice correspondence is operative; a variety of Burgers vectors were found on a single δ plate. These discrepancies suggest a lattice correspondence is not present between the δ precipitate and the parent fcc phase in Inconel 718.

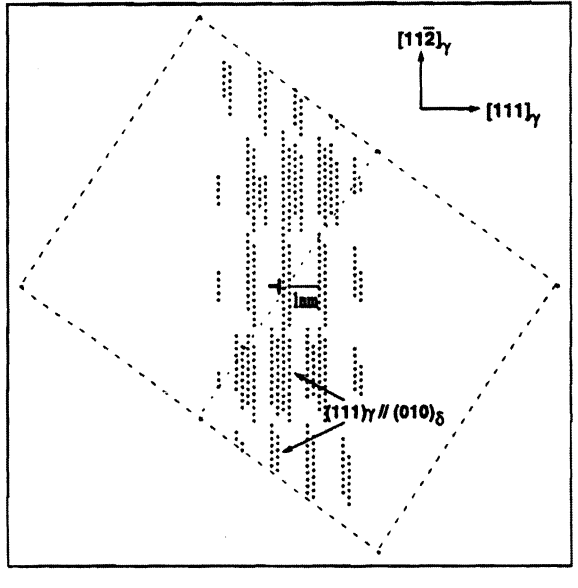
In the absence of a lattice correspondence, the interfacial structure can be interpreted in terms of geometric matching of atoms across the interphase boundary. This is done in an accompanying paper [45] using a near-coincident site approach to calculate the regions where the atoms in the fcc matrix and the δ precipitate match well. Figure 3.6 (a) shows a projection of the near-coincident sites projected on the $(1\bar{1}0)_\gamma$ plane. In this figure, the points represent places where atom positions in the two phases almost coincide. The near-coincident sites cluster in small patches of adjacent atoms that lie in the $(111)_\gamma$ plane. The spacing between the adjacent vertical row of near coincident sites is 0.21 nm, and the width of a cluster along the $[11\bar{2}]_\gamma$ direction is 3.4 nm. The vacant regions between the clusters are places where the atoms of the two crystals do not match well, so the degree of matching can be characterized by the continuity or size of the near-coincident patches.

The degree of matching in the $(111)_\gamma$ habit plane can be improved by introducing misfit dislocations. Near-coincident site calculations were carried out with dislocations of various magnitudes (all of whose Burgers vectors were consistent with the experimentally observed strong and weak contrast dislocations) placed at a variety of locations in the $(111)_\gamma$ plane. The greatest continuity of near-coincident sites was obtained with $\mathbf{b}= \frac{1}{6}[11\bar{2}]_\gamma$ dislocations (Figure 3.6(b)). These dislocations increase the width of the near-coincident site cluster in the $(111)_\gamma$ plane from 3.4 to 9.9 nm. The dislocation Burgers vector and spacing of the simulated matching of Figure 3.6(b) agree with those of the $M1$ dislocations

tions. The relation [73] $D_d = b_{\text{eff}}/\text{Misfit}$ where D_d is the dislocation spacing and b_{eff} is the misfit dislocation Burgers vector resolved in the misfit direction.



(a)



(b)

Figure 2.9: $(1\bar{1}0)_\gamma$ projections of the γ/δ near-coincident sites with the origin (a coincident site) located at the left end of the scale markers; (a) contains no misfits dislocations, (b) and (c) have misfit dislocations located at the dislocation symbols with **b**= $\frac{1}{6}[11\bar{2}]_\gamma$ and **b**= $\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ Burgers vectors, respectively.

shown in Figures 2.3 and 2.4. Including a b= $\frac{1}{6}[2\overline{1}\overline{1}]_\gamma$ misfit dislocation (an $M2$, or strong contrast dislocation) in the matching simulation, Figure 3.6(c), increased the size of the near-coincident site cluster, particularly in the $[111]_\gamma$ direction, but it did not significantly improve the matching in the $(111)_\gamma$ habit plane.

High-resolution electron microscopy observations of the habit plane provide further insight into the nature of the matching in the interface. First, the HREM micrograph in Figure 2.10(a) shows the habit plane is not atomically flat. The location of the interface, which was determined by following lattice planes from γ and δ to the interface region, steps up and down to adjacent $(111)_\gamma$ planes. Two such steps in a small portion of the interface are shown within the circuit drawn in Figure 2.10(a) and the path of the interface over a region of about 53 nm is shown schematically in Figure 2.10(b). The height of the steps are 0.21 nm, and they do not have a net dislocation character[3]. Figures 3.6(a) and (c) suggest the steps lie within patches of near-coincident sites or connect adjacent patches. Since the atomic matching along such a step is good, they are expected to contribute little to the overall energy of the interphase boundary.

To attempt to verify the Burgers vector of the misfit dislocations, a Burgers circuit is drawn in Figure 2.11 to include a section of the interface boundary extending from one region of good lattice matching to another. Based on conventional TEM observations, this section of the boundary includes both weak and strong contrast misfit dislocations. The interface in Figure 2.11 is viewed along the $[1\overline{1}0]_\gamma$ direction, which is parallel to the line direction of the misfit dislocations. The circuit has a projected closure fault with respect to the fcc γ lattice of 0.22 nm along the $[11\overline{2}]_\gamma$ direction. This agrees reasonably well with the combined projected closure fault of one $\frac{1}{6}[11\overline{2}]_\gamma$ misfit dislocation (weak contrast) and one

[3]A Burgers circuit on a micrograph that includes a portion of the interphase boundary can detect unit dislocations with respect to the bulk phases, but not partial or DSC type dislocations

$\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ misfit dislocation (strong contrast) in the $[11\bar{2}]_\gamma$ direction: $0.147+0.074 = 0.22nm$. The distance in the habit plane from one point of good matching to the next (the width of the circuit in Figure 2.11) is 6.7 nm and is consistent with that predicted by the matching simulation (6.6 nm) and measured spacing of the weak contrast misfit dislocations observed with conventional TEM (7 nm).

Comparison of the experimentally observed misfit dislocations and predicted spacings based on atomic matching (near-coincident sites) and on a lattice correspondence approach are summarized in Table 2.2. Misfit dislocation spacings based upon atomic matching agree more closely with the experimental values than those obtained using the assumed lattice correspondence.

The source of misfit dislocations can be explained as follows: the interphase boundary nucleates a dislocation loop with the Burgers vector of $\frac{1}{2}[10\bar{1}]_\gamma$ (positive) and $\frac{1}{2}[\bar{1}01]_\gamma$ (negative) [74]. The former one can dissociate into $M1$ and $M2$ types:

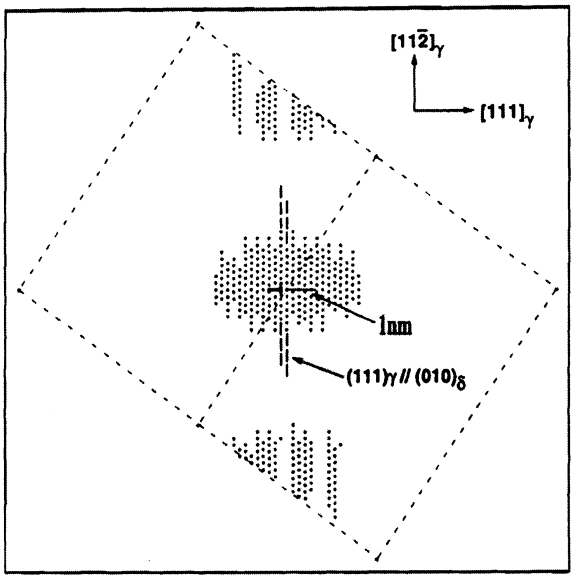
$$\mathbf{b_{matrix}} = \mathbf{b_{M1}} + \mathbf{b_{M2}}$$

$$\frac{1}{2}[10\bar{1}]_\gamma = \frac{1}{6}[11\bar{2}]_\gamma + \frac{1}{6}[2\bar{1}\bar{1}]_\gamma$$

This reaction would produce even numbers $M1$ and $M2$. At the same time, the latter one, $\frac{1}{2}[\bar{1}01]_\gamma$, would be ejected into the matrix, and these dislocations should be close to the precipitate. This is in agreement with experimental observation that most common nearby dislocations in the matrix, type A and B (in Figure 2.7 and Figure 2.8), were with the Burgers vector of $\frac{1}{2}[\bar{1}01]_\gamma$.

Meanwhile, the interphase boundary would nucleate another dislocation loop with the Burgers vector of $\frac{1}{2}[01\bar{1}]_\gamma$ (positive) and $\frac{1}{2}[0\bar{1}1]_\gamma$ (negative). The former one may react with types A or B dislocations to create purely $M1$ type misfit dislocations:

$$\frac{1}{2}[10\bar{1}]_\gamma(A) + \frac{1}{2}[01\bar{1}]_\gamma(C) = 3(\frac{1}{6}[11\bar{2}]_\gamma)(3M1)$$

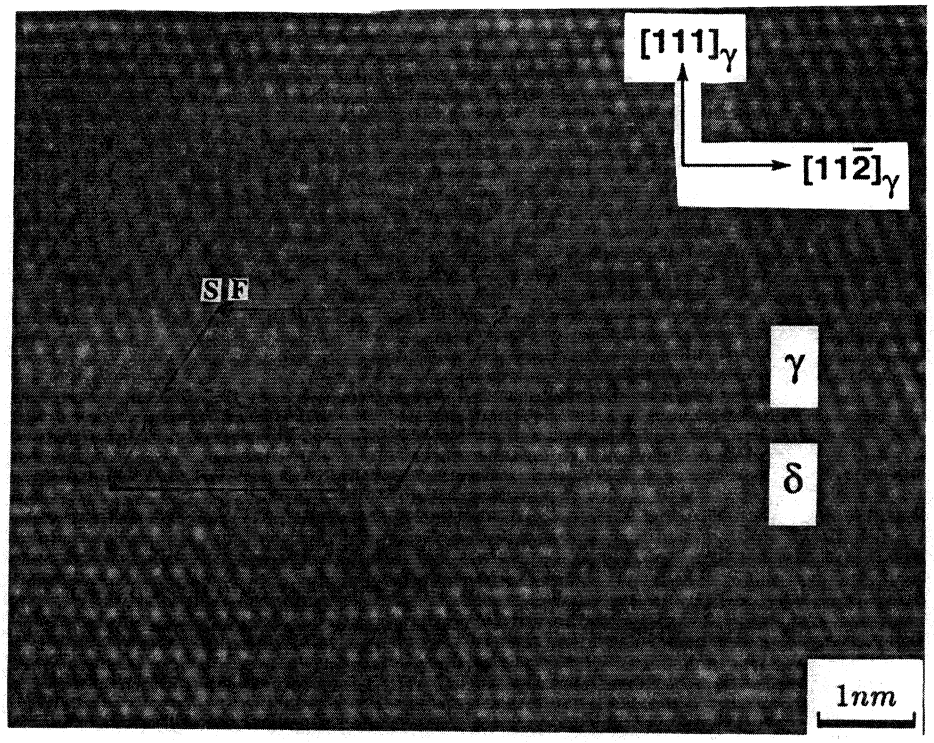


(c)

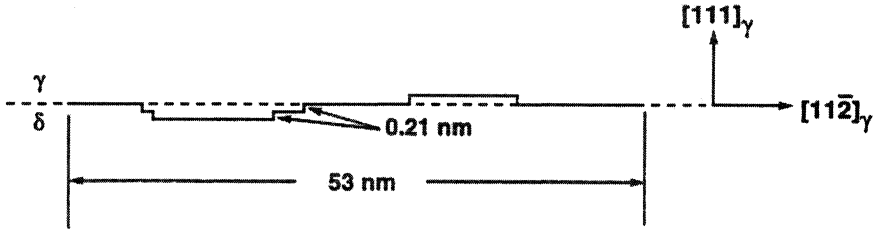
Figure 3.6 cont.

Table 2.2: Experimental results and theoretical approaches for the interfacial dislocations of γ/δ interface in Inconel 718

	Burgers Vector	Line Direction	Spacing (nm)
Experimental	(M1 type) $[11\bar{2}]_\gamma$	$[1\bar{1}0]_\gamma$	7.0
	(M2 type) $[2\bar{1}\bar{1}]_\gamma$	$[1\bar{1}0]_\gamma$	23.5
Lattice Correspondence	$\frac{1}{6}[11\bar{2}]_\gamma$	$[1\bar{1}0]_\gamma$	6.9
	$\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$	$[1\bar{1}0]_\gamma$	47.3
Atomic Matching	$\frac{1}{6}[11\bar{2}]_\gamma$	$[1\bar{1}0]_\gamma$	6.6
	$\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$	$[1\bar{1}0]_\gamma$	24.2



(a)



(b)

Figure 2.10: (a) High-resolution electron micrograph of the $(111)_\gamma$ habit plane. The fcc matrix is at the top, the δ precipitate is at the bottom, the location of the interphase boundary is indicated by the dashed line in the inset. (b) Schematic of the interphase boundary plane over a distance of about 53 nm.

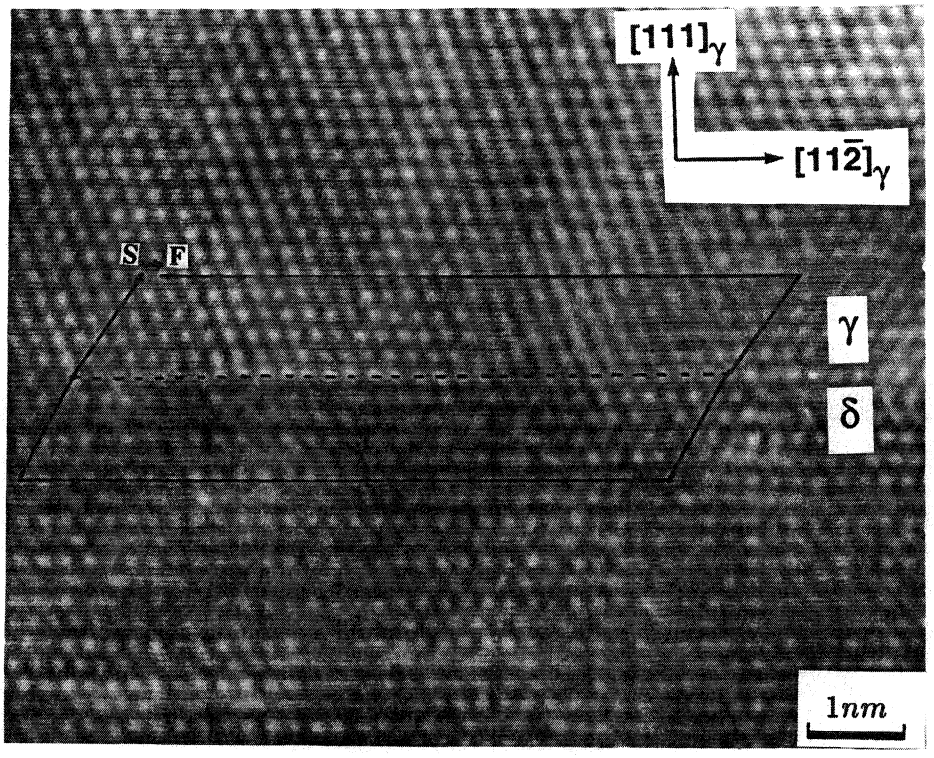


Figure 2.11: High-resolution electron micrograph of misfit dislocations in the $(111)_\gamma$ habit plane.

Another one, $\frac{1}{2}[0\bar{1}1]_\gamma$, would also be merged into the nearby matrix, and this is in agreement with the experimental observed C type dislocations near to the A or B type dislocations (in Figure 2.8).

The above reactions would make more numbers of $M1$ type dislocations than that of $M2$ type. This can be interpreted as $M1$ was observed everywhere in the interface by TEM. Meanwhile, the latter reaction would leave some region without $M2$ dislocations.

2.5 Conclusions

The interphase boundary structure and defects related to the growth of δ plates in Inconel 718 were characterized. The function of the defects were interpreted with the aid of atomic matching calculations based on near-coincidence sites. Despite the resulting plate morphology, a lattice correspondence does not appear to obtain during δ precipitation. The following specific conclusions were obtained:

1. The orientation relationship between the γ and δ phases is: $(111)_\gamma \parallel (010)_\delta$, $[1\bar{1}0]_\gamma \parallel [100]_\delta$. The habit plane of the δ plates is $(111)_\gamma$, which is the conjugate plane of the orientation relationship.

2. Two types of misfit dislocations are found in interface and both are parallel to the $[1\bar{1}0]_\gamma$ direction. One type has a $\frac{1}{6}[11\bar{2}]_\gamma$ Burgers vector and 7 nm spacing and is believed to accommodate the misfit within the conjugate plane, i.e., $(111)_\gamma \parallel (010)_\delta$. The other type has a $\frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ Burgers vector and 23.5 nm spacing but is only observed in some portions of the interface. These dislocations appear to accommodate misfit normal to the $(111)_\gamma$ plane and allow the habit plane to curve away from an exact $(111)_\gamma$ orientation.

3. Interfacial misfit dislocations are proposed to result from the dissociation or reaction of the dislocation loops nucleated on the interphase boundary.

4. Curved, irregularly spaced ledges are also present in the δ habit plane. The effective Burgers vector of the ledges is not unique.

5. Thickening of δ occurs by a ledge mechanism. Growth ledges provide a way of connecting good atomic matching patches. The interfacial structure is not consistent with the maintenance of a lattice correspondence during growth.

Chapter 3

Determining Interphase Boundary Orientations from Near Coincident Sites

3.1 Introduction

Approaches to modeling the interfacial structure of precipitate plates fall into two categories: those based on geometric matching and those derived from lattice deformation theories. These two approaches imply fundamentally different boundary migration mechanisms and have different implications for precipitate shape.

Lattice deformation approaches describe a phase change during precipitation by a mathematical transformation matrix. They assume a one-to-one correspondence between the parent and product lattices and the phase change is accomplished by a homogenous, or uniform, deformation of the parent lattice. The Phenomenological Theory of Martensite Crystallography (PTMC) [27], the invariant line theory [29, 30], and the O-lattice theory [28, 31, 32, 33, 75, 76, 72] fall into this category and are useful for rationalizing the shape and habit planes of some precipitatess. However, if the transformation mechanism does not involve a unique lattice correspondence or if the lattice correspondence is not known *a priori* [77, 78, 43], these methods are difficult to apply. This can be problematic for many precipitation reactions since the atomic displacements at the growth interface that are needed to identify a lattice correspondence are difficult to infer from experimental observations.

On the other hand, matching approaches are based upon the geometric arrangement

of atoms at a boundary with no consideration given to the atom displacements occurring when one crystal transforms to the other. The coincidence-site lattice (CSL) concept [28] falls into this category. Computer modeling of interfacial structure using this approach has been employed to investigate the influence of orientation relationship and lattice parameter ratio on the degree of matching at fcc/bcc boundaries [41, 79] and fcc/hcp boundaries [42]. These investigations identified regions of good matching associated with coherent atom pairs across the conjugate planes of the orientation relationship.

In this study, the computer matching approach is applied in 3-dimensions to investigate the factors that determine the conjugate plane, the best matching plane, and the habit plane. Two different systems, fcc/bcc in Ni–45 wt%Cr and fcc/orthorhombic in Inconel 718 were chosen for this purpose. The modeling is also used to explain the observed misfit compensating defects observed in the interphase boundaries of these systems.

3.2 Procedure

The procedure for determining the degree of matching between crystal pairs follows Bollmann [28]. The two crystal lattices are extend in space from a common origin (allow to interpenetrate) in a specified orientation relationship. Coincident sites are defined as points that lie on both lattices. The origin is one such point, and there are may be additional coincident sites when the two crystals are arranged in specific orientation relationships. Planes with a high density of coincident sites contain a higher proportion of atoms in "correct" positions with respect to the two lattices and these planes are thus good candidates for low-energy boundaries.

There are generally few coincident sites between two phases with different crystal structures and lattice parameters. However, many atoms in boundaries are *close* to atomic

sites with respect to both phases. We refer to these as near-coincident sites (NCS). Near-coincident sites are identified relative to atom positions in the two phases rather than lattice points, and they do not constitute a lattice of their own. Nevertheless, the density of such points have been used by numerous investigators [e.g. [41, 79, 42, 43]] to provide a measure of the matching in an interphase boundary.

Two specific cases of interphase boundary matching are considered here: interfaces between fcc and bcc phases in Ni–Cr [16, 48, 19, 69, 71] and interfaces between fcc and orthorhombic phases in Inconel 718 [44]. The corresponding orientation relationships and lattice parameters are shown in Table 3.1 [80, 49, 81, 82, 46, 69, 55]. These pairs of phases were chosen because their crystallography is well-characterized and they represent two interesting cases. For bcc precipitates in Ni–Cr, the parallel planes between the fcc and bcc phases (the conjugate planes) are not parallel to a prominent facet of the bcc precipitate, whereas the conjugate planes of fcc and δ (orthorhombic phase) in the Inconel 718 are parallel to the habit plane of the δ precipitate plates.

To locate near-coincident sites, a coincidence criterion must be adopted to identify when an atom in one phase is close enough to an atomic site in the other to be considered nearly-coincident. The choice the allowable distance between atomic sites in the two phases is arbitrary, but it influences the number of near-coincident sites in a region of the interpenetrating phases. A coincidence criterion of 15% of the interatomic distance in a closed-packed direction has been used previously [41, 42] and is used here as well.

The atomic sites of the fcc phase and either the bcc or orthorhombic phase were generated by computer in a cube roughly 30 fcc unit cells to a side. Fcc sites within the coincidence criterion of an atomic site in the overlapping phase (either bcc or orthorhombic) were identified as near-coincident sites (NCS).

3.3 Results

3.3.1 Fcc/Bcc

The near-coincident sites for fcc/bcc crystals in a Kurdjumov-Sachs orientation relationship are arranged in 3-dimensional clusters[43]. When viewed along the $[10\bar{1}]_f$ direction, these clusters appear as in Figure 3.1. The orientation of the parallel conjugate planes of the Kurdjumov-Sachs orientation relationship ($(1\bar{1}1)_f$ and $(101)_b$) is indicated by the position of the solid lines. The dashed lines represent the bounds of the cube-shaped region over which the fcc and bcc phases overlap ($30a_f \times 30a_f \times 30a_f$). Solid and dashed lines have the same meaning in subsequent figures.

The clusters of near-coincident points have several interesting characteristics. First, they consist of adjacent atomic sites, that is, all the atoms within a given cluster are near-coincident sites. It follows that the crystallographic plane with the greatest density of of near-coincident sites must be a closest-packed plane. This plane is the conjugate plane of the orientation relationship, and it is indicated by the three short solid lines in Figure 3.1. Second, near-coincidence in a conjugate plane are limited to a narrow region about 1.0 nm (in length) $\times$ 0.22 nm (in width); the clusters are extended much further by stepping along successive conjugate planes. These steps, which are the structural ledges of Rigsbee and Aaronson [41], give the cluster for this orientation relationship and lattice parameter ratio an average habit plane of $(1\ \bar{2}\ 1)_f$. The relatively high density of near-coincident sites in this plane was used to explain the orientation of the habit plane [43]. However, the density of near-coincident sites in the crystallographic $(1\ \bar{2}\ 1)_f$ plane is not as high as it is in the conjugate planes within a cluster; the cluster merely extends further along an average $(1\ \bar{2}\ 1)_f$ plane than it does along the conjugate plane.

The separation of adjacent near-coincident site clusters is approximately 1 nm, roughly

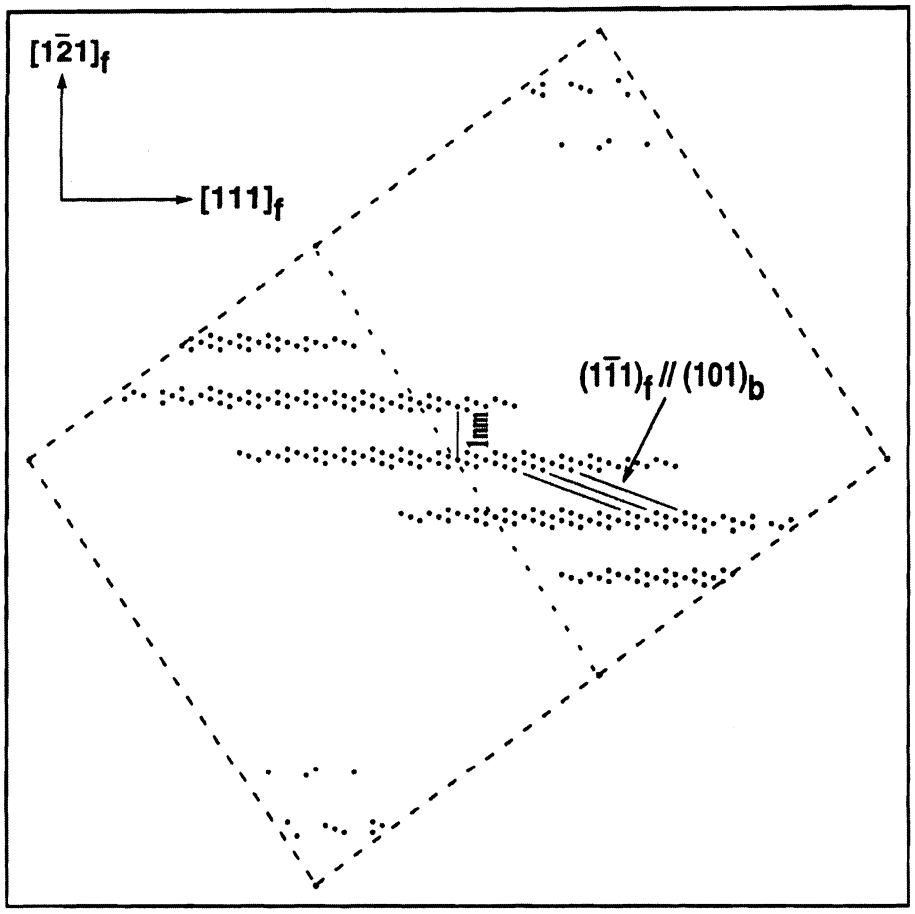
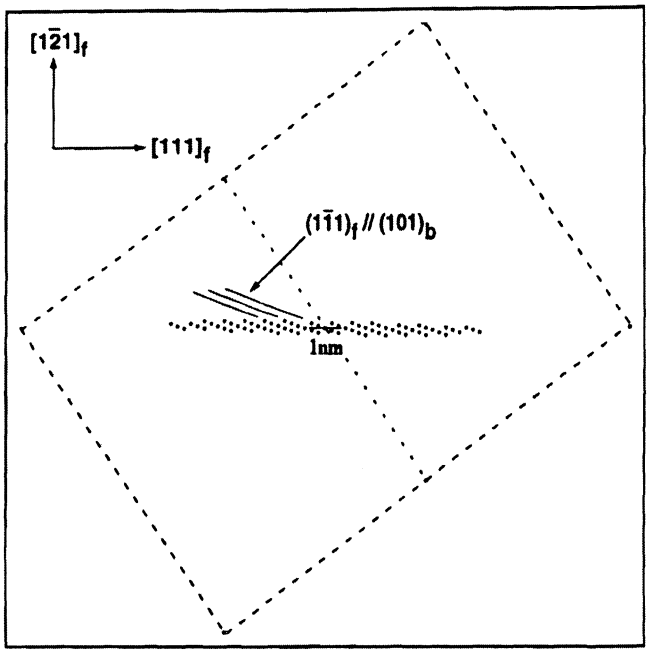


Figure 3.1: A projection of the fcc/bcc near-coincident sites onto the $(10\bar{1})_f$ plane.

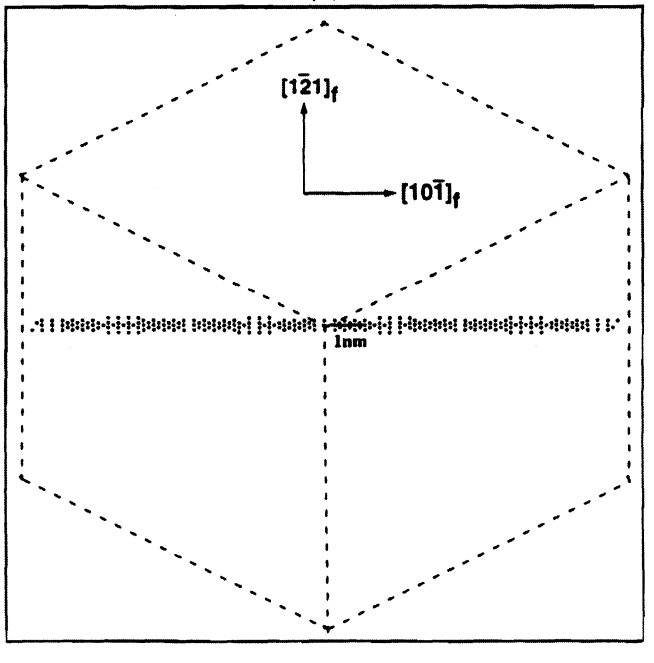
the height of growth ledges in the habit plane of bcc precipitates in Ni–Cr [83]. This spacing as well as the relative position of the adjacent clusters is insensitive to the coincidence criterion. For a deviation from exact coincidence of 35% of the interatomic distance, the clusters extend somewhat further in the $(1\ \bar{2}\ 1)_f$ plane and thicken slightly, but they do not link up. This reflects the fact that the misfit is large between adjacent clusters. In fact, experimental studies show a misfit dislocation is generally present in growth ledges on the $(1\ \bar{2}\ 1)_f$ plane [48, 19].

A single cluster of near-coincident sites is shown in Figure 3.2 viewed along several different directions. Figure 3.2(a) shows the cluster viewed along $[10\bar{1}]_f$, close to the direction with the least amount of misfit strain. This direction is referred to here as the best-matching direction. Figure 3.2(b) and (c) shows the cluster viewed along the $[111]_f$ and $[1\bar{2}1]_f$ directions, respectively; both are 90° to view in (a).

As intuition suggests, the cluster of near-coincident sites extends the furthest and is continuous in the best-matching direction. The broad, flat shape of the cluster indicates the average $(1\ \bar{2}\ 1)_f$ plane is good matching over a width of about 9.2 nm. Some compensation of misfit is evidently needed to extend the near-coincident regions the precipitate interface beyond this width in the $(1\ \bar{2}\ 1)_f$ plane. Two misfit compensating mechanisms were explored in the matching calculations: elastic strain and partial dislocations associated with stacking fault. Elastic strain was incorporated in the model by increasing the spacing of the $(101)_b$ planes to match that of the $(1\bar{1}1)_f$ planes; this crudely approximates the presence of an elastic strain that matches the spacing of the conjugate planes. The resulting near-coincident sites are shown in Figure 3.3. The elastic distortion increases the length of the near-coincident site cluster about 20% and rotates the plane of the cluster by 5.2° about the $[10\bar{1}]_f$ direction. However, the near-coincident site cluster still did not advance to the ends of the simulation block.



(a)

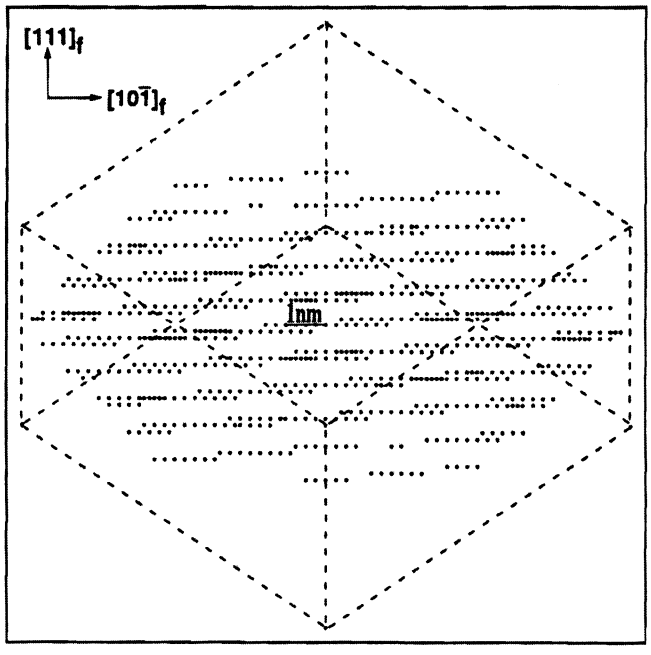


(b)

Figure 3.2: A projection of single cluster of fcc/bcc near-coincident sites onto the (a) $(10\bar{1})_f$ plane, the (b) $(111)_f$ plane, (c) the $(1\bar{2}1)_f$ plane.

Table 3.1: Necessary data of fcc : bcc and fcc : orthorhombic systems for computer modeling

System	Matrix : Precipitate	Conjugate plane	Conjugate direction	Lattice parameters(nm)
Ni–Cr	fcc : bcc	$(1\bar{1}1)_f \parallel (101)_b$	$[\bar{1}01]_f \parallel [\bar{1}\bar{1}1]_b$	a_f=0.3664, a_b=0.2920
Inconel 718	fcc : orthorhombic	$(111)_\gamma \parallel (010)_\delta$	$[1\bar{1}0]_\gamma \parallel [100]_\delta$	a_γ=0.3616, a_δ=0.5141 b_δ=0.4231 c_δ=0.4534



(c)

Figure 3.2 cont.

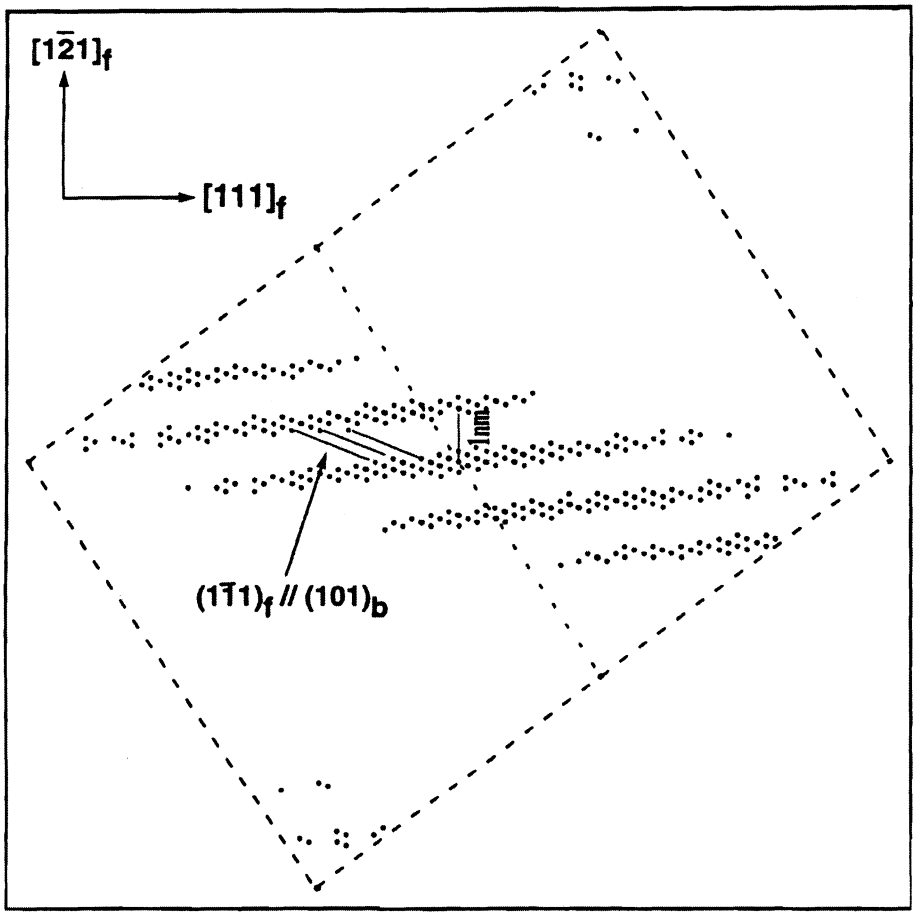


Figure 3.3: A projection of the fcc/bcc NCS onto the $(10\bar{1})_f$ plane with elastic strain included to improve interfacial matching.

The only defect observed experimentally in the $(1\ \bar{2}\ 1)_f$ habit plane of the precipitates in Ni–Cr are partial dislocations associated with stacking faults extending into the matrix from the precipitate [83, 19]. These dislocations could accommodate some misfit in the interface. The Burgers vector of the partial dislocations was suggested to be either $\frac{1}{6}[11\bar{2}]_f$ or $\frac{1}{6}[\bar{2}11]_f$ and the dislocations had a typical spacing of 11 nm. The influence of these dislocations on the geometric matching in the interface was investigated by shifting the fcc atoms to the right of the dislocation and above the $(1\ \bar{2}\ 1)_f$ plane by $\frac{+\mathbf{b}}{2}$, and by $\frac{-\mathbf{b}}{2}$ for atoms to the left of the dislocation. While in this case, the simulation block is with the size of. The resulting distribution of near-coincident sites is shown in Figure 3.4 for a $\frac{1}{6}[11\bar{2}]_f$ dislocation located at the origin of a $50a_f \times 50a_f \times 50a_f$ simulation block. Similar results were also obtained for a $\frac{1}{6}[\bar{2}11]_f$ Burgers vector.

It was found the cluster of near-coicident sites extended 18.4 nm along the $(1\ \bar{2}\ 1)_f$ plane, compared with 9.2 nm when no partial dislocation was present (Figure 3.1). In other words, boundary matching is improved considerably along the $[111]_f$ and $[1\bar{2}1]_f$ directions by stacking faults in the $(111)_f$ plane. Somewhat surprisingly, matching along the best matching direction, $[10\bar{1}]_f$, is not changed significanly by the stacking faults. The partial dislocations associated with the stacking faults improve matching in the $[111]_f$ and $[1\bar{2}1]_f$ directions, and the extent of the cluster of adjacent near-coincident sites increases to the bounds of simulation block. It is important to point out that the improvement in matching does not depend solely upon the Burgers vector of partial dislocation; the location of the dislocation is important and not always intuitively obvious. For example, one might expect that a misfit dislocation should appear in the regions where the matching begins to deteriorate. However, improved matching in Figure 3.4 resulted from a dislocation located at the origin — a place where the matching was originally good (Figure 3.1).

Figure 3.5 shows the effect on the near-coincident site clusters of both elastic strain and

misfit accommodation associated with a stacking fault. This combination produced the best matching obtained in the simulations.

3.3.2 γ/δ (Fcc/Orthorhombic)

As in the fcc/bcc case, the near-coincident sites for γ/δ are arranged in clusters. The best matching direction, or the direction with the longest continuous row of near-coincident sites is the $[1\bar{1}0]_\gamma$ direction. When viewed along this direction the clusters of adjacent near-coincident sites appear as in Figure 3.6. These clusters are isolated, and within each cluster, the conjugate plane of the orientation relationship, $(111)_\gamma$, has the highest density of near-coincident sites over a small area (0.3 nm width).

Two parallel arrays of interfacial dislocations on the interface $(111)_\gamma/(010)_\delta$ plane have been reported [44]. Both arrays of dislocations are parallel to the $[1\bar{1}0]_\gamma$ direction. One array consists of weakly contrasting dislocations spaced 7 nm apart with Burgers vector $\mathbf{b}= \frac{1}{6}[11\bar{2}]_\gamma$; those in the second array have stronger contrast, are spaced 24 nm apart, and have a $\mathbf{b}= \frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ Burgers vector. As in the fcc/bcc case, the influence of these misfit dislocations on the boundary atomic matching was investigated by shifting the atoms in the fcc phase above a $(111)_\gamma$ plane by $\frac{+\mathbf{b}}{2}$ on one side of the dislocation and $\frac{-\mathbf{b}}{2}$ on the other side of the dislocation. The resulting distribution of NCS are shown in Figure 3.7 and Figure 3.8. Both cases show an increase in the number of near-coincident sites and an increased continuity of the near-coincident site clusters. The introduction of interfacial dislocation $\mathbf{b}= \frac{1}{6}[11\bar{2}]_\gamma$ also connects isolated clusters, and makes the locally best matching plane extends to become a good matching plane over a large region. This continuity of matching is probably responsible for the precipitate adopting this plane as its habit plane. An array of misfit dislocation with $\mathbf{b}= \frac{1}{6}[2\bar{1}\bar{1}]_\gamma$ Burgers vector increases the number of near-coincident sites along the $[111]_\gamma$ direction (normal to the habit plane), but it does not

appear to be effective at increasing the extent of matching in the habit plane.

As in the fcc/bcc system, elastic strain [44] can also be introduced to the matching calculation for the γ/δ system (Figure 3.9), because the spacing of the conjugate planes, $(111)_\gamma$ and $(010)_\delta$, differs. It is found that the number of near-coincident sites increases, but the size of the individual clusters does not change much.

Figure 3.10(a) and (b) are orthogonal views of a single cluster of adjacent near-coincident sites (viewed along $[1\bar{1}0]_\gamma$ and $[11\bar{2}]_\gamma$, respectively). Unlike the fcc/bcc case, the apparent habit plane coincides with the conjugate plane. In addition, the near-coincident sites in a cluster do not form an atomically flat plane. A plane through these clusters of near-coincident sites can deviate as much as 3° from the $(111)_\gamma$. Experimentally, the habit plane of δ precipitates does wander 3° about the $(111)_\gamma$ [84].

3.4 Discussion

Both the fcc/bcc and γ/δ systems suggest that conjugate planes of the orientation relationship are determined by the locally best matching planes, defined as the planes with the highest density of near-coincident sites over a small region. The physical origin for this correlation is probably in the nucleation stage of precipitate formation. The interphase boundary of a critical nucleus contains no more than a few tens of atoms. Minimizing the energy of the interphase boundary (which we equate with finding the best matching across it) would be achieved by making parallel the planes in the two phases that were most similar. The "habit" or facet planes on the critical nucleus are thus likely to be the conjugate planes rather than the planes observed at much larger precipitate sizes.

As the precipitate grows in size, misfit in the conjugate planes increases, and elastic strain is lessened if a better interface can be found. Such a boundary is manifested by a

continuous cluster of near-coincident sites that extends over a large area. This cluster may or may not correspond to a low index plane depending upon the orientation relationship and the lattice parameter ratio. If a continuous cluster of near-coincident sites is not available for a given orientation relationship, then misfit accomodating defects must be introduced for precipitate growth to proceed.

By incorporating obeserved misfit dislocation structures into the matching calculation for a γ/δ interface not only is the number of near-coincident sites in the experimental habit plane greatly increased, but isolated clusters connect into a large region of good matching (Figure 3.7). Thus, continuity of near-coincident sites seems to be a requirement for a precipitate habit or facet plane. The near-coincident sites in the cluster array are not coplanar suggesting the interphase boundary plane need not be atomically flat.

For fcc/bcc matching, partial dislocations $\frac{1}{6}[11\bar{2}]_f$ or $\frac{1}{6}[\bar{2}11]_f$ associated with a stacking fault in parent fcc phase extend clusters of near-coincident sites in poor-matching directions. This extension does not occur along the conjugate planes, but rather along an average plane near $(1\bar{2}1)_f$. The interface wanders in step-wise fashion from one near-coincident site cluster to the next producing an array of structural ledges. As in the case of the γ/δ habit plane, the exact location of the interphase boundary plane is not rigid, but wanders slightly about the average boundary plane.

These characteristics of the clusters of adjacent near-coincident sites in the two cases considered here suggest the following hypothesis: the conjugate plane of the orientation relationship between a precipitating phase and its parent phase is that plane which has the best matching (greatest planar density of near-coincident sites) over a small, local region. The average habit plane, or largest facet plane, follows the extension of the cluster of near-coincident sites. In other words, it is believed that the conjugate plane is the locally best matching plane over a small area, and locally best matching plane is generally

a closest-packed plane in each of the two phases. Thus, for any two crystals with their crystal structures and lattice parameters known, the hypothesis provides a starting point how nature selects the orientation relationships, and by some adjustment of the close-packed planes and directions, the conjugate planes may be determined by the locally near-coincident site clusters, Extension or linking up of near-coincident site clusters determines the oriention of the microscopic (but not atomic level) habit plane.

The partial dislocations (stacking faults in the fcc/bcc case and misfit dislocations in the fcc/orthorhombic case) improve the atomic matching, particularly along the directions other than the best matching direction in the habit plane. This extends the clusters of near-coincident sites and thus provides a favorable path for the interphase boundary. When the conjugate plane coincides with the apparent habit plane, as in the case of γ/δ, the misfit dislocations must have Burgers vectors lying in the habit plane [44] in order to compensate the misfit. However, when the conjugate plane does not coincide with the apparent habit plane, as in the case of fcc/bcc, partial dislocations with Burgers vector lying out of the habit plane can extend the near-coincident sites cluster along other directions rather than the best matching direction $[10\bar{1}]_f$. In this case, the Burgers vector of the dislocations need not have a component in the direction of the misfit in the interface. The improve matching by introducing a small step in the boundary (a structural ledge [41]) that introduces a "pattern advance" [85] in the habit plane. The modeling results for both systems show that partial dislocations associated with either misfit dislocations or stacking faults are much more effective at extending the clusters of near-coincident sites than elastic strain.

Although geometric matching does identify the likely path of the interphase boundary and the directions which require misfit accommodation, there are limitations to the approach. For example, it is difficult to predict what type of dislocation will be acquired to compensate the misfit. In some systems, a less than optimal habit plane may be selected

3.5 Conclusions

The concept of near-coincidence sites can be employed to understand orientation relationships and interphase boundary orientations. The computational nature of the method avoids the need for specifying a lattice correspondence or a homogeneous deformation.

1. The conjugate plane of the orientation relationship is likely to be the best-matching plane over a small area identified as the plane with the highest density of near-coincident sites.

2. The average habit plane lies in an array of closely-spaced and continuous near-coincident sites. These sites need not be planar.

3. When clusters of near-coincident sites are not continuous, they can be extended by: (a) dislocations with a Burgers vector in the conjugate plane to connect adjacent clusters. (b) dislocations with a Burgers vector out of the conjugate plane which produce a "pattern advance" to extend a cluster.

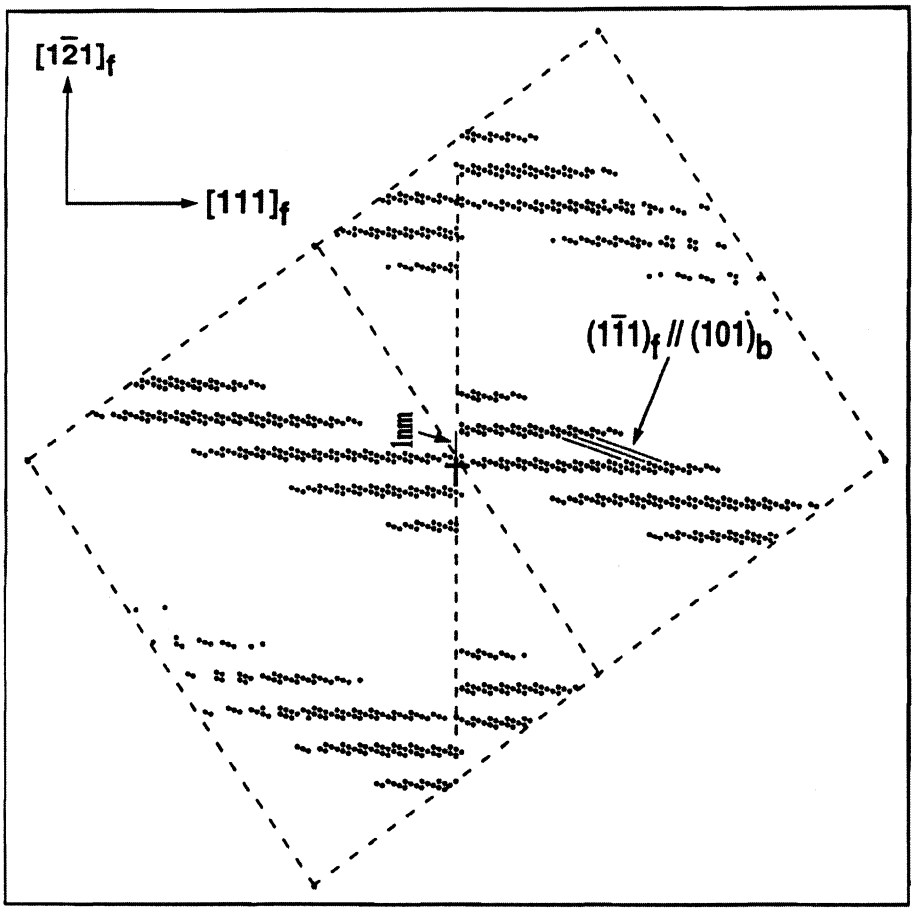


Figure 3.4: A projection of the fcc/bcc NCS onto the $(10\bar{1})_f$ plane with partial dislocation $\frac{1}{6}[11\bar{2}]_f$ located in the origin considered.

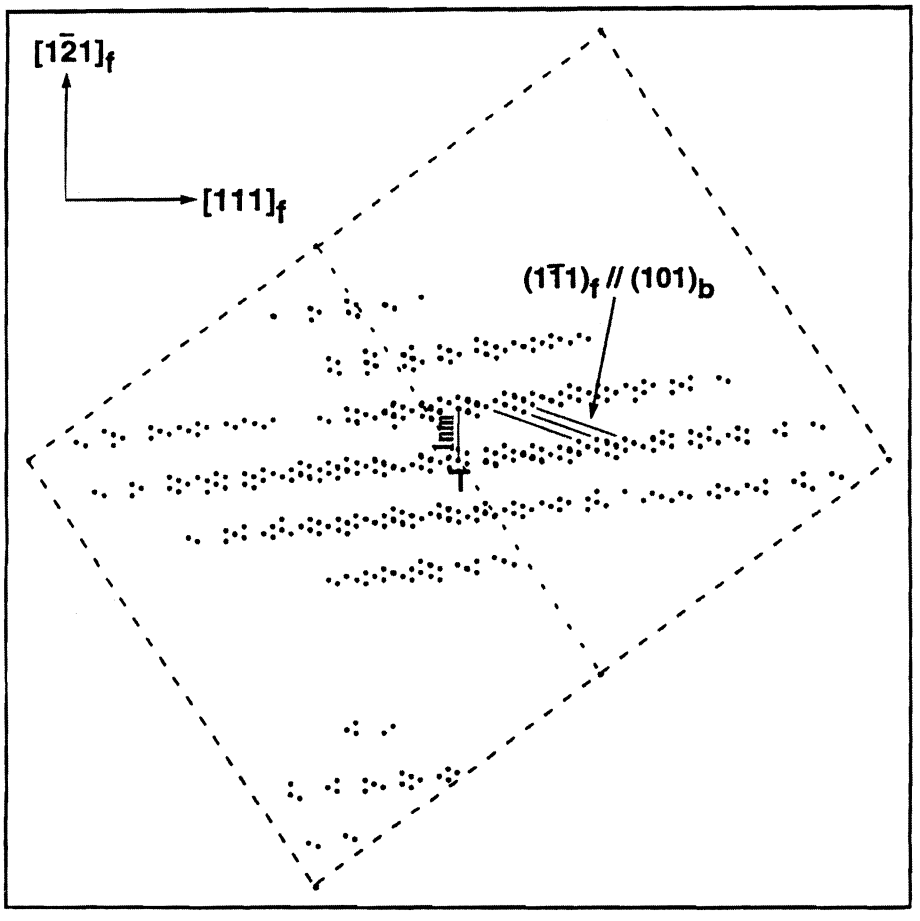


Figure 3.5: A projection of the fcc/bcc NCS onto the $(10\bar{1})_f$ plane with both elastic strain and partial dislocation $\frac{1}{6}[11\bar{2}]_f$ associated with SF considered.

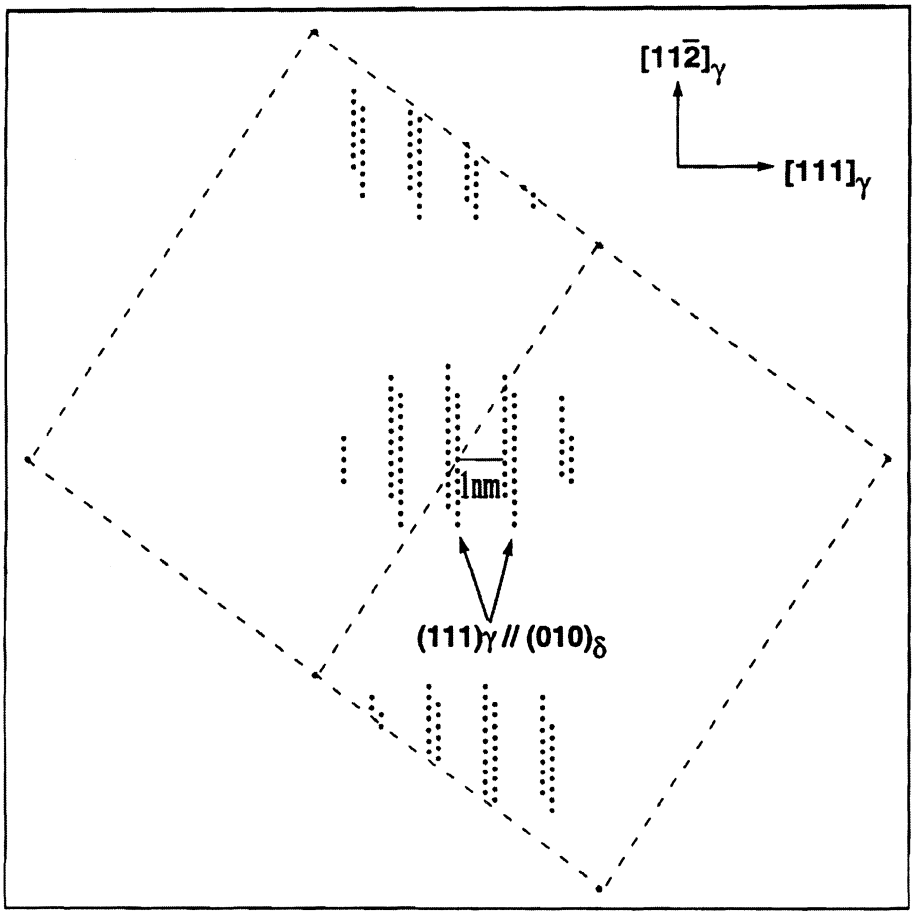


Figure 3.6: A projection of the γ/δ near-coincident sites onto the $(1\bar{1}0)_\gamma$ plane.

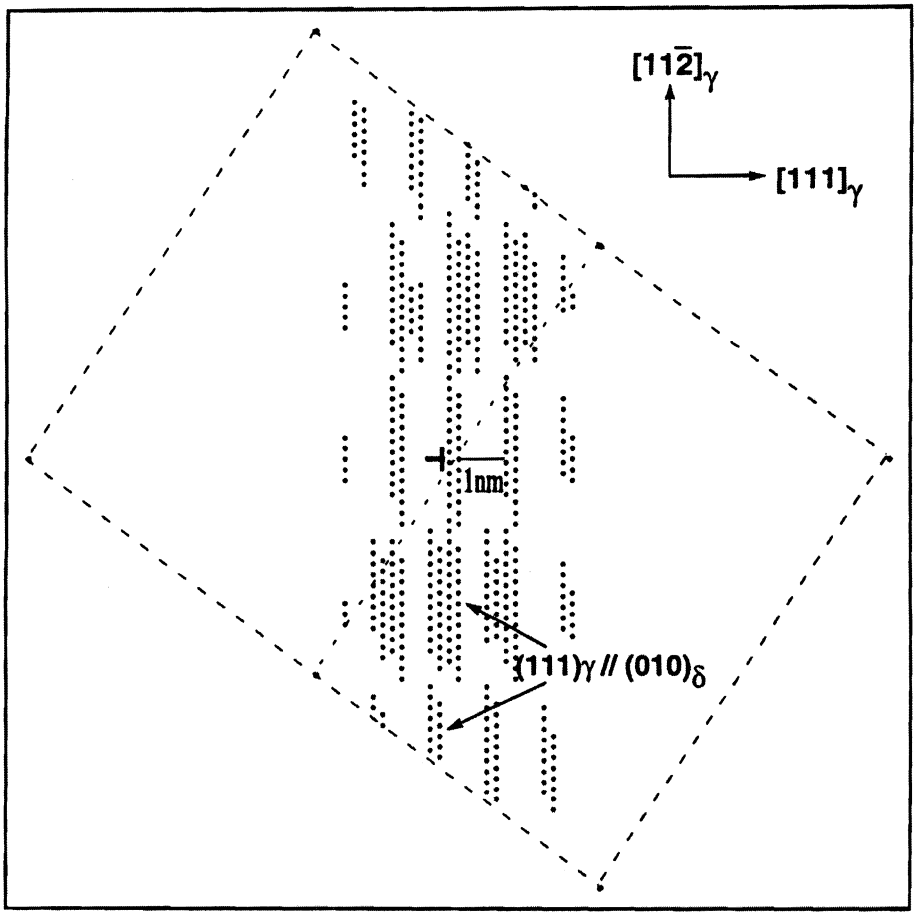


Figure 3.7: A projection of the γ/δ near-coincident sites onto the $(1\bar{1}0)_\gamma$ plane; a $\mathbf{b}= \frac{1}{6}[11\bar{2}]_\gamma$ misfit dislocation is located at the origin.

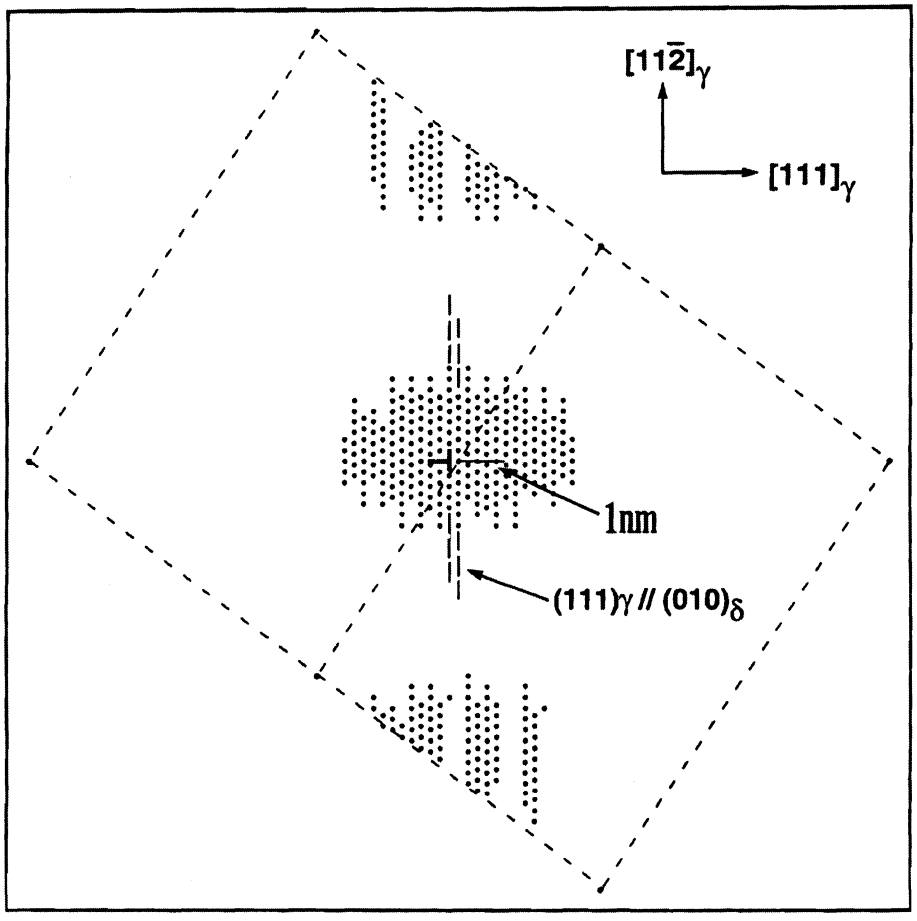


Figure 3.8: A projection of the γ/δ near-coincident sites onto the $(1\bar{1}0)_\gamma$ plane; a **b**= $\frac{1}{6}[2\overline{11}]_\gamma$ misfit dislocation is located at the origin.

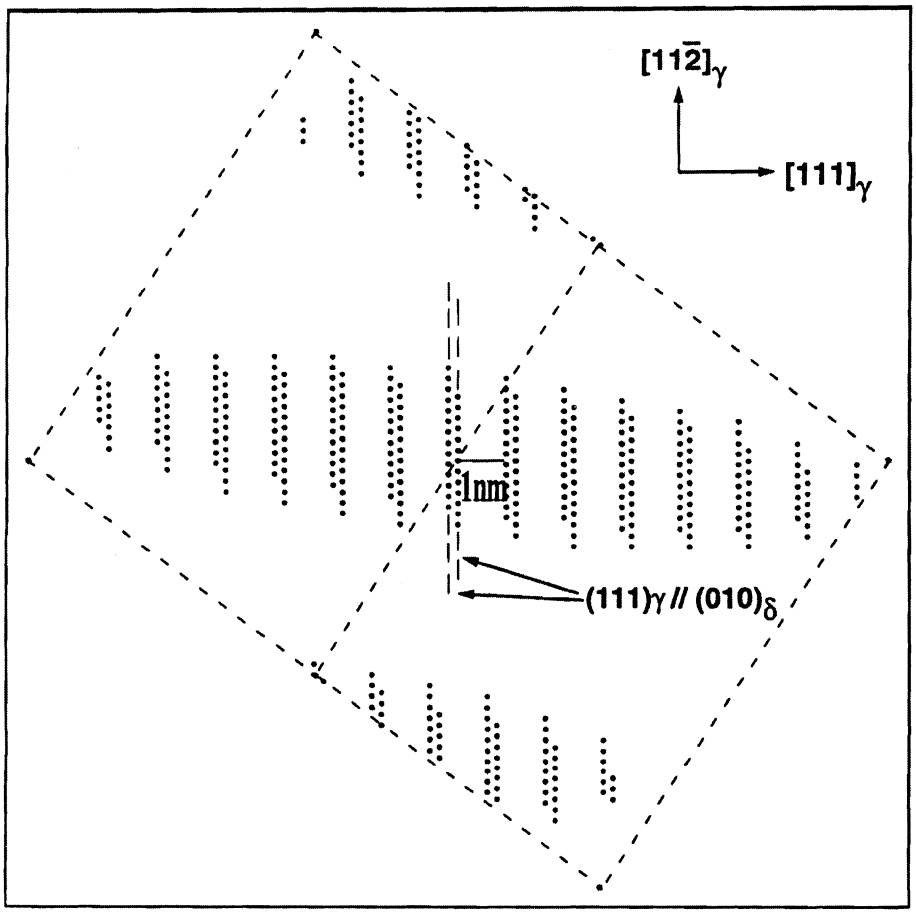


Figure 3.9: A projection of the γ/δ NCS onto the $(1\bar{1}0)_\gamma$ plane with elastic strain considered.

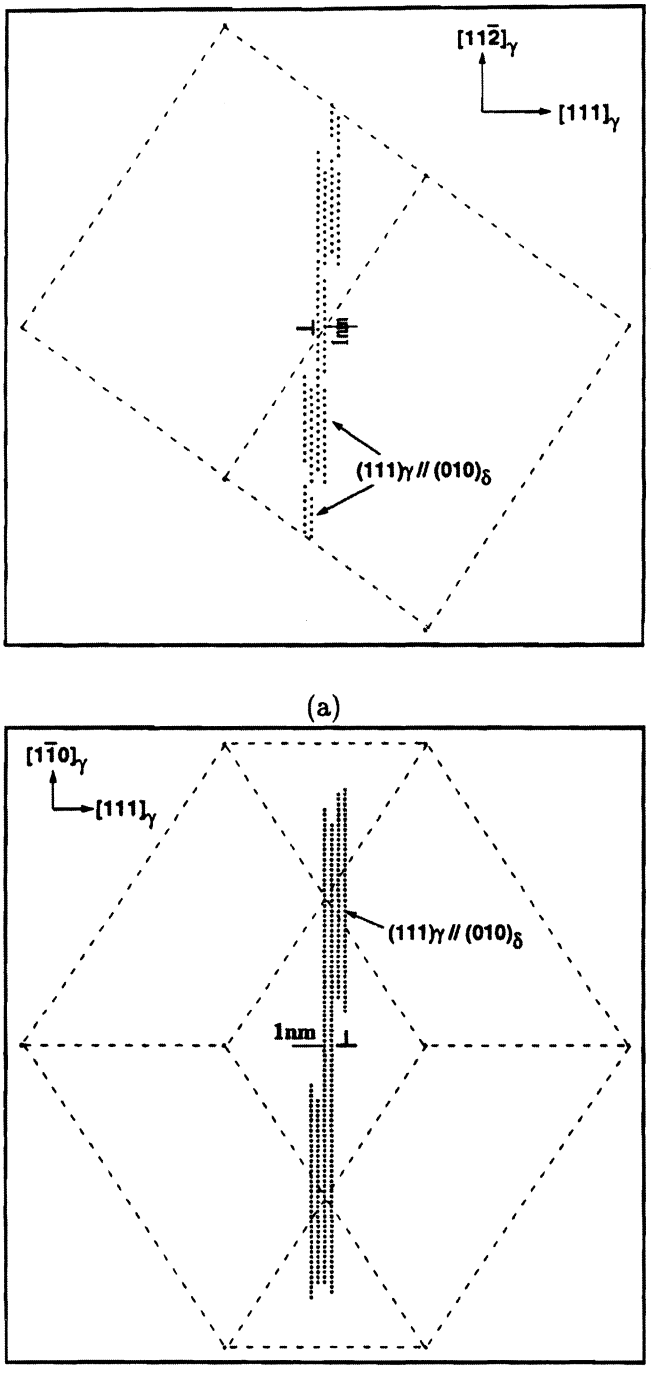


Figure 3.10: A projection of any two adjacent γ/δ NCS within 0.7 nm onto the (a) $(1\bar{1}0)_\gamma$ plane and (b) $(11\bar{2})_\gamma$ plane, with a misfit dislocation b= $\frac{1}{6}[11\bar{2}]_\gamma$ located in the origin.

Chapter 4

Summary

This work does a complete interphase boundary structure and associated defects study of δ precipitates in γ matrix in Inconel 718, compared with those previous results of fcc/orthorhombic interface in $Ni - Ni_3Nb$ eutectic, or similar system $Ni_3Al(\gamma')/Ni_3Nb(\delta)$ [71, 69, 70]. The experimental results serve as two purposes: first, provide the evidence to determine the controlling mechanism on the precipitate growth; second, along with the results form fcc/bcc in N–45% Cr by JKChen [19], provide a source for atomic matching approach based upon the near-coincident sites to analyze the interface boundary directions and precipitate growth direction.

In Chapter 2, an array of misfit dislocations on interface is always found parallel to $[1\bar{1}0]_\gamma$ direction with Burgers vector of $\frac{1}{6}[11\bar{2}]_\gamma$. They are served to accommodate the misfit between locally best matching patches (parallel to $(111)_\gamma$) and connect them as overall habit plane. On the other hand, in fcc/bcc case, the effective Burgers vector $\frac{1}{6}[11\bar{2}]_f$ giving rise to stacking faults, is served to extend the overall best matching plane $(1\bar{2}1)_f$. All these analysis are carried out in Chapter 3 by geometric atomically matching with the aid of near-coincident site.

The general procedure of determining the interphase boundary and growth directions of precipitate is described in Chapter 3.

The following conclusions can be drawn:

1. Interphase boundary structure and defects by TEM observations of γ/δ in Inconel 718

implies that the thickening of δ plate occurs by ledge mechanism, as well inconsistent with the maintenance of shear mechanism based upon the lattice correspondence.

2. The conjugate direction between two crystals is parallel to the best matching direction.

3. Conjugate plane is determined by the locally best matching plane identified as the plane with the highest areal density of near-coincident sites over a relatively small region.

4. Interphase boundary planes follow a cluster of closely spaced near-coincident sites. In other words, the average habit plane, or largest facet plane, follows the extension of the cluster of near-coincident sites over a large area. The near-coincident sites within a cluster are contiguous, but they need not be exactly planar.

5. Defects with Burgers vector lying on the interface should exist when the habit plane coincides with the conjugate plane. Otherwise, defects with Burgers vector lying out of the interface should be present to extend the near-coincident sites clusters.

6. It may be possible to predict precipitate orientation relationship and shapes from the crystal structures by using coordination polyhedra to identify orientation relationship with high local near-coincident sites density and then finding the near-coincident sites clusters under these orientation relationships.

REFERENCES

[1] H. I. Aaronson, in *The Decomposition of Austenite by Diffusional Processes*, V. F. Zackay and H. I. Aaronson, eds., Interscience, NY, 1962, pp. 387–549.

[2] H. I. Aaronson, C. Laird, and K. R. Kinsman, in *Phase Transformations*, ASM, Metals Park, OH, 1970, pp. 313–396.

[3] G. C. Weatherly, P. Humble, and D. Borland, *Acta Metall.*, 1979, vol. 27, pp. 1815–1823.

[4] D. J. Dyson, S. R. Keown, D. Raynor, and J. A. Whiteman, *Acta Metall.*, 1966, vol. 14, p. 867.

[5] U. Dahmen, P. Ferguson, and K. H. Westmacott, *Acta Metall.*, 1984, vol. 32, pp. 803–810.

[6] U. Koster, *Mat.Sci.Eng*, 1969, vol. 5, pp. 174–182.

[7] E. Ozawa and H. Kimura, *Acta Metall.*, 1970, vol. 18, pp. 995–1002.

[8] U. Dahmen and K. H. Westmacott, *Science*, 1986, vol. 233, pp. 875–880.

[9] H. Emoto, R. Ni-Ikura, and M. Kikuchi, in *Phase Transformations '87*, G. W. Lorimer, ed., The Institute of Metals, London, 1988, pp. 169–173.

[10] M. Kikuchi, A. F. Guillermet, M. Hillert, G. Cliff, and G. W. Lorimer, *Acta Metall. Mater.*, 1990, vol. 38, pp. 165–171.

[11] J. M. Howe, U. Dahmen, and R. Crosky, *Phil. Mag. A*, 1987, vol. 56, pp. 31–61.

[12] U. Dahmen and K. H. Westmacott, *Acta Metall.*, 1986, vol. 34, pp. 475–482.

[13] J. M. Lang, U. Dahmen, and K. H. Westmacott, *Phys. Stat. Sol. (a)*, 1983, vol. 75, pp. 409–418.

[14] P. Ferguson, , U. Dahmen, and K. H. Westmacott, *Scr. Metall.*, 1984, vol. 18, pp. 57–66.

[15] G. Spanos and H. I. Aaronson, *Acta Metall.*, 1990, vol. 38, pp. 2721–2732.

[16] C. P. Luo and G. C. Weatherly, *Acta Metall.*, 1987, vol. 35, pp. 1963–1972.

[17] H. Zhao, K. T. Aust, and G. C. Weatherly, *Acta Metall. Mater.*, 1992, vol. 40, pp. 1961–1968.

[18] C. P. Luo, U. Dahmen, and K. H. Westmacott, *Acta Metall. Mater.*, 1994, vol. 42, pp. 1923–1932.

[19] J. K. Chen, G. Chen, and W. T. Reynolds Jr., Virginia Polytechnic Institute and State University, to be submitted to *Acta Metall. Mater.* or *Phil. Mag. A.*, 1995.

[20] J. W. Christian, *The Theory of Transformations in Metal and Alloys, Part I: Equilibrium and General Kinetic theory*, volume 2nd ed. of *International series on materials science and technology v.15*, Pergamon Press, Oxford, England, 1975.

[21] H. I. Aaronson, T. Furuhara, J. M. Rigsbee, W. T. Reynolds Jr., and J. M. Howe, *Metall. Trans. A*, 1990, vol. 21A, pp. 2369–2409.

[22] H. I. Aaronson and M. G. Hall, *Metall. Mater. Trans. A*, 1994, vol. 25A, pp. 1797–1819.

[23] M. S. Wechsler, D. S. Lieberman, and T. A. Read, *Trans. AIME*, 1953, vol. 197, pp. 1503–1515.

[24] J. S. Bowles and J. K. Mackenzie, *Acta Metall.*, 1954, vol. 2, pp. 129–137.

[25] J. K. Mackenzie and J. S. Bowles, *Acta Metall.*, 1954, vol. 2, pp. 138–147.

[26] J. S. Bowles and J. K. Mackenzie, *Acta Metall.*, 1954, vol. 2, pp. 224–234.

[27] C. M. Wayman, *Introduction to the Crystallography of Martensitic Transformation*, Macmillan Series in Materials Science, Macmillan, NY, 1964.

[28] W. Bollmann, *Crystal Defects and Crystalline Interfaces*, Springer-Verlag, New York, 1970.

[29] U. Dahmen, *Scr. Metall.*, 1981, vol. 15, pp. 77–81.

[30] U. Dahmen, *Acta Metall.*, 1982, vol. 30, pp. 63–73.

[31] W. Bollmann, *Phys. Stat. Sol.*, 1974, vol. A21, pp. 543–550.

[32] W. Bollmann, *Phil. Mag.*, 1967, vol. 16, pp. 363–381.

[33] W. Bollmann, *Phil. Mag. A*, 1967, vol. 16, pp. 383–399.

[34] E. C. Bain and N. Y. Dunkirk, *Trans. AIME*, 1924, vol. 70, pp. 25–46.

[35] M. A. Jaswon and J. A. Wheeler, *Acta Cryst.*, 1948, vol. 1, pp. 216–224.

[36] U. Dahmen and K. H. Westmacott, in *Proceedings of an International Conference on Solid-Solid Phase Transformations*, R. F. Sekerka H. I. Aaronson, D. E. Laughlin and C. M. Wayman, eds., TMS-AIME, Warrendale, PA, 1983, pp. 433–437.

[37] J. W. Christian and D. V. Edmonds, in *Phase Transformations in Ferrous Alloys*, A. R. Marder and J. I. Goldstein, eds., TMS-AIME, Philadelphia, PA, 1983, pp. 293–325.

[38] W. T. Reynolds Jr., H. I. Aaronson, and G. Spanos, *Mater. Trans., JIM*, 1991, vol. 32, pp. 737–746.

[39] H. I. Aaronson and C. Laird, *Trans. AIME*, 1968, vol. 242, pp. 1437–1448.

[40] J. M. Rigsbee and H. I. Aaronson, *Acta Metall.*, 1979, vol. 27, pp. 365–376.

[41] J. M. Rigsbee and H. I. Aaronson, *Acta Metall.*, 1979, vol. 27, pp. 351–363.

[42] T. Furuhara and H. I. Aaronson, *Acta Metall. Mater.*, 1991, vol. 39, pp. 2857–2872.

[43] J. K. Chen and W. T. Reynolds Jr., Virginia Polytechnic Institute and State University, submitted to *Acta Metall. Mater.* or *Phil. Mag. A.*, 1995.

[44] Q. Liang and W. T. Reynolds Jr., Virginia Polytechnic Institute and State University, to be submitted to *Metall. Trans.*, 1996.

[45] Q. Liang and W. T. Reynolds Jr., Virginia Polytechnic Institute and State University, to be submitted to *Acta Metall. Mater.* or *Phil. Mag. A.*, 1996.

[46] I. Kirman, *JISI*, 1969, vol. 207, pp. 1612–1618.

[47] H. I. Aaronson, W. T. Reynolds Jr., G. J. Shiflet, and G. Spanos, *Metall. Trans. A*, 1990, vol. 21A, pp. 1343–1380.

[48] C. P. Luo and G. C. Weatherly, *Phil. Mag. A*, 1988, vol. 58, pp. 445–462.

[49] T. Fujii, T. Mori, and M. Kato, *Acta Metall. Mater.*, 1992, vol. 40, pp. 3413–3420.

[50] J. A. Hren and G. Thomas, *Trans. Metall. Soc. AIME*, 1963, vol. 227, pp. 308–318.

[51] J. M. Howe, H. I. Aaronson, and R. Crosky, *Acta Metall.*, 1985, vol. 33, pp. 639–648.

[52] C. Laird and H. I. Aaronson, *Acta Metall.*, 1969, vol. 17, pp. 505–525.

[53] G. J. Mahon and J. M. Howe, *Metall. Trans. A*, 1990, vol. 21, pp. 1655–1662.

[54] J. W. Brooks and P. J. Bridges, in *Superalloys 1988*, E. A. Loria, ed., TMS-ASM, Warrendale, PA, 1988, pp. 33–38.

[55] C. T. Sims, N. S. Stoloff, and W. C. Hagel, *The Superalloys*, volume 2nd ed. of *Wiley Series on the Science and Technology of Materials*, John Wiley & Sons, New York, USA, 1987.

[56] I. Kirman and D. H. Warrington, *Metall. Trans.*, 1970, vol. 1, pp. 2667–2675.

[57] M. H. Loretto and R. E. Smallman, *Defect Analysis in Electron Microscopy*, Chapman and Hall, London, 1975.

[58] A. K. Head, P. Humble, L. M. Clarebrough, A. J. Morton, and C. T. Forwood, *Computed Electron Micrographs and Defect Identification*, volume 7 of *Defects in Crystalline Solids*, North-Holland, Amsterdam, 1973.

[59] C. T. Forwood and L. M. Clarebrough, *Electron Microscopy of Interfaces in Metals and alloys*, Electrom microscopy in materials science series, Adam Hilger, Bristol, England, 1991.

[60] G. C. Weatherly, *Acta Metall.*, 1971, vol. 19, pp. 181–192.

[61] G. C. Weatherly and T. D. Mok, *Surf. Sci.*, 1972, vol. 31, pp. 335–341.

[62] D. J. H. Cockayne, I. L. F. Ray, and M. J. Whelan, *Phil. Mag.*, 1969, vol. 20, Ser.8, pp. 1265–1270.

[63] D. J. H. Cockayne, *J. Microscopy*, 1973, vol. 98, Pt. 2, pp. 116–134.

[64] M. Sundararaman, P. Mukhopadhyay, and S. Banerjee, *Metall. Trans. A*, 1988, vol. 19, pp. 453–465.

[65] D. Senicourt and P. Annarumma, *Compt. Rend.*, 1969, vol. 269C, pp. 591–598.

[66] A. Royer and M. Gantois, *Mem.Sci.Rev.Met.*, 1971, vol. 58, p. 1.

[67] V. Perovic and G. C. Weatherly, *Acta Metall.*, 1989, vol. 37, pp. 813–821.

[68] K. Chattopadhyay and H. I. Aaronson, *Acta Metall.*, 1986, vol. 34, pp. 695–711.

[69] Y. G. Nakagawa and G. C. Weatherly, *Mater. Sci. Eng.*, 1972, vol. 10, pp. 223–.

[70] G. Garmong and C. G. Rhodes, *Trans. Met. Soc. AIME*, 1975, vol. 6 A, pp. 2209–2216.

[71] P. Annarumma and M. Turpin, *Metall. Trans.*, 1972, vol. 3, pp. 137–146.

[72] J. K. Chen and W. T. Reynolds Jr., in *Proceeding of International Conference on Solid-Solid Phase Transformations in Inorganic Materials '94*, W. C. Johnson, J. M. Howe, D. E. Laughlin, and W. A. Soffa, eds., TMS, Warendale, PA, 1994, pp. 1091–1096.

[73] G. Thomas, *Transmission Electron Microscopy of Metals*, Electrom microscopy in materials science series, John Wiley & Sons Inc., U.S.A, 1962.

[74] J. P. Hirth and J. Lothe, *Theory of Dislocations*, Wiley, New York, 1982.

[75] W. Z. Zhang and G. R. Purdy, *Phil. Mag. A*, 1993, vol. 68, pp. 279–290.

[76] W. Z. Zhang and G. R. Purdy, *Phil. Mag. A*, 1993, vol. 68, pp. 291–303.

[77] K. M. Knowles, *Phil. Mag. A*, 1982, vol. 46, pp. 951–969.

[78] K. M. Knowles and D. A. Smith, *Phil. Mag. A*, 1983, vol. 48, pp. 527–553.

[79] M. G. Hall, H. I. Aaronson, and K. R. Kinsman, *Surf. Sci.*, 1972, vol. 31, pp. 257–263.

[80] G. V. Kurdjumov and G. Sachs, *Z. Physik*, 1933, vol. 64, pp. 325–345.

[81] K. Ameyama, G. C. Weatherly, and K. T. Aust, *Acta Metall. Mater.*, 1992, vol. 40, pp. 1835–1846.

[82] D. W. Bare, E. D. Gibson, and O. N. Carlson, *Trans. TMS-AIME*, 1964, vol. 230, pp. 934–936.

[83] G. Chen and W. T. Reynolds Jr., to be submitted to *Acta Metall. Mater.*, 1995.

[84] Q. Liang and W. T. Reynolds Jr., Unpublished work, Virginia Polytechnic Institute and State University, 1994.

[85] J. H. Van Der Merwe, G. J. Shiflet, and P. M. Stoop, *Metall. Trans. A*, 1991, vol. 22, pp. 1165–1175.

Appendix A

Computer Programs

The computer programs used in this work are mainly those employed to calculate the near coincident sites (NCS) of γ/δ in Inconel 718 and fcc/bcc in Ni–45% Cr systems [44]. They are listed as follows:

A.1 List of programs

1. *ncs718elds.c* produces the near-coincident sites between γ/δ phases in Inconel 718, if the atoms position change by elastic strain and misfit dislocations is taken into account or not. Two output files, one text file and another binary format data file, are available when program finish running. The binary output file can also be viewed along selected directions by *Mathematica* and generates those projection plots. Some important parameters have to be input:

   (1) Orientation relationship, lattice parameters;

   (2) Simulation block size, coincidence criterion;

   (3) elastic strain;

   (4) Burgers vector from misfit dislocation, and where to insert.

2. *ncsfbelds.c* is partially inherited from ncs718elds.c. It produces the near-coincident sites between fcc/bcc phases in Ni–45%wt Cr, if the atoms position change by elastic strain and partial dislocations associated with the stacking faults is taken into account

or not. Two output files, one text file and another binary format data file, are available when program finish running. Input parameter are almost the same as ncs718elds.c requires, except the effective Burgers vector caused by stacking faults rather than misfit dislocations.

3. *plane.c* generates an array of near-coincident sites, where any two adjacent NCS lying on this array is within some small distance. The average direction of the array can be treated as the average habit plane over the simulation block. Important input parameters include :

   (1) Binary NCS data file produced by ncs718elds.c or ncsfbelds.c;

   (2) The mismatch distance between any two adjacent near-coincident sites.

4. *bmd.c* generates an array of near-coincident sites, where any two adjacent NCS are within the least distance. The direction of these NCS (in a line) can be treated as the besting matching direction between two crystals.

A.2 Source codes

Programs included are *ncs718elds.c*, *ncsfbelds.c*, *plane.c*, and *bmd.c*.

```
/*
    ncs718elds.c
    using J.K.Chen's MatrixOperation.h

    Qiang Liang, T_Group, MSE, Va Tech, USA
    March, c1996

    Calculating the NCS between fcc/orthorhombic in In718 for all atom
    sites. An burgers vector caused by misfit dislocations has been put
    into calculation to change atoms positions in fcc in some region.
    An elastic strain, elastic1, is placed in [1,1,1]fcc direction in
    order to make d(111)fcc = d(010)orth.

*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include </usr/users/jkchen/Program/MatrixOperation/MatrixOperation.h>
                                        /* matrix operation utility */

#define elastic2 1.017544    /* Elastic strain=d(010)orth/d(111)fcc */
#define af 3.616             /* Lattice parameter of matrix fcc */
#define aorth 5.141          /* Lattice parameters of precipitate bcc */
#define borth 4.231
#define corth 4.534
#define worse 0.35*0.35      /* Degree of coherency between atoms */

void   preJoin();            /* Function translates all ppt lattice points
                                to matrix basis */
void   Join();               /* Function simplifies the matrix lattice points */
struct corres * allocate();
struct corres
{   float   S[3];
    struct  corres  *next;
};

/* Structure that stores the NCS between matrix:ppt */
struct corres  *firstFatom, *thisFatom;
struct corres  *firstPatom,  *thisPatom, *newPatom;
struct corres  *fitPoint, *thisPoint;
long length=0;
```

```
main()
{
  int i, j, k, n;
  float fi, fj, fk;
  int i1, j1, k1, i2, j2, k2;
  int xmin1, xmax1, ymin1, ymax1, zmin1, zmax1; /* Block size for matrix*/
  int xmin2, xmax2, ymin2, ymax2, zmin2, zmax2; /* Block size for ppt */
  int lengthFatom, lengthPatom;
  float **L, **T, **fcc, **orth, **orthi, **R, **p2f;
  float **tp1, **tp2, **tp3, **tp4;
  float **el,**xyz,**xyzi,**xyz2el,**el2xyz;
  float a, b, c, tmp1, tmp2, temp[3], count;
  float x, y, z, distsq, rn, std;
  float min, max;
  float temp1[3][3]={{1,1,1}, {1,-1,0},{1,1,-2}},
        temp2[3][3]={{0,1,0}, {1,0,0}, {0,0,-1}},
        temp4[3][3]={{1,0,0}, {0,1,0},{0,0,1}};
    /* Based on OR. "1" refer to matrix, "2" precipitate */

  FILE *fp, *fpt, *fp_worse;
  struct corres *worsePt, *thatPoint;

    /* Change matrix size */
  for(i=0; i<3; i++)
  {
     temp1[0][i]=temp1[0][i]/sqrt(3);
     temp1[1][i]=temp1[1][i]/sqrt(2);
     temp1[2][i]=temp1[2][i]/sqrt(6);
  }
  /* initializing matrices */

  xyz2el=DMatrix(3,3);      /* Translation matrix from cartesian basis to
                               {x,y,z}={{1,1,1}, {1,-1,0},{1,1,-2}}fcc basis */
  el2xyz=DMatrix(3,3);      /* Inverse for above matrix */
  xyz=DMatrix(3,3);         /* Matrix for Cartesian basis */
  xyzi=DMatrix(3,3);        /* Inverse of Cartesian matrix */
  el=DMatrix(3,3);          /* Translation matrix for elastic deformation */

  L=DMatrix(3, 3);          /* Axis length change matrix from orth to fcc */
  T=DMatrix(3, 3);
  fcc=DMatrix(3, 3);        /* fcc unit cell */
  orth=DMatrix(3, 3);       /* orth unit cell */
  orthi=DMatrix(3, 3);      /* Inverse of fcc unit cell */
  R=DMatrix(3, 3);          /* Rotation matrix between fcc/bcc based on OR */
```

```
p2f=DMatrix(3, 3);          /* Transformation matrix */
tp1=DMatrix(3, 3);          /* Follows are temporary 3 by 3 matrices */
tp2=DMatrix(3, 3);
tp3=DMatrix(3,3);
tp4=DMatrix(3,3);

for(i=0; i<3; i++)
  for(j=0; j<3; j++)
  {
  tp1[i][j]=temp1[i][j];
   tp2[i][j]=temp2[i][j];
   tp3[i][j]=temp1[i][j];
  tp4[i][j]=temp4[i][j];
  }

/* Important change here */
rn=0.11;          /* Range of two atoms needed for further calculation */
std=0.10*0.10;   /* Coherency criterion of NCS, using af as unit */

printf("Please enter the block size of matrix.\n");
scanf ("%d", &i1);
printf("Please enter the block size of ppt.\n");
scanf ("%d", &i2);

  /* Making equal size in x, y, z directions for the block */
j1=i1;
k1=i1;

j2=i2;
k2=i2;

  /* Obtain the axis length change matrix between two crystals */
for(i=0; i<3; i++)
  for(j=0; j<3; j++)
  {
  if((i==0) && (j==0))
     L[i][j]=aorth/af;
   else if((i==1) && (j==1))
   L[i][j]=borth/af;
   else if((i==2) && (j==2))
   L[i][j]=corth/af;
  else
   L[i][j]=0.0;
  }
```

```
   fi=i; fj=j; fk=k;
     preJoin(fi, fj, fk, i1, j1, k1, p2f);
     preJoin(fi+1.0/2, fj, fk, i1, j1, k1, p2f);
     preJoin(fi, fj+1.0/2, fk+0.69, i1, j1, k1, p2f);
    preJoin(fi+1.0/2, fj+1.0/2, fk+0.69, i1, j1, k1, p2f);
     preJoin(fi+1.0/4, fj, fk+1.0/2, i1, j1, k1, p2f);
    preJoin(fi+1.0/4, j+1.0/2, k+0.19, i1, j1, k1, p2f);
     preJoin(fi+3.0/4, fj, fk+1.0/2, i1, j1, k1, p2f);
    preJoin(fi+3.0/4, fj+1.0/2, fk+0.19, i1, j1, k1, p2f);
   }
lengthPatom=length;
printf("\nlengthFatom=%d   ", lengthFatom);
printf("lengthPatom=%d\n", lengthPatom);
fflush(stdout);


/* Following calculate the NCS sites */
length=0;
fitPoint=allocate();
thisPoint=fitPoint;
thisFatom=firstFatom;
worsePt=allocate();
thatPoint=worsePt;
for(i=1; i<=lengthFatom; i++)
{
  thisPatom=firstPatom;
  for(j=1; j<=lengthPatom; j++)
  {
  x=(thisFatom->S[0] - thisPatom->S[0]);
  y=(thisFatom->S[1] - thisPatom->S[1]);
    z=(thisFatom->S[2] - thisPatom->S[2]);
  if( (x<rn && x>-rn) && (y<rn && y>-rn) && (z<rn && z>-rn) )
    {
          distsq=x*x +y*y +z*z;
          if(distsq <= std)
            {
             for(k=0; k<3; k++)
             thisPoint->S[k]=thisFatom->S[k];
              thisPoint->next = allocate();
             thisPoint=thisPoint->next;
              length++;
            }
             if(distsq >= worse)
              {
                for(k=0; k<3; k++)
```

```
                thatPoint->S[k]=thisFatom->S[k];
                thatPoint->next = allocate();
                thatPoint=thatPoint->next;

              }

    }
   thisPatom=thisPatom->next;
   }
   thisFatom=thisFatom->next;
 }
/* End of Calculation */

/* Record the NCS referred to matrix index*/
 fp=fopen("n718elds.dat", "w");
 fpt=fopen("fit718elds.dat", "w");
 fprintf(fp, "The fcc size(total unit cells in one axis direction)
             in In718 is: %d\n", i1);
 fprintf(fp, "The total lattice points in fcc is: %d\n", lengthFatom);
 fprintf(fp, "The ppt size(total unit cells in one axis direction)
             in In718 is: %d\n", i2);
 fprintf(fp, "The total lattice points in ppt is: %d\n", lengthPatom);
 fprintf(fp, "The closest pair points representated by fcc are\n");


  /* Store all NCS sites in files as binary format */
 thisPoint=fitPoint;
 j=0;
 fprintf(fp, "{");
 for(i=0; i<length; i++)
 {
   if(j==3)
   {
    j=0;
    fprintf(fp,"\n");
   }
   fprintf(fp, "{%6.3f,", thisPoint->S[0]);
   fprintf(fp, "%6.3f,", thisPoint->S[1]);
   if(i==length-1)
     fprintf(fp, "%6.3f}", thisPoint->S[2]);
   else fprintf(fp, "%6.3f},", thisPoint->S[2]);
   j++;
```

```
    fprintf(fpt, " %6.3f", thisPoint->S[1]);
    fprintf(fpt, " %6.3f", thisPoint->S[1]);
    fprintf(fpt, " %6.3f", thisPoint->S[2]);

    thisPoint=thisPoint->next;
  }
  fprintf(fp, "}\n The number of total ncs is: %d", length);
  fclose(fp);
  fclose(fpt);


    /* Record the NCS as coherency degree you input  */
  fp_worse=fopen("worsel.dat", "w");
  thisPoint=worsePt;
  while(thisPoint->next != NULL)
  {
    fprintf(fp_worse, " %6.3f", thisPoint->S[0]);
    fprintf(fp_worse, " %6.3f", thisPoint->S[1]);
    fprintf(fp_worse, " %6.3f", thisPoint->S[2]);
    thisPoint=thisPoint->next;
  }
  fclose(fp_worse);

}

void preJoin(x, y, z, in, jn, kn, p2f)
float x, y, z, **p2f;
int in, jn, kn;
{
  int i;
  float temp[3];
  for(i=0; i<3; i++)
    temp[i] = p2f[i][0]*x+p2f[i][1]*y+p2f[i][2]*z;
  Join(temp[0], temp[1], temp[2], in, jn, kn);
}

void   Join(i, j, k, in, jn, kn)
float i, j, k;
int in, jn, kn;
{
  extern struct corres *thisPoint;
  float xmin, xmax, ymin, ymax, zmin, zmax;
  xmin=floor(-in/2.0);
  xmax=floor(in/2.0);
  ymin=floor(-jn/2.0);
```

```
  ymax=floor(jn/2.0);
  zmin=floor(-kn/2.0);
  zmax=floor(kn/2.0);
  if( (i<xmin) || (i>xmax) || (j<ymin) || (j>ymax) || (k<zmin) || (k>zmax))
     ;
  else{
    thisPoint->S[0]=i;
    thisPoint->S[1]=j;
    thisPoint->S[2]=k;
    thisPoint->next = allocate();
    thisPoint=thisPoint->next;
    length++;
    }
}

struct corres * allocate()
{
    struct corres * newone;
    newone = (struct corres *) malloc (sizeof(struct corres));
     newone->next = NULL;
     return(newone);
}
```

```
/*
    ncsfbelsf.c
    using J.K.Chen's MatrixOperation.h

    Qiang Liang, T_Group, MSE, Va Tech, USA
    March, c1996

    Calculating the NCS between fcc/bcc in Ni-45wt%Cr for all atom
    sites. An effective vector caused by stacking faults has been put
    into calculation to change atoms positions in fcc in some region.
    An elastic strain, elastic1, is placed in [1,-1,1]fcc direction
    in order to make d(1-11)fcc = d(101)bcc.

    */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include </usr/users/jkchen/Program/MatrixOperation/MatrixOperation.h>
                                        /* matrix operation utility */

#define elastic1 0.976052       /* Elastic strain=d(101)bcc/d(1-11)fcc */
#define af 3.664                /* Lattice parameter of matrix fcc */
#define ab 2.920                /* Lattice parameter of precipitate bcc */
#define worse 0.18*0.18         /* Degree of coherency between atoms */

void  Join();                   /* Function simplifies the matrix lattice points */
struct corres * allocate();
struct corres
{  float   S[3];
   struct  corres  *next;
};

/* Structure that stores the NCS between matrix/ppt */
struct corres   *firstFatom,  *thisFatom;
struct corres   *firstPatom, *thisPatom, *newfirstPatom, *newPatom;
struct corres   *fitPoint, *thisPoint;
long length=0;

main()
{
  int i, j, k, n;
  float fi, fj, fk;
```

```
int i1, j1, k1, i2, j2, k2;
int xmin1, xmax1, ymin1, ymax1, zmin1, zmax1;          /* Block size for matrix*/
int xmin2, xmax2, ymin2, ymax2, zmin2, zmax2;          /* Block size for ppt  */
long lengthFatom, lengthPatom;
float **L, **T, **fcc, **bcc, **bcci, **R, **p2f,
      **tp1,**tp2,**tp3,**tp4;
float **el,**xyz,**xyzi,**xyz2el,**el2xyz;
float a, b, c, tmp1, tmp2, temp[3],count;
float x, y, z, distsq, rn, std;
float  temp1[3][3]={{1,-1,1}, {-1,0,1},{-1,-2,-1}},
       temp2[3][3]={{1,0,1}, {-1,-1,1}, {1,-2,-1}},
       temp3[3][3]={{1,-1,1}, {1,0,-1},{1,2,1}},
   temp4[3][3]={{1,0,0}, {0,1,0},{0,0,1}};
  /* Based on OR. "1","3" refer to matrix, "2","4" precipitate */

FILE *fp, *fpt, *fp_worse;
struct corres *worsePt, *thatPoint;

  /* Change matrix */
for(i=0; i<3; i++)
{
   temp1[0][i]=temp1[0][i]/sqrt(3);
  temp1[1][i]=temp1[1][i]/sqrt(2);
  temp1[2][i]=temp1[2][i]/sqrt(6);
}
for(i=0; i<3; i++)
{
   temp2[0][i]=temp2[0][i]/sqrt(2);
  temp2[1][i]=temp2[1][i]/sqrt(3);
  temp2[2][i]=temp2[2][i]/sqrt(6);
}
for(i=0; i<3; i++)
{
   temp3[0][i]=temp3[0][i]/sqrt(3);
  temp3[1][i]=temp3[1][i]/sqrt(2);
  temp3[2][i]=temp3[2][i]/sqrt(6);
}

/* initializing matrices */
xyz2el=DMatrix(3,3);          /* Translation matrix from cartesian basis to
                                 {x,y,z}={{1,-1,1}, {1,0,-1},{1,2,1}}fcc basis */
el2xyz=DMatrix(3,3);          /* Inverse for above matrix */
xyz=DMatrix(3,3);             /* Matrix for Cartesian basis */
```

```
xyzi=DMatrix(3,3);              /* Inverse of Cartesian matrix */
el=DMatrix(3,3);                /* Translation matrix for elastic deformation */
L=DMatrix(3, 3);                /* Axis length change matrix from bcc to fcc */
T=DMatrix(3, 3);
fcc=DMatrix(3, 3);              /* fcc unit cell */
bcc=DMatrix(3, 3);              /* bcc unit cell */
bcci=DMatrix(3, 3);             /* Inverse of fcc unit cell */
R=DMatrix(3, 3);                /* Rotation matrix between fcc/bcc based on OR */
p2f=DMatrix(3, 3);              /* Transformation matrix */
tp1=DMatrix(3, 3);              /* Follows are temporary 3 by 3 matrices */
tp2=DMatrix(3, 3);
tp3=DMatrix(3,3);
tp4=DMatrix(3,3);

for(i=0; i<3; i++)
  for(j=0; j<3; j++)
  {
   tp1[i][j]=temp1[i][j];
   tp2[i][j]=temp2[i][j];
   tp3[i][j]=temp3[i][j];
   tp4[i][j]=temp4[i][j];
  }

/* Important change here */
rn=0.115;                       /* Range of two atoms needed for further calculation */
std=0.11*0.11;          /* Coherency criterion of NCS, using af as unit */

printf("Please enter the block size of matrix.\n");
scanf ("%d", &i1);
printf("Please enter the block size of precipitate.\n");
scanf ("%d", &i2);

  /* Making equal size in x, y, z directions for the block */
j1=i1;
k1=i1;

j2=i2;
k2=i2;

  /* Obtain the axis length change matrix between two crystals */
for(i=0; i<3; i++)
  for(j=0; j<3; j++)
  {
  if((i==0) && (j==0))
     L[i][j]=ab/af;
```

```
    else if((i==1) && (j==1))
    L[i][j]=ab/af;
    else if((i==2) && (j==2))
    L[i][j]=ab/af;
   else
    L[i][j]=0.0;
   }

  /* Follows are some matrix operation to finally obtain "p2f" */
Transpose(3,3,L,T);
Transpose(3,3,tp1,fcc);
Transpose(3,3,tp2,bcc);
Transpose(3,3,tp3,el);
Transpose(3,3,tp4,xyz);
InMatrix(3,xyz,xyzi);
InMatrix(3,bcc,bcci);
MatrixMult(3,3,fcc,3,3,bcci,R);
MatrixMult(3,3,R,3,3,T,p2f);

MatrixMult(3,3,el,3,3,xyzi,xyz2el);
InMatrix(3,xyz2el,el2xyz);

printf("p2f=");
printMatrix(3,3,p2f);
printf("xyz2el=");
printMatrix(3,3,xyz2el);

firstFatom = allocate();              /* make the head of the list */
thisPoint = firstFatom;
xmin1=floor(-i1/2.0);
xmax1=floor(i1/2.0);
ymin1=floor(-j1/2.0);
ymax1=floor(j1/2.0);
zmin1=floor(-k1/2.0);
zmax1=floor(k1/2.0);

/* Following locate all the atom positions for matrix */
for(i=xmin1; i<=xmax1; i++)
  for(j=ymin1; j<=ymax1; j++)
   for(k=zmin1; k<=zmax1; k++)
    {
      fi=i; fj=j; fk=k;
```

```
  /* Following consider atoms position change caused by
  stacking faults. Otherwise, this loop can be ignored  */
      if( fi+fj+fk > 0)
       {
       fi=i+1/12.0;
        fj=j+1/12.0;
        fk=k-1/6.0;
       }
       else if( fi+fj+fk < 0)
      {
       fi=i-1/12.0;
        fj=j-1/12.0;
        fk=k+1/6.0;
       }
  /* End of the loop */

     Join(fi, fj, fk, i1, j1, k1);
     Join(fi+1.0/2, fj+1.0/2, fk, i1, j1, k1);
      Join(fi, fj+1.0/2, fk+1.0/2, i1, j1, k1);
     Join(fi+1.0/2, fj, fk+1.0/2, i1, j1, k1);
   }
lengthFatom=length;

/* Following estimate the atoms position change caused by
  elastic strain. Otherwise, please skip.  */
thisPoint = firstFatom;
while(thisPoint->next!=NULL)
{
 for(n=0; n<3; n++)
 {
 if(n!=0)
    temp[n]=(xyz2el[n][0]*thisPoint->S[0]+xyz2el[n][1]*thisPoint->S[1]
          +xyz2el[n][2]*thisPoint->S[2]);
  else
   temp[n]=elastic1*(xyz2el[n][0]*thisPoint->S[0]

   +xyz2el[n][1]*thisPoint->S[1]
              +xyz2el[n][2]*thisPoint->S[2]);
  }
  for(n=0; n<3; n++)
  thisPoint->S[n]=(el2xyz[n][0]*temp[0]+el2xyz[n][1]*temp[1]
          +el2xyz[n][2]*temp[2]);
 thisPoint=thisPoint->next;
}
```

```
/* Following locate all the precipitate atom positions */
length=0;
firstPatom = allocate();
thisPoint = firstPatom;
xmin2=floor(-i2/2.0);
xmax2=floor(i2/2.0);
ymin2=floor(-j2/2.0);
ymax2=floor(j2/2.0);
zmin2=floor(-k2/2.0);
zmax2=floor(k2/2.0);
for(i=xmin2; i<=xmax2; i++)
  for(j=ymin2; j<=ymax2; j++)
   for(k=zmin2; k<=zmax2; k++)
    for(count=0; count<0.9; count=count+0.5)
    {
     fi=i+count; fj=j+count; fk=k+count;
      for(n=0; n<3; n++)
        temp[n]=(p2f[n][0]*fi+p2f[n][1]*fj+p2f[n][2]*fk);
     Join(temp[0], temp[1], temp[2], i1, j1, k1);
    }
lengthPatom=length;
printf("\nlengthFatom=%d  ", lengthFatom);
printf("lengthPatom=%d\n", lengthPatom);
fflush(stdout);

/* Following calculate the NCS sites */
length=0;
fitPoint=allocate();
thisPoint=fitPoint;
thisFatom=firstFatom;
worsePt=allocate();
thatPoint=worsePt;
for(i=1; i<=lengthFatom; i++)
{
  thisPatom=firstPatom;
  for(j=1; j<=lengthPatom; j++)
  {
  x=(thisFatom->S[0] - thisPatom->S[0]);
  y=(thisFatom->S[1] - thisPatom->S[1]);
    z=(thisFatom->S[2] - thisPatom->S[2]);
  if( (x<rn && x>-rn) && (y<rn && y>-rn) && (z<rn && z>-rn) )
    {
      distsq=(x*x +y*y +z*z);
```

```
        if(distsq <= std)
          {
           for(k=0; k<3; k++)
               thisPoint->S[k]=thisFatom->S[k];
                thisPoint->next = allocate();
               thisPoint=thisPoint->next;
                length++;
          }
       if(distsq >= worse)
         {
           for(k=0; k<3; k++)
             thatPoint->S[k]=thisFatom->S[k];
             thatPoint->next = allocate();
            thatPoint=thatPoint->next;
         }

     }
  thisPatom=thisPatom->next;
  }
  thisFatom=thisFatom->next;
}
/* End of Calculation */

/* Record the NCS referred to matrix index */
 fp=fopen("ncselsf.dat", "w");
 fpt=fopen("fitelsf.dat", "w");
 fprintf(fp, "The fcc size(total unit cells in one axis direction)
              in fcc/bcc is: %d\n", i1);
 fprintf(fp, "The total lattice points in fcc is: %d\n", lengthFatom);
 fprintf(fp, "The bcc size(total unit cells in one axis direction)
             in fcc/bcc is: %d\n", i2);
 fprintf(fp, "The total lattice points in bcc is: %d\n", lengthPatom);
 fprintf(fp, "The closest pair points representated by fcc are\n");

 /* Store all NCS sites in files as binary format */
 thisPoint=fitPoint;
 j=0;
 fprintf(fp, "{");
 for(i=0; i<length; i++)
 {
   if(j==3)
   {
    j=0;
```

```
      fprintf(fp,"\n");
    }
    fprintf(fp, "{%6.3f,", thisPoint->S[0]);
    fprintf(fp, "%6.3f,", thisPoint->S[1]);
    if(i==length-1)
      fprintf(fp, "%6.3f}", thisPoint->S[2]);
    else fprintf(fp, "%6.3f},", thisPoint->S[2]);
    j++;

    fprintf(fpt, " %6.3f", thisPoint->S[0]);
    fprintf(fpt, " %6.3f", thisPoint->S[1]);
    fprintf(fpt, " %6.3f", thisPoint->S[2]);

    thisPoint=thisPoint->next;
  }
  fprintf(fp, "}\n The number of total ncs is: %d", length);
  fclose(fp);
  fclose(fpt);

  /* Record the NCS as coherency degree you input */
  fp_worse=fopen("worselsf.dat", "w");
  thisPoint=worsePt;
  while(thisPoint->next != NULL)
  {
    fprintf(fp_worse, " %6.3f", thisPoint->S[0]);
    fprintf(fp_worse, " %6.3f", thisPoint->S[1]);
    fprintf(fp_worse, " %6.3f", thisPoint->S[2]);
    thisPoint=thisPoint->next;
  }
  fclose(fp_worse);
}



void   Join(i, j, k, in, jn, kn)
float i, j, k;
int in, jn, kn;
{
  extern struct corres *thisPoint;
  float xmin, xmax, ymin, ymax, zmin, zmax;
  xmin=floor(-in/2.0);
  xmax=floor(in/2.0);
```

```
  ymin=floor(-jn/2.0);
  ymax=floor(jn/2.0);
  zmin=floor(-kn/2.0);
  zmax=floor(kn/2.0);
  if( (i<xmin) || (i>xmax) || (j<ymin) || (j>ymax) || (k<zmin) || (k>zmax))
     ;
  else{
    thisPoint->S[0]=i;
    thisPoint->S[1]=j;
    thisPoint->S[2]=k;
    thisPoint->next = allocate();
    thisPoint=thisPoint->next;
    length++;
    }
}

struct corres * allocate()
{
    struct corres * newone;
    newone = (struct corres *) malloc (sizeof(struct corres));
     newone->next = NULL;
     return(newone);
}
```

```
/*
    plane.c

    using J.K.Chen's MatrixOperation.h

    Qiang Liang, T_Group, MSE, Va Tech, USA
    March, c1996

    Finding the array of NCS whose any two adjacent NCS distance
    is less than some value. The average plane of these NCS lying
    on over the whole simulation block can be treated as habit plane.
    Input data file comes from the ncsfbelsf.c or ncs718elds.c results.
    And it should be binary format.
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include </usr/users/jkchen/Program/MatrixOperation/MatrixOperation.h>

    /* matrix operation utility */

struct corres * allocate();
void find_points();
float Min_Dist();
float a, radius;
int delete();
struct corres
{   float      S[3];
   float    distance;
    struct  corres  *next;
};

struct corres *thisPoint, *thatPoint, *fitPoint, *Point,
              *directPoint,*planePoint, *tempPoint;
    /* Record the NCS information on best matching direction and plane */


main()
{
 int i, j, k, i1, del, num;
```

```
int NumPoint=0, MaxPoint=0, length=0, DirPoint;
float temp, bpoint1[3], bpoint2[3],dir[3], dir1[3], dirMax[3], dist,
      dist_min, density, distmin;
float planex, planey, planez, x, y, z;
char filename[20];
FILE *fp, *fpt;
struct corres *samePoint;

 /* Important here! Input the correct values for the system. */
printf("Please enter the lattice constant of matrix (fcc) .\n");
scanf ("%f", &a);

printf("Please enter the radius which contains any two adjacent NCS
   when their distance is less than that value .\n");
scanf ("%f", &radius);

printf("Please enter the block size of matrix .\n");
scanf ("%d", &i1);

printf("Please enter the data filename \n");
scanf( "%s", filename);

/* Input the data file and locate all NCS */
fp=fopen(filename, "r");
fitPoint=allocate();                    /* make the head of the list */
thisPoint=fitPoint;
while(!feof(fp))
{
  fscanf(fp, "%f", &temp);
  thisPoint->S[0]=temp;
  fscanf(fp, "%f", &temp);
  thisPoint->S[1]=temp;
  fscanf(fp, "%f", &temp);
  thisPoint->S[2]=temp;
  thisPoint->next=allocate();
  thisPoint=thisPoint->next;
  length++;
}
printf("length=%d", length);
fflush(stdout);

/* Finding the plane consisting of any two adjacent NCS less than radius */
planePoint=allocate();              /* make the head of the list */
thatPoint=planePoint;
```

```
 thatPoint->S[0]=0; thatPoint->S[1]=0; thatPoint->S[2]=0;
 thatPoint->next=allocate();
 thatPoint=thatPoint->next;
 find_points(0.0, 0.0, 0.0);

 /* Store the proper NCS in file as binary format*/
 fp=fopen("plane.dat", "w");
 NumPoint=0;
 i=1;
 thisPoint=planePoint;
 while(thisPoint->next!=NULL)
 {
   if(i==1)
   {
     fprintf(fp, "%4.1f %4.1f %4.1f ", thisPoint->S[0],

   thisPoint->S[1], thisPoint-> S[2]);
     i++;
    }
    else if(i==2)
    {
     fprintf(fp, "%4.1f %4.1f %4.1f ", thisPoint->S[0],

   thisPoint->S[1], thisPoint-> S[2]);
    i++;
    }
    else if(i==3)
    {
     fprintf(fp, "%4.1f %4.1f %4.1f \n", thisPoint->S[0],

   thisPoint->S[1], thisPoint-> S[2]);
     i=1;
    }
    NumPoint++;
  thisPoint=thisPoint->next;
 }
 fclose(fp);
}

 /* Function for finding the satisfied NCS under requirment */
void find_points(xx,yy,zz)
float xx, yy, zz;
{
  int k;
  float x, y, z;
```



```
  float dist;
```

```
  struct corres *FthisPoint;
  FthisPoint=fitPoint;
  while(FthisPoint->next!=NULL)
  {
    if(delete(FthisPoint->S[0], FthisPoint->S[1], FthisPoint->S[2])!=1)
    {
      x=FthisPoint->S[0]-xx;
      y=FthisPoint->S[1]-yy;
      z=FthisPoint->S[2]-zz;
      dist=a*a*(x*x+y*y+z*z);
      if( dist < (radius*radius))
      {
        for(k=0; k<3; k++)
         thatPoint->S[k]=FthisPoint->S[k];
        thatPoint->next=allocate();
        thatPoint=thatPoint->next;
        find_points(FthisPoint->S[0], FthisPoint->S[1], FthisPoint->S[2]);
      }
    }
    FthisPoint=FthisPoint->next;
  }
}


struct corres * allocate()
{
    struct corres * newone;  /* Set list pointer */
    newone = (struct corres *) malloc (sizeof(struct corres));
     newone->next = NULL;
     return(newone);
}


int delete(xx, yy, zz)
float xx, yy, zz;
{
    struct corres *FPoint;
    /* Set list pointer */
     int del;
    FPoint=planePoint;
     del=0;
    while(FPoint->next!=NULL)
    {
      if((xx!=FPoint->S[0]) ||
        (yy!=FPoint->S[1]) ||
```

```
        (zz!=FPoint->S[2]))
        ;
        else {
              del=1; break;
        }
         FPoint=FPoint->next;
    }
     return(del);
}
```

==========

==========

```
/*
    bmd.c
    using J.K.Chen's MatrixOperation.h

    Qiang Liang, T_Group, MSE, Va Tech, USA
    March, c1996

    Starting from origin, the program sorts out a line of NCS whose any
    two adjacent NCS in the line is the shortest compared to all other
    NCS. This line can be treated as the best matching direction over
    the whole simulation block. Input data file comes from the
    ncsfbelsf.c or ncs718elds.c results. And it should be binary format.


*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include </usr/users/jkchen/Program/MatrixOperation/MatrixOperation.h>

struct corres * allocate();
float Min_Dist();
int delete();
struct corres
{   float   S[3];
   float   distance;
   struct  corres  *next;
};
```

```
/* Record the NCS information on best matching direction */
struct corres *thisPoint, *thatPoint, *fitPoint, *Point, *directPoint, *roadPoint,
              *tempPoint;

main()
{
 int i, j, k, i1, del, num;
 int NumPoint=0, MaxPoint=0, length=0, DirPoint;
 float temp, bpoint1[3], bpoint2[3],dir[3], dir1[3], dirMax[3],
    dist, dist_min, density, distmin;
 float planex, planey, planez, x, y, z;
 char filename[20];
 FILE *fp, *fpout, *fpt;
 struct corres *samePoint;

 printf("Please enter the block size of matrix .\n");
 scanf ("%d", &i1);

 printf("Please enter the data filename \n");
 scanf( "%s", filename);

 printf("Please enter the number of NCS you want in the line \n");
 scanf("%d", &num);

   /* Locate the NCS from input data file */
 fp=fopen(filename, "r");
 fitPoint=allocate();
   /* Make the head of list */
 thisPoint=fitPoint;
 while(!feof(fp))
 {
   fscanf(fp, "%f", &temp);
   thisPoint->S[0]=temp;
   fscanf(fp, "%f", &temp);
   thisPoint->S[1]=temp;
   fscanf(fp, "%f", &temp);
   thisPoint->S[2]=temp;
   thisPoint->next=allocate();
   thisPoint=thisPoint->next;
   length++;
 }
 printf("length=%d", length);
 fflush(stdout);
```

```
/* Open output data file */
fpout=fopen("line.dat", "w");
fprintf(fpout, "The total unit cells in the matrix is %d\n", i1);
fprintf(fpout, "The %d ncs points on the road are: \n", num);

/* Finding the line consisting of nearest NCS at any two adjacent ones */
roadPoint=allocate();
thatPoint=roadPoint;

thatPoint->S[0]=0; thatPoint->S[1]=0; thatPoint->S[2]=0;
fprintf(fpout, "(%5.1f, %5.1f, %5.1f), %5.1f\n", 0,0,0,0);
for(i=0; i<num; i++)
{
  dist_min=1000;
  thisPoint=fitPoint;
  while(thisPoint->next!=NULL)
  {
   del=0;
    x=thisPoint->S[0]-thatPoint->S[0];
    y=thisPoint->S[1]-thatPoint->S[1];
    z=thisPoint->S[2]-thatPoint->S[2];
   dist=x*x+y*y+z*z;
    if(dist!=0)
      if(dist<dist_min)
     {
        del=delete(thisPoint->S[0], thisPoint->S[1], thisPoint->S[2]);
      if(del!=1)
          {
            dist_min=dist;
             for(k=0; k<3; k++)
                  bpoint1[k]=thisPoint->S[k];
          }
     }
    thisPoint=thisPoint->next;
  }

  thisPoint=fitPoint;
  samePoint=allocate();
  Point=samePoint;
  Point->S[0]=bpoint1[0]; Point->S[1]=bpoint1[1]; Point->S[2]=bpoint1[2];
  Point->next=allocate();
  Point=Point->next;
  while(thisPoint->next!=NULL)
  {
    x=thisPoint->S[0]-thatPoint->S[0];
```

```
      y=thisPoint->S[1]-thatPoint->S[1];
      z=thisPoint->S[2]-thatPoint->S[2];
      dist=x*x+y*y+z*z;
      if(dist!=dist_min) ;
      else
     {
       del=delete(thisPoint->S[0],thisPoint->S[1], thisPoint->S[2]);
        if(del!=1 && ((thisPoint->S[0]!=bpoint1[0]) &&
         (thisPoint->S[1]!=bpoint1[1]) &&  (thisPoint->S[2]!=bpoint1[2]))
                                                                    )
        {
        for(k=0; k<3; k++)
          Point->S[k]=thisPoint->S[k];
         Point->distance=Min_Dist(Point->S[0], Point->S[1], Point->S[2]);
         Point->next=allocate();
         Point=Point->next;
        }
     }
      thisPoint=thisPoint->next;
    }


    Point=samePoint;
    Point=Point->next;
    distmin=1000;
    while(Point->next!=NULL)
     {
     if(Point->distance<distmin)
     {
       distmin=Point->distance;
        for(k=0; k<3; k++)
           bpoint1[k]=Point->S[k];
     }
     Point=Point->next;
     }

   fprintf(fpout, "(%5.1f, %5.1f, %5.1f), %5.1f\n", bpoint1[0],
                  bpoint1[1], bpoint1[2], dist_min);

   thatPoint->next=allocate();
   thatPoint=thatPoint->next;
   for(k=0; k<3; k++)
    thatPoint->S[k]=bpoint1[k];
}
fclose(fpout);
```

```
 /* Open a data record file as binary format */
 fp=fopen("view.dat", "w");
 NumPoint=0;
 i=1;
 thisPoint=roadPoint;
 while(thisPoint!=NULL)
 {
    if(i==1)
    {
      fprintf(fp, "%4.1f %4.1f %4.1f ", thisPoint->S[0],

    thisPoint->S[1],thisPoint-> S[2]);
      i++;
     }
     else if(i==2)
     {
      fprintf(fp, "%4.1f %4.1f %4.1f ", thisPoint->S[0],

    thisPoint->S[1], thisPoint-> S[2]);
     i++;
     }
     else if(i==3)
     {
      fprintf(fp, "%4.1f %4.1f %4.1f \n", thisPoint->S[0],

    thisPoint->S[1], thisPoint-> S[2]);
      i=1;
     }
     NumPoint++;
   thisPoint=thisPoint->next;
 }
 fclose(fp);
}


struct corres * allocate()
{
    struct corres * newone;
    newone = (struct corres *) malloc (sizeof(struct corres));
     newone->next = NULL;
     return(newone);
}
```

```
/* Function finds nearest two adjacent NCS */
float Min_Dist(xx, yy, zz)
float xx, yy, zz;
{
    struct corres *FthisPoint, *FPoint;
    int del, k;
    float x, y, z, dist, dist_min, bpoint1[3];
    FthisPoint=fitPoint;
    while(FthisPoint!=NULL)
    {
     del=0;
      x=FthisPoint->S[0]-xx;
      y=FthisPoint->S[1]-yy;
      z=FthisPoint->S[2]-zz;
     dist=x*x+y*y+z*z;
      if(dist!=0)
        if(dist<dist_min)
       {
          del=delete(FthisPoint->S[0],FthisPoint->S[1],FthisPoint->S[2]);
           if(del!=1)
              {
               dist_min=dist;
                 for(k=0; k<3; k++)
                    bpoint1[k]=thisPoint->S[k];
              }
       }
      FthisPoint=FthisPoint->next;
    }
    return(dist_min);
}

 /* Function deletes previous NCS after each step finding nearest NCS */
int delete(xx, yy, zz)
float xx, yy, zz;
{
    struct corres *FPoint;
     int del;
    FPoint=roadPoint;
     del=0;
    while(FPoint->next!=NULL)
    {
      if((xx!=FPoint->S[0]) ||
        (yy!=FPoint->S[1]) ||
        (zz!=FPoint->S[2]))
      ;
```

```
        else{
          del=1; break;
        }
         FPoint=FPoint->next;
    }
     return(del);
}
```

The vita has been removed from the scanned document