MODEL-ASSISTED NONDESTRUCTIVE EVALUATION FOR MICROSTRUCTURE QUANTIFICATION

Thesis

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Materials Engineering

By

Darius Rayshawn Johnson

Dayton, Ohio

May, 2015
MODEL-ASSISTED NONDESTRUCTIVE EVALUATION FOR MICROSTRUCTURE QUANTIFICATION

Name : Johnson, Darius Rayshawn

APPROVED BY :

______________________________  ________________________________
Charles E. Browning, Ph.D.                           James L. Blackshire, Ph.D.
Advisory Committee Chair                                  Research Advisor
Chair of Chemical & Materials                                          Senior Materials Engineer
Engineering Department                                                 Air Force Research Laboratory

______________________________  ________________________________
P. Terrence Murray, Ph.D.   Eddy M. Rojas, Ph.D., M.A.,P.E.
Committee member                                      Dean
Professor                                                      School of Engineering
Chemical & Materials Engineering                           School of Engineering
Department
ABSTRACT

MODEL-ASSISTED NONDESTRUCTIVE EVALUATION FOR MICROSTRUCTURE QUANTIFICATION

Name: Johnson, Darius Rayshawn
University of Dayton

Advisor: Dr. Charles E. Browning

Modern computational tools are permitting realistic complex 2-dimensional (2D) and 3-dimensional (3D) geometry structures, material state properties, and multi-physics realism to be included into Computational Nondestructive Evaluation Models (CNDE), which allows a direct comparison of local material property statistics with sensing model results. The goal of this research was to develop and demonstrate ultrasound model-assisted nondestructive evaluation (NDE) methods for characterizing and mapping 2D/3D microstructures. A framework was created using the concept of Integrated Computational Materials Engineering (ICME) that allows for the incorporation of real material data sets to be described explicitly within computational NDE models. The Framework was tested using real and synthetically generated 2D/3D material data sets, where material state properties were characterized and correlated with NDE model sensing results. The implications of research are that the development of the framework is now allowing for studies to observe and understand complex elastic wave scattering due to polycrystalline microstructures.
This thesis is dedicated to my mother, father and brother.
ACKNOWLEDGMENTS

I am grateful for the opportunity to have continued my educational endeavors. There have been mentors, friends and family along my journey that have motivated me to continue to pursue a passion and dream. The University of Dayton has provided a setting for me to grow into a young leader. Through my undergraduate and graduate trek, I’ve grown mentally, spiritually, academically, mentorships and friendships. I give thanks to the University of Dayton for this experience. I would like express great gratitude to Dr. Browning for support, encouragement and most importantly believing in me, Thank you. I am grateful for the wise words, advice, love and support of Mr. Cofield, the MEP and MLP Programs. One of the greatest opportunities has been to conduct research under the guidance Dr. Blackshire at Wright Patterson Air Force Base (WPAFB). Dr. Blackshire has been a teacher, mentor, and friend, over the past four years. I’ve grown into a young research engineer, which I attribute to him. I give special thanks to Dr. Adam Pilchak assisting in my research project and providing a wealth of knowledge. I also thank Dr. Joseph Tucker, Dr. Michael Uchic, Dr. Michael Groeber, Aaron Cherry, Matt Cherry, and Josiah Dierken for being great resources at WPAFB. Lastly I would like to thank my family and Ms. Tessa Terrell for being my biggest motivators and supporters.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................................................... iv

DEDICATION ......................................................................................................................................... v

ACKNOWLEDGMENTS ....................................................................................................................... vi

LIST OF FIGURES .................................................................................................................................. xii

LIST OF TABLES ..................................................................................................................................... xviii

LIST OF SYMBOLS, ABBREVIATIONS AND SPECIAL NOMENCLATURE ........................................... xix

1. INTRODUCTION ............................................................................................................................... 1
   1.1) Motivation ................................................................................................................................... 1
   1.2) Objective of Research ................................................................................................................. 3
   1.3) Introduction of Complex Polycrystalline Structures ................................................................. 5
   1.4) Introduction of Ultrasound NDE Technique and Key Concepts ............................................. 7
   1.5) Introduction of Finite Element Models and Computational NDE ......... 13

2. ICME FRAMEWORK ......................................................................................................................... 15
   2.1) Chapter Introduction and Background ...................................................................................... 15
      2.1.1) Computational NDE .............................................................................................................. 15
      2.1.2) Integrated Computational Materials Engineering ............................................................. 15
   2.2) Framework Objective.................................................................................................................... 18
4.3.1) Whole Sample Analysis-MTR Ti ................................................ 66
4.3.2) Characterization of Segmented Regions .............................. 69
4.4) 3-Dimensional Microstructures Quantification ................................. 71

5. NDE MODEL ................................................................................................. 74
   5.1) Chapter Introduction and Background ........................................ 74
   5.2) General Model Goals and Conditions ....................................... 74
   5.3) LSHR Nickel Specific Models and Highlighted Results .............. 77
   5.4) MTR Titanium Alloy Specific Models and Highlighted Results .... 83
   5.5) 3-Dimensional Microstructure Samples ..................................... 89
       5.5.1) Synthetic Equiaxed Models .............................................. 89
       5.5.2) In100 Models ................................................................. 90

6. RESULTS AND DISCUSSION ...................................................................... 92
   6.1) Introduction .................................................................................... 92
   6.2) LSHR Results ................................................................................ 92
       6.2.1) Summary of Microstructure Quantification ...................... 93
       6.2.2) LSHR Nickel Model Results ........................................... 98
       6.2.3) Discussion and Correlation .............................................. 100
   6.3) F110 Micro-Textured Titanium Sample ....................................... 102
       6.3.1) Summary of Microstructure Quantification ...................... 102
       6.3.2) MTR Titanium Model Results ........................................ 106
       6.3.3) Discussion and Correlation .............................................. 108
   6.4) 3D Sample Results ........................................................................ 111
       6.4.1) Synthetic 3D Microstructure ............................................ 111
D3: 3 Dimensional Samples Pitch-Catch Base Model ............................................. 172

E. MICROSTRUCTURE CHARACTERIZATION .......................................................... 177

E1: Nickel Based Super Alloy 1000X8000um Segmentation Characterization ..... 178

E2: MTR Titanium Alloy 1000X8500um Segmentation Characterization ....... 188
LIST OF FIGURES

Figure 1.1: A schematic of the tailored Nickel based super alloy turbine disk [2] ...................... 2

Figure 1.2: Overall research objectives. .......................................................................................... 5

Figure 1.3: LSHR nickel (top) and MTR titanium (bottom) inverse pole figures and reference material properties used in the NDE sensing models .................................................. 7

Figure 1.4: A schematic of the propagation of longitudinal and shear waves. ............................ 8

Figure 1.5: A depiction of an incident wave reflecting and transmitting at the boundary interface of two media such as grain boundaries .............................................................................. 10

Figure 1.6: A depiction of an incident wave reflecting and transmitting at the boundary as well as mode converting from longitudinal wave to shear waves ........................................... 11

Figure 1.7: Elastic wave interactions within the Rayleigh, stochastic, and geometric scattering regimes [18]. .................................................................................................................. 12

Figure 1.8: Backscattering of an elastic wave in a polycrystalline structure, and characteristic ultrasound signal highlighting the backscatter data of the signal. ...................... 12

Figure 2.1: A viewgraph of the four major aspects of ICME highlighting computational methods for microstructure-property relationships [22]. .......................................................... 17

Figure 2.2: A viewgraph of crystallographic orientation as it pertains to element orientations described by Euler angles ($\phi_1, \phi, \phi_2$) (left). A polycrystalline material representing the crystallographic orientations of the grains (right) [26 28]. ............................................... 21

Figure 2.3: Viewgraphs depicting a typical set-up of EBSD measurements and measured diffraction patterns. Diffraction patterns help to determine orientation and boundaries as viewed in micrographs and inverse pole figures (far right) [24]. .............................................. 21
**Figure 2.4:** Representative EBSD maps and analysis using EDAX TSL OIM software to map out the microstructure and extrapolate data such as misorientation distributions and pole figures. ................................................................................................................................ 22

**Figure 2.5:** DREAM.3D data structures including: cell data, field data, and ensemble data [30]. ..................................................................................................................................... 23

**Figure 2.6:** The graphical user interface of Dream.3d software. A green outline highlights the filter library. A blue outline highlights the list of all available filters. A red outline highlights the pipeline of filters that are in current use, and orange highlights any errors and progress when the software is running. .............................................................................. 24

**Figure 2.7:** A set of DREAM.3D pipelines used to take EBSD orientation file and do analysis within DREAM.3D. Pipeline 1 (left) imports the orientation files from the TSL EDAX OIM software into the DREAM.3D environment. Pipeline 2 (right) gathers data on the spatial coordinates, grain sizes, mis-orientations, and Euler angles, and output this information to be used as an input to third portion of the framework involving Matlab software code. ............................................................................................................................ 26

**Figure 2.8:** a. The graphical user interface (GUI) of DREAM.3D’s software StatsGenerator.exe. The interface shows a statistical grain size distribution with controls to vary the distribution width, height, size and cut off values. b. Representative synthetically generated microstructures are depicted for 3D equiaxed grain microstructure (bottom left) and elongated grain microstructure (bottom right). ................... 28

**Figure 2.9:** A viewgraph of a 6x6 crystallographic stiffness matrix highlighting the 21 value positions used to describe materials in FEM models.................................................................29

**Figure 2.10:** A viewgraph of sample matlab text output describing xyz position of each element, grain id and 21 value stiffness matrix. The text File is used as an aide to describe polycrystalline materials. ............................................................................................................................ 30

**Figure 2.11:** A viewgraph of “REGN” PZflex command and variables required for command to function successfully. ............................................................................................. 31

**Figure 2.12:** A sample output text file from the matlab code “Cij Spatial Coordinates 2D M-file used as an input file to PZflex FEM model to describe the spatial coordinates of every node/element in NDE Sensing Models for a 2-dimesional microstructure. .................... 31

**Figure 2.13:** A viewgraph of “Prop” PZflex command and variables required for command to function successfully. ............................................................................................. 32

**Figure 2.14:** A sample output text file from the matlab code “Cij Stiffness Matrix IDs M-file”/“Cij Stiffness Matrix IDs 3D M-file” used as an input file to PZflex FEM model to describe the crystallographic properties every grain in NDE Sensing Models. .......................... 32
Figure 2.15: Progression of steps in order to create PZflex ultrasound FEM sensing model [20].................................................................................................................................. 34

Figure 2.16: Progression of the ICME framework taking a polycrystalline microstructure, generating an FEM mapping, assigning material properties, and finally conducting ultrasound wave propagation, scattering and sensing model [2]...................................................... 34

Figure 2.17: The ICME framework providing a means for incorporating realistic and synthetic polycrystalline microstructures into NDE ultrasonic FEM sensing models.................. 36

ICME Framework Example: Step By Step Example with screenshots of approach ................. 37

Figure 3.1: Schematic diagram of a turbine disk and cut away section identifying microstructure features of the LSHR nickel sample (top figure) [39], and the Inverse Pole Figure (IPF) mapping of the LSHR Nickel super alloy grains and grain boundaries (bottom image)........................................................................................................................................ 48

Figure 3.2: The Inverse Pole Figure (IPF) mapping of the Micro Textured Region (MTR) titanium alloy sample showing grains and grain boundaries............................................ 50

Figure 3.3: Graphical user interface (GUI) of the StatsGenerator.exe program used to generate a synthetic equiaxed 3D microstructure. .............................................................................. 52

Figure 3.4: StatsGenerator.exe resulting 3D microstructure with visualization of the X+,Y+,Z+ faces ................................................................................................................................. 52

Figure 3.5: Viewgraph depicting the IN 100 structure with visualizations of the X+,Y+,Z+ faces........................................................................................................................................... 53

Figure 4.1: a. Unique grain ID images showing an example of an identified twinned boundary, and twin removal clean up on same grain. b. Unique grain ID images showing the LSHR nickel data with twinned boundaries in some grains, and c. Unique grain ID images LSHR nickel structure after the twin removal cleanup process............................................. 57

Figure 4.2: Cropped 1000um X 8000um segment regions of the LSHR nickel super alloy sample........................................................................................................................................ 60

Figure 4.3: Log normal fits of three segmented grain locations (0-1000um, 6000-7000um, 18000-19000um), for area fraction, grain size distribution, and misorientation. ...... 62

Figure 4.4 Chart summarizes Mean Grain Size for lognormal equation for twined and untwined and actual mean grain size for twinned and untwined cases............................... 64

Figure 4.5: a. the original sample without clean up b. Viewgraph depicts the cleaned and cropped sample used for further analysis c. IPF Color Scale.............................................................................. 65
Figure 4.6 a. Pole Figure of whole sample b. Hand segmentation of the MTR Titanium sample into thirteen key regions.

Figure 4.7 Pole figures for the thirteen segmented regions showing dominant orientations within each region.

Figure 4.8 IPF maps of the 49 segmented regions of the MTR titanium alloy.

Figure 4.9 Pole figures (PF) for four specified locations (0-1000, 2000-3000, 11000-12000, 19000-2000), quantification of orientation.

Figure 4.10 Grain quantification of synthetic 3-dimensional equiaxed nickel microstructure highlighting a. grain size distribution, b. area fraction, and c. Misorientation.

Figure 4.11 Grain quantification of the Inconel 100 3d volume highlighting a. Grain Size Distribution b. Area Fraction c. Misorientation.

Figure 5.1. PZFlex ultrasonic model conditions for the pulse-echo and pitch-catch segmented microstructure models.

Figure 5.2 a. 5MHz longitudinal wave propagation through the small grain region segment at different model time steps, b. the transition region, and c. the coarse grain region. Each model shows subtle scattering occurring after the initial wave.

Figure 5.3. Representative y-displacement pulse-echo signals for the LSHR nickel ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.

Figure 5.4. Representative y-displacement pitch-catch signals for the LSHR nickel ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.

Figure 5.5 a. The maximum displacement of the elastic waves in x and y directions and pressure for the small grain, b. transition grain, and c. coarse grain regions.

Figure 5.6 a. Wave propagation through a titanium region with minor texture shown at different time steps, b. a texture region with indications of a small macro grain, c. a large well-defined macro grain, d. and two regions of texture.

Figure 5.7. Representative y-displacement pulse-echo signals for the MTR titanium ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.
Figure 5.8. Representative y-displacement pitch-catch signals for the MTR titanium ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz................................................................................................................................. 87

Figure 5.9 a. Maximum displacements of the wave in x and y directions and pressure for the minor texture region, b. small macro grain region, c. large preferred orientation macro grain region, d. and two textured regions........................................................................... 88

Figure 5.10 a. Time-resolved propagation of a longitudinal wave in the y-direction for the synthetic nickel microstructure 3-dimensional volume, and b. maximum displacements of the wave on the opposite face of propagation direction................................. 89

Figure 5.11 a. Time-resolved propagation of a longitudinal wave across the x plane and peak outer surface displacement of wave, and b. maximum displacement of the wave on a internal surface with the model.......................................................................................... 91

Figure 6.1. Segmentation of the LSHR nickel sample and example segment analysis........ 93

Figure 6.2: a. Position vs. Number of Grains, b. Position vs. As-Large-As Grain, and c. Position vs. Mean Grain Size......................................................................................................................... 97

Figure 6.3: a. Position vs. Average Misorientation, and b. Position vs. Largest Misorientation ........................................................................................................................................ 97

Figure 6.4: Pulse-echo average signal peak-to-peak analysis and standard deviation for NDE sensing models for LSHR nickel microstructure............................................................ 99

Figure 6.5: Pitch-catch average signal peak-to-peak analysis and standard deviation for NDE sensing models for the MTR nickel microstructure...................................................................................... 100

Figure 6.6: Correlation of 5 MHz and 10 MHz signal analysis with mean grain size statistics and as-large-as grain feature sizes............................................................................................... 101

Figure 6.7: Correlation of 5 MHz and 10 MHz signal analysis with number of grains per segment. .................................................................................................................................................. 102

Figure 6.8. Segmentation of the MTR titanium sample and example segment analysis. ........ 103

Figure 6.9: Identified key MTR regions in titanium microstructure and associated pole figures. ........................................................................................................................................... 104

Figure 6.10: Peak intensity of dominant orientation for segmented locations. .................... 105

Figure 6.11: Pulse-echo average signal peak-to-peak analysis and standard deviation for NDE sensing models for the MTR titanium microstructure.................................................................................. 106
Figure 6.12: Pitch-catch average signal peak-to-peak analysis and standard deviation for NDE sensing models for the MTR titanium microstructure.

Figure 6.13: Correlation of 5 MHZ signal Analysis with geometric angles of grains and signal.

Figure 6.14: Correlation of 10 MHz signal analysis with peak-to-peak signal analysis and dominate orientation intensities.

Figure 6.15: Segment in the 11,000 um region with pole figures, wave propagation for 10MHz, and peak y-displacement amplitude levels.

Figure 6.16: a. The 2D surface on the opposite side of the 3D grain models. b. Model Y-displacement showing lobbing of energy correlation to grain sizes. c. Grain size distribution for sample.

Figure 6.17 a. 3D model representation b. 2D Midplane of sample. c. The 2D surface on the opposite side of the 3D grain models d. Model Y-displacement showing lobbing of energy correlation to grain sizes. e. Grain Size Distribution for Sample.

Appendix E. Microstructure Characterization for segmented regions of the a. Nickel Based Super Alloy and b. MTR Titanium Alloy.
LIST OF TABLES

Table 2.1: A table listing the computational tools used in the ICME framework and their capabilities within the scope of the research. .......................................................... 19

Table 4.1: Table comparing dimensional data for the LSHR Nickel sample in the cases of twin and twin-removed boundaries present ......................................................... 59

Table 4.2 Summary chart of log normal fit parameters for LSHR nickel segmented positions ......................................................................................................................... 63

Table 4.3 Chart detailing number of points within each hand segmented region .............. 68

Table 4.4 Chart detailing dominant crystallographic orientations for each of the thirteen regions. ......................................................................................................................... 69

Table 5.1 Chart detailing the amount and types of models ran on the microstructure data sets. All together 520 models were run and used in the thesis studies .............................. 75

Table 6.1: Table showing calculated statistical data of dimensional and misorientation features for segmented LSHR nickel super alloy with twin boundaries remaining .............. 94

Table 6.2: Table showing calculated statistical data of dimensional and misorientation features for segmented LSHR nickel super alloy without twin boundaries ........................ 95

Table 6.3 Summary table of dominant orientations for all 49 segments in the MTR titanium microstructures ........................................................................................................ 104
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDE</td>
<td>Non Destructive Evaluation</td>
</tr>
<tr>
<td>CNDE</td>
<td>Computational Non Destructive Evaluation</td>
</tr>
<tr>
<td>ICME</td>
<td>Integrated Computational Materials Engineering</td>
</tr>
<tr>
<td>MGS</td>
<td>Mean(Average) Grain Size</td>
</tr>
<tr>
<td>GSD</td>
<td>Grain Size Distribution</td>
</tr>
<tr>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>kHz</td>
<td>kilohertz</td>
</tr>
<tr>
<td>um</td>
<td>Microns or Micrometers</td>
</tr>
<tr>
<td>$V$</td>
<td>Velocity</td>
</tr>
<tr>
<td>$\alpha, \beta$</td>
<td>Angle of Incidents</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$D$</td>
<td>Distance</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength</td>
</tr>
<tr>
<td>$f$</td>
<td>Frequency</td>
</tr>
<tr>
<td>$d$</td>
<td>Grain Size Diameter</td>
</tr>
<tr>
<td>$Z$</td>
<td>Impedance</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reflected Energy</td>
</tr>
<tr>
<td>$Te$</td>
<td>Transmitted Energy</td>
</tr>
<tr>
<td>EBSD</td>
<td>Electron Backscatter Detection</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SEM</td>
<td>Scanning Electron Microscope</td>
</tr>
<tr>
<td>OIM</td>
<td>Orientation Imaging Microscopy</td>
</tr>
<tr>
<td>DREAM.3D</td>
<td>Digital Representation Environment for Analysis of Microstructure in 3D</td>
</tr>
<tr>
<td>FEM</td>
<td>Finite Element Modeling</td>
</tr>
<tr>
<td>2D</td>
<td>Two dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>LSHR</td>
<td>Low Solvus High Refractory</td>
</tr>
<tr>
<td>DMHT</td>
<td>Dual Microstructure Heat Treatment</td>
</tr>
<tr>
<td>MTR</td>
<td>Micro Textured Region(s)</td>
</tr>
<tr>
<td>IPF</td>
<td>Inverse Pole Figure</td>
</tr>
<tr>
<td>C.I</td>
<td>Confidence Index</td>
</tr>
<tr>
<td>( xc )</td>
<td>Mean</td>
</tr>
<tr>
<td>( w )</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>( A )</td>
<td>Area</td>
</tr>
<tr>
<td>( y_o )</td>
<td>Y Offset</td>
</tr>
<tr>
<td>( V_l )</td>
<td>Longitudinal Velocity</td>
</tr>
<tr>
<td>( V_{bl} )</td>
<td>Longitudinal Velocity (Bulk)</td>
</tr>
<tr>
<td>( V_t )</td>
<td>Traverse Velocity</td>
</tr>
<tr>
<td>( E )</td>
<td>Young’s Modulus</td>
</tr>
<tr>
<td>( G )</td>
<td>Shear Modulus</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>( \varphi_1, \phi, \varphi_2 )</td>
<td>Euler Bunge Angles</td>
</tr>
<tr>
<td>P.F</td>
<td>Pole Figure</td>
</tr>
</tbody>
</table>
1.1) Motivation

The mission of research done at the Air Force Research Laboratory is to develop and sustain technologies associated with the war fighter and men and women of the Air Force [1]. Key to sustaining the fleet is the concept of nondestructive evaluation (NDE). As the name suggests, NDE uses techniques to characterize a material in ways that are noninvasive or non-damaging to the sample or system. With regards to current and emerging Air Force materials, this characterization includes describing the properties of a material system, in addition to detecting mission critical flaws, cracks, and voids that can compromise the integrity of an aircraft to function in a safe and effective manner.

In the most critical parts of an aircraft, structural integrity is critical for its function, for example in the turbine disc section of an engine. Modern turbine disks are made of super-alloy materials, which can include engineered, tailored microstructures. The microstructure depicted in Figure 1.1, for example, is tailored so that the grain size varies based on the position and specific properties needed in a particular location. At the disk rim the microstructure’s average grain size is approximately 100um, which helps with fatigue damage resistance, whereas in the disk bore section, the average grain size is 10 um, which helps with creep resistance. The
integrity of these material properties and structure must remain the same in the initial and evolution of use during the lifetime of the disc. The NDE based technique of ultrasound can aide in the evaluation of the turbine disk material properties. Microstructure quantification can be achieved with ultrasonic techniques by studying the elastic wave interactions with the material and the correlating scattering response with material property features of interest such as grain size or misorientation.

Figure 1.1: A schematic of the tailored Nickel based super alloy turbine disk [2].

The widespread use of ultrasound NDE for microstructure characterization has not been realized in many practical applications, because of the difficulty in correlating material properties with the complex scattering phenomena of elastic waves occurring in bulk materials. In previous research there has not been a need to further validate the microstructure of a material, but with the increasing usage of complex materials, such as dual grain nickel based
super alloys and the micro textured titanium alloys used in aircraft it is imperative to validate the microstructure and properties of the material for their intended use.

Modern computational models are permitting realistic material state properties, complex 3-dimensional geometry structures, and multi-physics realism to be included in the model framework, which permits direct comparisons of realistic properties and conditions to be evaluated and compared for model verification and validation purposes [3,4]. Ultrasound sensing models provide virtual representations of realistic complex microstructures, which allow the observation and study of complex scattering of elastic waves. This can be used to aid in the understanding of ultrasonic wave interactions with complex polycrystalline materials, and can be used to study and address many of the problems identified in the previous paragraph. Computational NDE also provides a way to verify NDE characterization methods with ground–truth quantified microstructures and can do so in a much more rapid and efficient way compared to traditional experimentation.

1.2) Objective of Research

The overall objective of the research was to develop a Computational Nondestructive Evaluation (CNDE) framework that builds on the recently described concept of Integrated Computational Materials Engineering (ICME) [3,5,6]. The technical definition of ICME is as follows, “the integration of materials information, captured in computational tools, with engineering product performance analysis and manufacturing-process simulation” [6]. The ICME concept will be described in more detail in Chapter 2 of the thesis, but very briefly involves the use of interconnected computational and experimental methods to enhance material property-performance optimization. In the present context, the ICME concept is being utilized to study ultrasonic NDE sensing interactions with realistic microstructures, with the ultimate goal of
developing and using model-assisted methods to better estimate material state properties using NDE sensing approaches.

As shown in Figure 1.2, the research involved three well defined and interconnected research specific tasks. The first research specific task involved the development of a framework that could take key microstructure information and input it directly into NDE sensing models, where material property information could be accounted for on an element-by-element basis in the model. This permits the defining and use of realistic and accurate boundary conditions that are needed to simulate realistic interactions of ultrasound waves with complex polycrystalline microstructures. The second research specific task used the new framework to systematically study ultrasonic wave interactions and scattering processes with two representative turbine engine microstructures, and to quantify and correlate the microstructure properties with the ultrasonic model prediction results. The material state properties of interest included mean grain size (MGS), grain size distribution (GSD), and misorientation for a Low-Solvus, High-Refractory (LSHR) nickel super-alloy material, and a textured titanium super-alloy material with macro-texture regions (MTRs). The framework allowed realistic microstructure information obtained using Electron Backscatter Diffraction (EBSD) measurements to be incorporated directly into the ultrasonic forward sensing models, where elastic wave propagation and scattering could be combined with microstructure statistical analysis to observe correlation of complex scattering phenomena with microstructure properties as the third research specific objective.
1.3) **Introduction of Complex Polycrystalline Structures**

To test the computational framework developed in the thesis, two commonly used materials in turbine-disc engines were evaluated, a nickel based super alloy and a micro-textured titanium alloy. Both were characterized and represented within the NDE sensing models using realistic grain structures, crystallographic orientations, and densities. The nickel...
and titanium materials were chosen because they represent actual materials used in turbine engine applications where microstructure properties are of critical importance [7].

The crystallographic data was gathered using electron backscattering detection (EBSD) scanning electron microscopy (SEM) methods. Detailed 3D elastic properties are needed at each node or element position in the model space, where for example the crystallographic stiffness matrix values and density are needed for each position in a polycrystalline material. In the present material case for nickel polycrystals, the crystallographic symmetry is cubic in nature and the titanium sample has a hexagonal crystal structure. For finite element modeling purposes, the stress-strain tensor relationship requires a 6x6 stiffness matrix with 21 independent values to fully describe the orientation-dependent behavior of each polycrystalline grain.

An accurate representation of the fully 3-dimensional polycrystalline system involves a conversion of Euler angle grain representations into the 21 independent stiffness matrix values for use in the Finite Element Models (FEM) ultrasound models [8]. This conversion is done using specialized Matlab code that takes the Euler angles and rotates a reference crystallographic stiffness matrix based on each grain's crystallographic orientation. Figure 1.3 highlights the reference crystallographic stiffness matrix and density values used for both materials as well as an inverse pole figure (IPF) representation of the material samples. Further explanation of the calculation of the crystallographic orientation will be covered in Chapter 2.
Characterization of each microstructure was done in order to have a ground-truth material property data set to compare with NDE sensing inversion and validation results. The characterization included spatially-resolved estimates of grain size distribution, mean grain size, and grain orientation/misorientation, which can be correlated with the scattering and attenuation of elastic waves as they propagate through a material [9-12]. Chapter 4 explains in more detail the key properties that were characterized based on background research of scattering phenomena and microstructural properties.

Figure 1.3: LSHR nickel (top) and MTR titanium (bottom) inverse pole figures and reference material properties used in the NDE sensing models.

1.4) Introduction of Ultrasound NDE Technique and Key Concepts

The basis of this research is rooted in the nondestructive technique of ultrasound. Ultrasound is characterized by acoustic energy above 20 kHz, with NDE analysis typically occurring in the working frequency range between 20 kHz to 100 Mhz. Ultrasound waves propagate effectively through either liquids or solids, where the wavelength, velocity and propagating modes of the ultrasonic waves depend on the elastic properties of the medium.
Ultrasonic wave generation can vary depending on the type of excitation source used, with common generation techniques involving mechanical, electromagnetic, electro-dynamic, and piezoelectric effects [13]. A typical experimental measurement generates and collects ultrasound using a piezoelectric transducer, which has the ability to convert electrical energy to mechanical energy, and to convert collected mechanical energy back into electrical energy for digital storage using a computer [13, 14]. Propagating elastic waves can involve many modes characterized by their oscillatory patterns as depicted schematically in Figure 1.4 [15]. The most common include longitudinal and traverse modes, with other forms used in NDE involving guided Lamb waves and surface or Rayleigh waves.

**Figure 1.4:** A schematic of the propagation of longitudinal and shear waves.

Similar to other wave phenomena, elastic waves in ultrasound NDE involve the propagation of wave energy over a physical distance ($D$) in the material, occurring over a certain amount of time ($t$). The elastic wave has a velocity ($V$), a wavelength ($\lambda$), and a corresponding frequency ($f$), which all play an important role in NDE measurements and analysis. Below,
Equation 1.1 describes the ultrasonic velocity in terms of time \( t \), propagation distance \( D \), wavelength \( \lambda \), and frequency \( f \):

\[
V = \frac{D}{t} \quad \text{or} \quad V = \lambda * f \tag{1.1}
\]

Because ultrasound NDE involves elastic wave propagation in a material, the elastic wave properties have direct dependencies on the host elastic material properties [16]. This is a critical and necessary connection that makes ultrasound NDE such a practical and important tool for characterizing material property information related to its elastic properties. To be more specific, the elastic wave velocities \( V_l, V_{bl}, \text{and } V_t \) are related to the bulk and shear elastic moduli (E and G) and density \( \rho \) of the material through the equations [15]:

\[
V_l = \sqrt{\frac{E}{\rho}} \tag{1.2}
\]

\[
V_{bl} = \sqrt{\frac{E(1-\sigma)}{\rho(1+\sigma)(1-2\sigma)}} \tag{1.3}
\]

\[
V_t = \sqrt{\frac{G}{\rho}} \tag{1.4}
\]

Where \( V_l, V_{bl} \) are the longitudinal (dilatational) velocity and \( V_t \) is the shear (transverse) velocity as described above and depicted in Figure 1.4. An ultrasonic NDE measurement can, therefore, in principle be used to estimate the elastic modulus material properties if the density is known.

Acoustic impedance is a concept that describes the elastic wave’s behavior in a material. Equation 1.5 describe acoustic impedance \( Z \) in terms of the material density \( \rho \) and the ultrasound wave’s velocity \( V \). In polycrystalline materials, as an elastic wave propagates
through a grain or across the boundary between two separate grains, the acoustic impedance varies at the transition of the boundaries [15]. Acoustic impedance can also be used to describe reflected \((Re)\) and transmitted \((Te)\) energy at boundaries in equations 1.6 and 1.7 respectively.

\[
Z = \rho V \quad \text{(1.5)}
\]

\[
Re = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1}\right)^2 \quad \text{(1.6)}
\]

\[
Te = 1 - Re = \left(\frac{4Z_1Z_2}{(Z_1 + Z_2)^2}\right) \quad \text{(1.7)}
\]

**Figure 1.5:** A depiction of an incident wave reflecting and transmitting at the boundary interface of two media such as grain boundaries.

Another important concept related to wave modes is that of mode conversion. This is the concept of a wave converting to a different mode because of the impedance mismatch between an interface and the angle of incidence. This concept is governed by Snell’s Law (Equation 1.8) [14, 15]:

\[
\frac{\sin(\alpha, \beta)_1}{\sin(\alpha, \beta)_2} = \frac{V_1}{V_2} \quad \text{(1.8)}
\]

Where \(V\) is velocity, and \(\alpha\) and \(\beta\) are angles of incidence, reflection, and refraction for a given pair of materials. NDE ultrasound techniques use this impedance mismatch concept to
detect and characterize cracks, voids, and other material boundaries [13], and in the present case involves reflection, transmission, and scattering of elastic energy by small differences between the polycrystalline grain boundary orientations and elastic properties in a given 3-dimensional direction.

![Figure 1.6: A depiction of an incident wave reflecting and transmitting at the boundary as well as mode converting from longitudinal wave to shear waves.](image)

Attenuation is the result of a loss of elastic wave energy due to absorption or scattering when a beam of energy is passing through a material [15]. In polycrystalline materials scattering most often occurs at the boundaries of grains and is dependent on the relationship of grain size \((d)\) to the wavelength \((\lambda)\) of the propagating wave [13, 17]. This relationship is characterized by three different scattering regimes where the wavelength is larger (Rayleigh), equal to (Stochastic), or smaller than the size of the grains (Geometric). Figure 1.6 depicts elastic wave interactions with each of the three scattering regimes. Within the Rayleigh scattering regime the attenuation is proportional to \(D^3f^4\), and in the stochastic regime the attenuation is proportional to \(Df^2\), where \(D\) is the average grain size and \(f\) is the ultrasonic frequency. In the geometric scattering regime, elastic waves are reflected and transmitted at grain boundaries.
based on the acoustic impedance, reflection, transmission, and mode conversion relationships described in Equations 1.5-1.8.

Figure 1.7: Elastic wave interactions within the Rayleigh, stochastic, and geometric scattering regimes [18].

When elastic waves are scattered at a grain boundary, a portion of the energy is scattered in the back-propagating direction. This is called backscattering and is depicted schematically in Figure 1.8, along with an example of a typical signal that returns to an ultrasonic sensor in the back-propagating direction. In principle, the backscatter signals contain information related to each grain scattering event, where the arrival time of the scattering signal and its amplitude can be related to the distance of the grain from the transducer and the acoustic impedance mismatch (crystallographic orientation mismatch) at each grain boundary using Equations 1.1–1.7.

Figure 1.8: Backscattering of an elastic wave in a polycrystalline structure, and characteristic ultrasound signal highlighting the backscatter data of the signal.
1.5) Introduction of Finite Element Models and Computational NDE

Computational models coupled with NDE physics and digital representations of microstructure are providing a new capacity to understand advanced material systems, microstructure-property relationships, and offering a validation of NDE sensing methods [3 19]. Finite element models such as PZFlex are allowing for complex and increasingly accurate simulations that are advancing computational NDE [19]. In particular, this thesis has focused on extending the capabilities of 2-dimensional and 3-dimensional ultrasound sensing models to include realistic polycrystalline microstructure states.

The PZFlex FEM software used in this thesis employs a mixed explicit-implicit time integration algorithm that results in decreased model size and large improvements in model computation speed for most problems [20] Explicit and implicit methods are approaches used in numerical analysis for obtaining numerical solutions of time-dependent ordinary and partial differential equations. Explicit methods calculate the state of a system at a later time from the state of the system at the current time, and implicit methods find a solution involving both the current state of the system and the later one.

For elastic wave generation, propagation, scattering, and sensing problems, a time-domain finite-element program for solving coupled mechanical-piezoelectric-acoustic problems is needed [20] In practice, the time-dependent calculations are applied to discrete nodes and elements defined in the model space. The accuracy of the models is dependent on controlling numerical errors within the calculations, which largely depend on the spatial and temporal discretization properties of the model grid. A general rule of thumb for elastic wave propagation FEM models is to have at least 20 model grid elements for the shortest elastic wavelength, and a minimal time step that accounts for the time to propagate the elastic wave within a grid
element distance. In the PZFlex FEM model space, the basic grid elements are squares (2D) or cubes (3D). The computational models and software are further explained in Chapter 2 and 5.
2.1) Chapter Introduction and Background

2.1.1) Computational NDE

Computational Non Destructive Evaluation (CNDE) is rooted in the idea of using computational tools to understand, evaluate, and analyze NDE based problems. As described by Blackshire, “The use of nondestructive evaluation (NDE) sensing models may provide an additional capability for understanding and accessing the critical properties of an advanced engineering material system” [3]. Advances in computational tools are allowing for multi-scale and multi-physics problems to be looked at with continually increasing accuracy and complexity [3, 4]. The complex problem of current research efforts outlined in this thesis involve ultrasonic interactions with polycrystalline materials, both 2-dimenional and 3-dimensional sensing problems. As computational methods are becoming a necessity, advances in materials science and engineering have evolved in the areas of materials development, microstructure properties-relationship, and design optimization into a process known as Integrated Computational Materials Engineering (ICME) [4, 6].

2.1.2) Integrated Computational Materials Engineering
ICME is defined as, “the integration of materials information, captured in computational tools, with engineering product performance analysis and manufacturing-process simulation” [6]. ICME provides a framework for combining model-driven methods that take into account material properties, manufacturing, and design in an attempt to create a holistic set of computational tools for materials scientist and engineers [21]. In layman’s terms it is the development of a computational blueprint for the study and analysis of material behavior and properties in a system. The 'integration' concept of ICME refers to the ability to model across different length-scales, different time-scales, and different aspects of a materials engineering problem, where ICME processes act as a bridge between computational materials science, multi-scale modeling, and material informatics [2, 22]. Figure 2.1 shows the four major areas in ICME [22], and how engineering problems, scientific understanding, and data-driven experimental methods can be combined in a computational framework. The 'microstructure-property' area highlighted in Figure 2.1 is the key area relevant to this thesis effort, where NDE sensing physics, models, and methods are being combined with microstructure property information using ICME methods and tools.

ICME is very much in its infancy as far as implementation on a broad scale in the scientific community but offers promising success in its plan [5]. The ICME approach varies from a traditional finite element analysis (FEA) approach because of its ability to describe materials and properties in greater detail than the assigning of nominal property values to materials within the modeling system [21]. For example the ability to incorporate orientation and elastic information of grains in a polycrystalline material would offer a much more effective analysis of looking at wave interactions at grain boundaries than having nominal values in its place.
As described by Ret, “ICME methodologies will enable “transparent” material and process substitutions and improvements without regeneration of exhaustive materials datasets” [23]. This goal described by Ret is much idealized and still being worked towards [5, 6]. There are, however, a few benchmark ICME methodology/model examples described in the literature [5, 6]. Most of these test cases have used various software packages and interfacing scripts to communicate between programs, which can be described as a framework [5]. A generalized concept of this framework is described in the article, "Advancement and implementation of integrated computational materials engineering (ICME) for aerospace application" [5]:

... Once a suite of models have been linked within the integration system, the user can select a subset of these and logically interconnect them, usually graphically into a specific network configuration. The output of one program then becomes the input to the next adjacent program in the network and so on until the final output prediction is achieved...
This linking of models and shared information is considered to be a key feature and current limitation of ICME. The lack of an all-inclusive program suit remains a challenge in a comprehensive ICME system [5]. As the discipline of ICME spreads in the scientific and engineering communities, however, this limitation will be overcome. With respect to the current thesis research, and NDE sensing in complex polycrystalline material systems, ICME methods offer a unique and unprecedented means for studying complex 3-dimensional characterization problems, which is thought to be groundbreaking for computational NDE [6]. The availability of an NDE sensing model framework, as part of a larger ICME toolset, will provide enabling capabilities for multiple length-scale material property evaluation, and comprehensive verification and validation opportunities using integrated and shared material property data sets [3].

2.2) Framework Objective

The first objective of the research was to develop a Computational Nondestructive evaluation (CNDE) framework that uses the methods and tools of Integrated Computational Materials Engineering (ICME) to understand and study ultrasonic NDE interactions with two and three dimensional microstructures to better quantify material state properties. The framework should provide a means for taking realistic microstructure information from electron backscatter diffraction (EBSD) scans or synthetically generated microstructure volumes, and incorporate the realistic material properties of the specimen into an NDE sensing model instead of traditional approaches, which utilize approximate or nominal material property values.

2.3) Computational Tools

The framework developed in this thesis needed to integrate several software packages capable of extracting/generating the material properties of the microstructure, characterizing
the material properties, calculating crystallographic information of grains in the polycrystalline structure, and incorporating the crystallographic information into the Finite Element models to study NDE sensing interactions. The primary software packages included: EDAX TSL OIM, DREAM.3D, Matlab, and PZFlex as summarized in Table 2.1. Secondary software packages included Paraview and OriginPro.

For the importation of EBSD scan data and the extraction of material properties, the software packages EDAX TSL OIM and DREAM.3D were used. The EDAX TSL OIM and DREAM.3D packages were also used to characterize the microstructure properties such as Euler angles, pole figures, grains size, crystallographic orientations, and grain mis-orientations. Mathworks Matlab was used to create custom code to calculate crystallographic orientation properties for each grain based on the Euler angle values and a reference crystallographic matrix and density for each material system. Matlab code also acted as a bridge between the material properties and model formats with code to reconfigure and structure the material property data. And finally, PZFlex Finite Element Modeling (FEM) software was used to simulate and study the elastic wave propagation, scattering, and NDE sensing relationships for each material system. The next sections describe each computational tool and their importance in the overall framework in more detail.

| TSL Orientation Imaging Microscopy (OIM) | Importation of EBSD data | 2D Microstructure Analysis | 3D Microstructure Analysis | Characterization Tools |
| Digital Representation Environment for Analysis of Microstructure in 3D (DREAM.3D) | Importation of EBSD data | 2D Microstructure Analysis | Synthetic Microstructure Generation | Characterization Tools |
| MATLAB | Calculate Crystallographic Properties | Reconfigure and Structure Properties | | |
| PZflex | NDE FEM Sensing Models | | | |

19
Table 2.1: A table listing the computational tools used in the ICME framework and their capabilities used within the scope of the research.

2.3.1) EDAX TSL OIM

The EDAX TSL OIM software package was used in the framework for several purposes. The primary use was to provide a means for incorporating realistic polycrystalline microstructure information from EBSD data sets into the PZFlex ultrasound sensing models. The EDAX TSL OIM software takes Electron Backscatter diffraction (EBSD) image data sets captured with the Scanning Electron Microscopy (SEM) system, and provides a variety of computational tools to reconstruct, analyze, clean, and manipulate the crystallographic data for material property characterization [24, 25]. The EBSD technique is rooted in SEM electron energy beams interacting with polycrystalline materials and diffracting at crystallographic planes [24]. The diffraction patterns can be used to determine orientation and phase at spatially resolved points on the material surface [24]. Grain boundary and orientation maps are created by comparing orientation values of neighboring elements in the specimen of interest [24].

Crystallographic orientation is of interest because it is able to determine the elastic stiffness of each grain based on position and rotation to a reference. The orientation is described in terms of Bunge Euler Angles [26]. There are three Euler angles (\( \varphi_1, \phi, \varphi_2 \)) used to describe the 3 rotations of a samples reference compared to a crystal reference frame for a material (Figure 2.2) [26]. This crystallographic orientation is used to create pole figures (PF) to characterize orientation distribution, inverse pole figures (IPF) to describe the position of the sample direction relative to the crystal reference frame, and the output of the Euler angles essential to calculate the rotated elastic stiffness matrix of polycrystalline grains as described by the Matlab coding created in the scope of the research (Figure 2.3) [26 27]. The orientation
data from the EBSD measurements are stored in *.ang files for the calculation of properties through EDAX TSL or the other computational package DREAM.3D.

**Figure 2.2:** A viewgraph of crystallographic orientation as it pertains to element orientations described by Euler angles \((\varphi_1, \phi, \varphi_2)\) (left). A polycrystalline material representing the crystallographic orientations of the grains (right) [26 28].

**Figure 2.3:** Viewgraphs depicting a typical set-up of EBSD measurements and measured diffraction patterns. Diffraction patterns help to determine orientation and boundaries as viewed in micrographs and inverse pole figures (far right) [24].

Within the scope of this research, the EDAX TSL OIM software was also used to clean the EBSD material data sets (remove measurement noise), to crop the overall data set into smaller regions, and to analyze the spatially-resolved microstructure properties related to grain orientations and distribution statistics, grain sizes and distribution statistics, pole figures, and mis-orientation of grains. Distribution and misorientation statistics were exported in column delimited ascii text files for further quantification using Excel and OriginPro to create Log normal
distribution plots. Pole figures (PF) and inverse pole figures (IPF) were output as jpeg images with orientation data output as a txt file for further quantification of dominant crystallographic orientations within the sample using Matlab code. Figure 2.4 shows representative EBSD maps and analysis for the two primary materials studied in the thesis: an LSHR nickel super alloy (left), and a textured titanium (right) sample using the TSL OIM software.

![Figure 2.4](image)

**Figure 2.4:** Representative EBSD maps and analysis using EDAX TSL OIM software to map out the microstructure and extrapolate data such as misorientation distributions and pole figures.

2.3.2) **DREAM.3D**

DREAM.3D is an acronym for "Digital Representation Environment for the Analysis of Microstructure in 3D". As described by the software’s primary developer, Dr. Michael Groeber, “the software allows for processing, segmenting, quantifying, representing, and manipulating digital microstructure data” [29]. DREAM.3D acts as a key linking tool for ICME methods by
bridging gaps between processing, microstructure evolution, and property models in an attempt to optimize both microstructure properties and performance of a material incorporated into a digital representation of a structure [29]. DREAM.3D provides a unique capability for interacting with microstructures on three levels, cell data or points, field data or collections of points, and ensemble data or collection of fields (Figure 2.5). This multi-level analysis provides a comprehensive amount of property information about the material system for ICME.

**Figure 2.5:** DREAM.3D data structures including: cell data, field data, and ensemble data [30].

DREAM.3D was chosen as a key computational tool for use in the current framework because of its ability to analyze and characterize 2D and 3D microstructures. Aspects of the microstructure that needed to be characterized included grain size information, grain orientations and neighbor misorientations, spatial coordinates, and Euler angles as primary information that needed to be incorporated into the NDE sensing models. To aid in the analysis of data structures, DREAM.3D uses the concept of ‘pipelines’, which contain a set of ‘filters’ tailored to the analysis of a specific goal. Each filter is its own individual program or algorithm that processes the raw data of the sample and stores the information [29]. Filters combined in a series make up a pipeline. Pipelines can be as simple as converting one file type to another, or
progress in complexity to a full cleanup, analysis, and reconstruction of a 3-dimensional microstructure sample. Figure 2.6 shows the graphical user interface (GUI) of DREAM.3D highlighting the filter library and pipeline compilation section of the interface.

Figure 2.6: The graphical user interface of Dream.3d software. A green outline highlights the filter library. A blue outline highlights the list of all available filters. A red outline highlights the pipeline of filters that are in current use, and orange highlights any errors and progress when the software is running.

Within the scope of the research there were two approaches taken within the framework to implement polycrystalline structures into the NDE sensing models. These approaches were differentiated by the type of polycrystalline structure used - realistic and synthetic. The first approach involved taking real crystallographic information from SEM EBSD scans and recreating the microstructure in a virtual environment. This approach is unique in the
sense that a virtual representation of an actual physical microstructure can be generated and used in analysis and modeling environments, where point-by-point representations of the crystallographic material properties define grains and grain boundary features. Having a virtual representation of a microstructure gives the ability to run virtual tests and simulations, to virtually segment and manipulate the data sets, and to study and quantify local microstructure properties, which is not feasible with the actual physical microstructure.

For realistic EBSD data sets, the framework used two DREAM.3D pipelines. The first pipeline contains one filter that imports orientations files output from the EDAX TSL OIM software, and consolidates and converts them into an H5EBSD file. This is needed because the H5EBSD file stores data in a hierarchical way that allows for organization and flow within the DREAM.3D computational tool [29]. The second pipelines imports the new HFEBSD file, and applies multiple filters to estimate grain positions and sizes, grain orientations, mis-orientations of neighboring grains, and outputs this data into a comma separated variable (*.csv) output file, along with a file with all grain positions and grain ids (*.ph) , and a visual representation file compatible with the Paraview software (*.xdmf). Figure 2.7 provides examples of the two DREAM.3D pipelines and typical values used in the various filters.
Figure 2.7: A set of DREAM.3D pipelines used to take EBSD orientation file and do analysis within DREAM.3D. Pipeline 1 (left) imports the orientation files from the TSL EDAX OIM software into the DREAM.3D environment. Pipeline 2 (right) gathers data on the spatial coordinates, grain sizes, mis-orientations, and Euler angles, and output this information to be used as an input to third portion of the framework involving Matlab software code.

The second approach used in the present research involved a very powerful capability of the DREAM.3D software environment, which can generate custom synthetic 3-dimentional microstructures for use in simulation tools [29]. The framework used DREAM.3D to generate synthetic microstructures, and then simulate and study NDE sensing in virtual grain structures.
that could be customized for a given application. The synthetic microstructures are created in a second DREAM.3D executable program called StatsGenerator.exe that allows for the generation of an xyz microstructure that is defined by the user based on statistical distributions of desired grain sizes, grain shapes, texture, misorientaion, number of phases, xyz spatial resolution sizes, and reference material properties. The benefit of this approach is the rapid generation of a controlled and idealized synthetic microstructure, where the statistical properties of the microstructure are well defined and known. Figure 2.8 shows the DREAM.3D GUI for the stats generator program, and representative 3-dimensional microstructure examples for equaixed and elongated microstructures generated using this computational tool. The pipelines for this second approach only vary from the first set of pipelines above, where instead of importing EBSD orientation data, the user imports a 2-dimensional or 3-dimensional synthetic area/volume that is based on the statistics and microstructure generated by the stat generator program. The steps to the pipelines will be viewed in greater detail in the framework section 2.8 of the thesis.
Figure 2.8: a. The graphical user interface (GUI) of DREAM.3D's software StatsGenerator.exe. The interface shows a statistical grain size distribution with controls to vary the distribution width, height, size and cut off values. b. Representative synthetically generated microstructures are depicted for 3D equiaxed grain microstructure (bottom left) and elongated grain microstructure (bottom right).
2.3.3) MATLAB

Matlab coding was created within the scope of the framework to take the Bunge Euler angles \((\varphi_1, \phi, \varphi_2)\) outputted from the DREAM.3D *.csv and *.pf file and calculate 21 anisotropic values [31] of the rotated elastic stiffness matrix for all identified grains in a polycrystalline sample with the FEM PZFlex software. Dr. Adam Pilchak, Air Force Research Labs, made extensive contributions in developing code that take Euler angles for a 2-dimensional sample and using the Bunge notation to take the calculated the rotation matrices. Equation 2.1 shows the rotation matrix where the Euler angles were applied to [31, 32, 64].

\[
\begin{align*}
    c_1 &= \cos(\varphi_1) , s_1 = \sin(\varphi_1) \\
    c_2 &= \cos(\varphi_2) , s_2 = \sin(\varphi_2) \\
    C &= \cos(\phi) , S = \sin(\phi) \\
    R &= \begin{bmatrix} 
        c_1 c_2 - s_1 s_2 C & s_1 c_2 + c_1 s_2 C & s_2 S \\
        -c_1 c_2 - s_1 s_2 C & -s_1 c_2 - c_1 s_2 C & c_2 S \\
        s_1 S & -c_1 S & C 
    \end{bmatrix}
\end{align*}
\]

The code (DREAM3D2PZFLEX M-File) requires the input of 6x6 material reference matrix, Bunge Euler angles, associated grain id’s and spatial coordinates. The output of the code is a 6x6 rotated matrix based on the grains unique set of Euler angles. Because of symmetry and PZFlex scripting requirements only 21 values were needed as shown in figure 2.9.

![Figure 2.9: A viewgraph of a 6x6 crystallographic stiffness matrix highlighting the 21 value positions used to describe materials in FEM models.](image-url)
The code was initially set up for 2-dimensional samples, with further manipulation the code was reconfigured for calculation of 2-d serial sections slices to compile a 3-d volume with a looping process. Figure 2.10 show a sample output where spatial coordinates, grain id and 21 stiffness matrix values are recorded in a txt file. Further review of the codes can be viewed in appendix C and the framework sample at the end of chapter 2.

**Figure 2.10:** A viewgraph of sample matlab text output describing xyz position of each element, grain id and 21 value stiffness matrix. The text File is used as an aide to describe polycrystalline materials.

The output from the DREAM3D2PZFEX code described above is used as an input to another set of codes created. The way the FEM models are described, it calls for two input files. The first file describes the spatial coordinates of the sample in 2 dimensions or 3 dimensions.

The “regn” command in PZflex FEM specifies properties for an element/voxel of the virtual representation of the polycrystalline structure [21] (figure 2.11). For a material grain the beginning and ending elements for an IJK space have to be defined. The code “Cij Spatial Coordinates 2D M-file” writes a text file describing each element of a microstructure spatial coordinate for a 2-dimensional structure. The codes “cij Spatial Coordinates 3D Brut Force non-clustering M-file” and “consolidate_CijSpatialCoordinates_files M-file” complete the same task for a 3-dimensional volume. The sets of codes are described in step by step procedure in appendix A and actual codes can be found in appendix C. Figure 2.11 shows a sample output text file from the computation of the code.
Figure 2.11: A viewgraph of “REGN” PZflex command and variables required for command to function successfully.

Figure 2.12: A sample output text file from the matlab code “Cij Spatial Coordinates 2D M-file” used as an input file to PZflex FEM model to describe the spatial coordinates of every node/element in NDE Sensing Models for a 2-dimensional microstructure.

The second input needed for the FEM models is a text file that describes crystallographic properties as rotated stiffness matrix for the grains of the polycrystalline microstructure. From the DREAM3d2pzflex output 21 stiffness matrix values were calculated based off grain Euler angles. The codes “Cij Stiffness Matrix IDs M-file” and “Cij Stiffness Matrix IDs 3D M-file” define the crystallographic elastic stiffness matrix information based on the PZFLEX fem command “Prop”. The prop command defines the 21 stiffness matrix values, density of material for all
grains in a respective material (figure 2.13). Figure 2.14 shows the sample text output from the code. The codes can be reference in appendices A and C for more information.

**Figure 2.13:** A viewgraph of “Prop” PZflex command and variables required for command to function successfully.

**Figure 2.14:** A sample output text file from the matlab code “Cij Stiffness Matrix IDs M-file”/“Cij Stiffness Matrix IDs 3D M-file” used as an input file to PZflex FEM model to describe the crystallographic properties every grain in NDE Sensing Models.
2.3.4) \textit{PZFLEX Finite Element Software}

A number of NDE sensing models are available for research. In the present thesis, an ultrasonic sensing model called \textit{PZFlex} was used [20]. The \textit{PZFlex} software is a combined implicit-explicit Finite Element Modeling (FEM) software package developed by Weidlinger Associates, which provides a means for modeling elastic wave propagation and scattering in polycrystalline materials, where the nodes and elements of the FEM model are assigned density and 3-dimensional stiffness matrix values at each model grid location. The ICME framework developed in this thesis takes advantage of \textit{PZFlex}'s ability to model each element of a polycrystalline structure, where input of realistic EBSD microstructure data sets or synthetic DREAM.3D data sets are incorporated into the \textit{PZFlex} model for sensing studies. The output data sets from the Matlab codes are used as an input into \textit{PZFlex} to build the microstructures with each spatially resolved element accounted for and assigning each element its crystallographic material properties.

All NDE sensing models follow the same basic steps to building an FEM model outlined in Figure 2.15 [20]. The ICME framework has an emphasis on step four of constructing the model, assigning material properties to areas/volumes. The framework attempts to realistically input a polycrystalline structure into the virtual sensing environment accounting for all material properties correctly (Figure 2.16). Chapter 5 will talk about particular ultrasonic model details for the thesis research in depth.
Figure 2.15: Progression of steps in order to create PZflex ultrasound FEM sensing model [20].

Figure 2.16: Progression of the ICME framework taking a polycrystalline microstructure, generating an FEM mapping, assigning material properties, and finally conducting ultrasound wave propagation, scattering and sensing model [2].

2.4) ICME Framework

The framework begins with either realistic microstructure obtained through EBSD scans, or synthetic microstructures generated using DREAM.3D. Either will be used as input to a set of DREAM.3D pipelines to begin transformation of the polycrystalline data information into format compatible with the FEM model input requirements. A key material property of interest is the Euler angles for each xyz spatial position, which is output from the EDAX TSL OIM of DREAM.3D software and is converted into Cij stiffness matrix values at each xyz position using the
framework developed in this thesis. The output files from DREAM.3D contain Euler angles, spatial coordinates, and also analyzed information related to grain ID's, grain sizes, misorientations, and other properties to correlate to NDE ultrasound sensing response later. The first set of Matlab codes takes the Euler angle and reference stiffness matrix of the material and calculates the rotated elastic stiffness matrix of each grain based on its orientation to the reference matrix. The second set of Matlab codes formats the spatial coordinates, grain identification numbers, and elastic stiffness matrix values into two text files that are able to be imported directly into the PZflex FEM modeling realm. Figure 2.17 shows the progression of the framework and computational tools used at each phase.
Figure 2.17: The ICME framework providing a means for incorporating realistic and synthetic polycrystalline microstructures into NDE ultrasonic FEM sensing models.
2.5) ICME Framework Example

Below is an abbreviated step-by-step example with screenshots of using the ICME framework to take a realistic EBSD microstructure data set and transform it for use in the PZflex NDE sensing model format. Attached in Appendix A is a detailed step-by-step procedure to navigate the ICME framework for real and synthetically created microstructures.

***Start of ICME FRAMEWORK EXAMPLE***

**Step 1 : EDAX TSL OIM Software - realistic microstructure**

- The EDAX TSL OIM software is used to map out and characterize 2D samples microstructure, once done the EBSD *.ANG files are used as a starting point for the DREAM.3D step.

**Step 2 : DREAM.3D**

**TSL_2_DREAM3D_1 pipeline 1**

*This pipeline takes the *.ang file(s) and converts them to an H5Ebsd file.*

- Set input directory
  - Folder where *.ang file are located
- Set output directory for h5ebsd file
  - If there is more than one ang file make sure to set prefix and suffix, number of files, and padding digits
  - In the case of more than 1 ang file, you need to have some types of sequential number mechanism in order to be recognized by DREAM.3D
- Run pipeline
- Below is the setup of DREAM.3D pipeline number one
TSL_2_DREAM3D_2 pipeline 2

Pipeline 2 takes the H5Ebsd file from pipeline 1 and goes through a process to calculate field data in order to find Euler angles of the grains, and to output data as a DREAM.3D formatted file, and finally converts the DREAM.3D file into PH and CSV files to be used as an input to the Matlab software code

- Read in H5EBSD file
  - Input the name of particular file and directory
  - Check cell data, field data and ensemble data boxes in this filter
- Generate name for output DREAM.3D file and directory
- Designate an output name and directory for the *.ph file
- Designate an output name and directory for the *.csv file
- Run pipeline
- Below is a typical setup of pipeline 2
Step 3: Matlab A: Calculation of Crystallographic orientation

The first set of Matlab codes opens the PH and CSV files output from the DREAM.3D pipeline, and takes the Euler angles of each grain id and calculates the crystallographic
stiffness matrix based on grain orientations relative to a reference material stiffness matrix

DREAM3D2PZLEX M-FILE

- Make sure that all m-files are in the same directory of code will not work properly
  - Main M-File – DREAM3D2PZLEX
  - Attached M-Files – RMatOFBunge, RotateCijklReal, RotateY, Cij2Cijkl, Cijkl2Cij, CijNickel.mat, CijklTi64.mat
- In Main M-File DREAM3D2PZLEX:
  - Text read in file directory of PH and CSV files of the data sample

```matlab
% specify the file outname and directory
phfile = textread('C:\Users\darius\Desktop\Dream3D to PZFiles\step 2ph_output.mat',...,'-i', 'HeaderLines', 1);
dream3d2puzzle = textread('C:\Users\darius\Desktop\Dream3D to PZFiles\step 2puzzle_output.mat',...,'-i', 'HeaderLines', 2);
outputname = 'C:\Users\darius\Desktop\Dream3D to PZFiles\step 3layer_7.out';
```

- Continuing in DREAM3D2PZLEX
  - Specify Number of voxels in X and Y direction and resolution

```matlab
% number of voxels in x and y and the resolution.
nx = 199;
ny = 199;
res = 1; % resolution in microns
```

- Load the reference material 6x6 material. (usually nickel or titanium)

```matlab
load CijNickel;
Cij = CijNickel;
```

- In the code where it says `eul = grainfile(i,7:9)' 7:9 are the columns with in the CSV file which contain the Euler angles, make sure to specify the right columns for particular data set.

```matlab
for i = 1:size(grainfile,1)
    gid = grainfile(i,1);
    eul = RangeOFMarlRMarOFQuant(grainfile(i,4:7)', 'radians');
    eul = grainfile(i,7:9);  % find these locations in the phfile
```
- Run the Matlab code
  - If ran correctly the code will generate a 2-dimensional grain map of the grain structure as such:

  ![Grain Map](image)

  - Code will also output a file (*.out) in format below:
    - % x-position y-position Stiffness Number c11, c12, c13, c14, c15, c16, c22, c23, c24, c25, c26, c33, c34, c35, c36, c44, c45, c46, c55, c56, c66

**Step 4 : Matlab B: RECONFIGURING DATA**

*Used to reconfigure or reformat two data text files which are needed to be used in the PZflex FEM model format. The first file describes the spatial coordinates of each pixel or voxel in the microstructure area or volume space and assigns a 'grain ID' to each xyz position. The second file accounts for the Cij stiffness matrix unique to each grain ID of the sample. There is a set of codes to reconfigure the data for a 2D data set and a set of codes for 3D data sets. Also there is specialized code to initialize a 3d volume of a 2d*
layer as well as clustering of similar voxels together for the purpose of memory of the computer.

**Cij_Spatial_Coordinates_2D**
- Takes output file from the DREAM3D2PZFLEX Matlab code and reconfigures the data in the form recognized by PZFLEX that accounts for material identification designation for all spatially resolve elements in the material model. The spatially resolved elements are identified by position in the xyz coordinate system, and explicitly written as x begin position, x end position, y begin position, and y end position. For 3-dimensional data sets an additional z begin position and z end position is included.
- Input file and directory is the output file from the Matlab A step's code

```matlab
% Define input and output file
% *** changes for every new data set ***
inputfile = 'C:\Users\darius\Desktop\Dream3D to PZFLEX\step 3\layer 1.txt';
outfile = textread(inputfile);
```

- Specify file name and directory for the output file

```matlab
fileWriteID = fopen('C:\Users\darius\Desktop\Dream3D to PZFLEX\step 4\pzflea_input_regn_i_j.txt','w');
```

- Run the Matlab code
  - Typical code output:
    - `regn mat169 1. 2. 1. 2.
      regn mat169 2. 3. 1. 2.
      regn mat169 3. 4. 1. 2.
      regn mat169 4. 5. 1. 2.
      regn mat169 5. 6. 1. 2.
      regn mat169 6. 7. 1. 2.
      regn mat169 7. 8. 1. 2.
      regn mat169 8. 9. 1. 2.
      regn mat169 9. 10. 1. 2.
      regn mat169 10. 11. 1. 2.
      regn mat169 11. 12. 1. 2.
      regn mat169 12. 13. 1. 2.
      regn mat169 14. 15. 1. 2.
      regn mat169 15. 16. 1. 2.
      regn mat169 16. 17. 1. 2.
      regn mat169 17. 18. 1. 2.
      regn mat169 18. 19. 1. 2.`

**Cij_Stiffness_matrix_IDS**
- Takes Dream3d2pzflex code output file and reconfigures the data so that each grain identifier has attached its rotated 21 elastic stiffness matrix values that are dependent on grain orientation. Also the computation of this code writes out the density of the material
- Input file and directory is the output file from the Matlab A step's code

```matlab
% Define input and output file
% *** changes for every new data set ***
inputfile = 'C:\Users\darius\Desktop\Dream3D to PZFLEX\step 3\layer 1.txt';
outfile = textread(inputfile);
```

- Run the Matlab code
- Specify file name and directory for the output file

```matlab
filenames = fopen('C:\Users\dariusj\Desktop\Dream3D to PZflex\step 4\prop_function.txt', 'u');
```

- Specify density of material

```matlab
raw_grain_number = outfile(:,3);
stiffness_matrix_values = outfile(:,4:24);
density = 7500;  % *** changes for a given material ***
```

- Run the Matlab code

```
step 5: PZflex Ultrasound FEM model

PZflex is a scripting language FEM with function calls, variables and boundary conditions amongst other features. The PZflex program allows 'text' file information to be read directly into the PZflex model script code, and therefore a goal of the framework was to provide a set of 'text' files that described the xyz grid element assignments of grain/microstructure information and the elastic property information for each of the materials in the model (in the present case the material properties are the 21-element stiffness matrix that describe the elastic stiffness properties in each grain and the density. The Matlab code and 'framework' provide these two text files that can be read into the PZflex model. Much more detail will be provided in chapter 5.
```
The two final files output by the Matlab step B codes are used as input files to the PZflex FEM modeling software in order to describe and create the complex polycrystalline microstructure in the FEM model space.
3.1) Chapter Introduction and Background

3.1.1) Introduction

Using the framework described in Chapter 2, four different polycrystalline samples were characterized, analyzed and imported into computational NDE models using the ICME framework created during the scope of the research. Two samples are of actual materials used in aircraft turbine engines and are the focal point of analysis. One of the samples is a nickel based super alloy while the second sample is a textured titanium alloy. Both of the data sets for these particular samples include rather large 2-dimensional EBSD maps in the centimeter size range and offer complex and vastly different polycrystalline structures. The second set of samples were analyzed and modeled to a lesser degree but show the capability and feasibility of the methodology to be used for NDE sensing analysis of 3-dimensional structures as well. Of the 3-dimensional samples, one was synthetically generating, while the other sample comes from serial sectioning and EBSD scans of a real nickel super alloy microstructure to create the 3D volume. For the two 3-dimensional sample data sets, the overall sizes were smaller, and were on the order of a few hundred microns to millimeters in size.

3.1.2) Background on Specific Microstructures
Due to the extreme conditions imposed on components during the operation of a turbine jet engine, aerospace material are regularly subjected to high temperature conditions and degradation over time[7]. Nickel based super alloys and titanium alloys are often used in components of the turbine engine such as turbine disk because of advanced material properties that are traced back to the microstructure [7]. Nickel based super alloys are a unique set of materials. Nickel based super alloys have superior properties related to high temperature strength, creep resistance, and resistance to degradation such as corrosion and oxidation [33].

The sample used for this research was a Low-Solvus High-Refractory (LSHR) nickel super alloy meaning low-solvus aids in the resistance of quench and dwell crack and the high refractory lends its self to high tensile strength and creep resistance [34]. The specific LSHR nickel microstructure in this research had a 'dual' microstructure gradient of fine grains on one side of the material, and a coarse grain on the other side. This dual microstructure is characteristic of a dual microstructure heat treatment processing (DMHT) [35]. DMHT is application dependent for super alloys in turbine disks [35]. Creep and dwell crack resistance is needed in the rim of the disk, whereas at the bore of the disk high strength and fatigue resistance is required [35]. This is achieved with a dual microstructure, fine equiaxed grains at the bore and coarse grains at the rim of the disk [35]. Any other microstructure would compromise microstructure dependent properties for the turbine disk leading to the component failing in intended usage [35]. In the present NDE sensing application the detection and characterization of the grain size distributions in the fine, coarse, and mixing region was desirable.

Titanium alloys are another aerospace grade material that posse’s suitability in high temperature applications [7]. The sample used in this research was near alpha titanium alloy,
and these types of alloys are desired for tensile strength, creep, fracture toughness, crack propagation resistance, and other key properties needed to operate in a turbine engine for its intended purpose [36]. The titanium sample used in this research had local regions in the sample where polycrystalline grains were similarly oriented into 'macro-zones' or Micro-Textured Regions (MTRs) as described by Pilchak [37]. In the present NDE sensing application the detection and characterization of these MTR regions was desirable.

The nickel LSHR super alloy and the titanium Ti-6Al-4V alloy with MTR regions have vastly different microstructures, where NDE sensing model interactions should be different making these two microstructures ideal for wave interaction comparisons using the developed methodology. The 3 Dimensional samples will be discussed later on in chapter 3.

3.2) 2-Dimensional Realistic Microstructures

3.2.1) Low-Solvus High-Refractory (LSHR) Nickel Super Alloy

The first sample used in this research was an LSHR nickel super alloy sample. The LSHR nickel super alloy has a face center cubic crystal structure [38]. The microstructure has equiaxed grains meaning there is random symmetry in the microstructure and the axes are of similar length in the microstructure [70]. This microstructure was examined and characterized in detail, the results of which will be described and presented in Chapter 4.

Figure 3.1a shows a schematic diagram of a turbine disk from which the nickel LSHR sample had previously been extracted along the radial direction [39]. As depicted in the figure, the sample included a coarse grain region towards the outer rim (blue outline), a fine grain region towards the inner bore (red outline), and a transition zone between the two regions (green outline) within the sample. Figure 3.1b also shows an EBSD scan Inverse Pole Figure (IPF) mapping of the polycrystalline structure, which describes the grain orientation directions
relative to crystal reference frame [27]. The EBSD data was measured using and SEM system in AFRL/RXC facilities by Dr. Adam Pilchak. The IPF map visually highlights the fine to coarse gradient of grains within the structure. The dimensions of the EBSD map were 19000 microns (1.9 cm) in the X direction and 8010 microns (0.801 cm) in the y direction, with a total area of 1.5219 cm$^2$. The step size of the EBSD scan was 5 microns, yielding more than 6 million total measurement points. The EBSD data set contained approximately 286,700 grains in the polycrystalline structure.
Figure 3.1: Schematic diagram of a turbine disk and cut away section identifying microstructure features of the LSHR nickel sample (top figure) [39], and the Inverse Pole Figure (IPF) mapping of the LSHR Nickel super alloy grains and grain boundaries (bottom image).

3.2.2) MTR Titanium 6Al-4V Alloy

The micro-textured region (MTR) titanium sample was also characterized in detail and will be covered in Chapter 4. Similar to the LSHR nickel sample, the titanium sample was extracted from an actual turbine disk component. The titanium alloy has a hexagonal closed packed crystal structure, with the microstructure categorized as a near alpha phase titanium, with trace amounts of a beta phase [40]. The extracted sample had an EBSD scan accomplished with an overall size of 25150 micron (2.515 cm) in the X direction and 8870 microns (0.887 cm) in the Y direction as displayed in the IPF result in Figure 3.2. The EBSD measurement was again experimentally obtained by Dr. Adam Pilchak of Air Force Research Laboratory (AFRL/RXCM) using an SEM system in the AFRL facilities. The step size of this scan was 10 microns, yielding 2.23 million total measurement points. The IPF map for the titanium MTR sample depicts very large grain regions with similar crystallographic orientations, which can occur in this type of material. The nominal grain size in this material was on the order of the scan resolution, which resulted in over 2 million grains being identified during standard automated grain size and grain identification analyses.
Figure 3.2: The Inverse Pole Figure (IPF) mapping of the Micro Textured Region (MTR) titanium alloy sample showing grains and grain boundaries.

3.3) 3-Dimensional Microstructures

3.3.1) 3D Synthetic Equiaxed Grain Structure

A 3-dimensional synthetic microstructure was generated using the capabilities available in the DREAM.3D software package. As describe previously, the StatsGenerator.exe program in Dream.3D can be used to create a polycrystalline microstructure with the desired grain property statistics in a 3-dimensional volume. Figure 3.3 depicts the graphical user interface (GUI) for the StatsGenerator.exe program, and the input parameters that were used to generate the synthetic microstructure for this thesis research, which included varying two variables, mu and sigma, which are related to the average value and standard deviation in the log normal size distribution. Manipulation of the sigma cut off values allow the user to disregard outlier larger and small grains from the statistical data set [30].
The goal was to create a microstructure with equiaxed grains that had an average size of approximately 33 microns and a narrow grain size distribution between 20 and 54 microns. The statistical data set was used in DREAM.3D to initialize a synthetic volume of 200um X 200um X 200um with a 1 micron resolution using the pipeline process for generation of a synthetic structure. The 3D volume contained 8,000,000 voxels and 399 identified grain. Figure 3.4 shows the 3-dimensional equiaxed grain structure that was created.

The synthetic microstructure was characterized by grains size and misorientation which will be highlighted in the next chapter. Also orientation values were assigned to each grain with a set of Euler angles output by DREAM.3D in a *csv file. Further processing of the synthetic volume include assigning calculated elastic stiffness matrix values to each grain with the framework’s Matlab code using the input of each grain’s Euler angles.

The process for assigning crystallographic orientation data is different for 3-dimensional volumes because the current framework developed as part of this thesis works with 2-dimensional data sets. Each layer of the 3-dimensional volume, therefore, needed to processed separately and then re-combined to form the 3-dimensional volume and microstructure for use in the PZFlex FEM models. Additional details on this process and step by step procedures are provided in Appendix A and Appendix B. The process of generating a 3-dimensional microstructure currently requires an extensive amount of time, but feasibility of using the framework for a synthetic 200x200x200 um equiaxed nickel microstructure was shown.
Figure 3.3: Graphical user interface (GUI) of the StatsGenerator.exe program used to generate a synthetic equiaxed 3D microstructure.

Figure 3.4: StatsGenerator.exe resulting 3D microstructure with visualization of the X+, Y+, Z+ faces.
3.3.2) 3D Inconel 100 Nickel Sample

The second 3-dimensional microstructure involved a realistic data set of an Inconel 100 nickel sample that was generated using serial-sectioning methods by Dr. Mike Uchic of Air Force Research Laboratory [30]. Inconel 100 is nickel-chromium based super alloy used in aerospace applications [41]. This particular sample data set was supplied as part of the DREAM.3D software package in one of its tutorials [30]. The data set included EBSD measured layers that were acquired sequentially using precision serial sectioning automated polishing methods. The sample set was small in overall size, but provided a useful data set to be used for proof of concept and feasibility studies for the scope of research project. The sample had dimensions of 121 um X 121 um X 117 um with a resolution of 0.25 microns. The 3D microstructure consisted of 2287 identified grains and is depicted in Figure 3.5.

![Figure 3.5: Viewgraph depicting the IN 100 structure with visualizations of the X+, Y+, Z+ faces.](image-url)
CHAPTER 4

MICROSTRUCTURE QUANTIFICATION

4.1) Chapter Introduction and Background

In this thesis, the characterization of the nickel based super alloy and the MTR titanium 6Al-4V alloy involved analyzing the features of the microstructures that might impede elastic wave propagation through the sample. Since the two microstructures possess different qualities, the characterization process varied based on dominant features of each microstructure that could be extrapolated for quantitative comparison with ultrasound signals as described further in the chapter. In general, the crystallographic orientations, mean grain sizes, grain size distributions, and MTR geometric morphology features are expected to cause variations in the ultrasonic velocity, scattering, reflection, and attenuation properties.

Thompson has noted that individual grains in a polycrystalline material can be defined by their shape, size, and crystallographic orientation [12]. When an elastic wave propagates through a polycrystalline microstructure, it scatters at the boundaries of grains because of elastic/grain anisotropy, misorientation, and morphology [10, 42, 43]. As discussed previously, when the grains are small compared to the ultrasonic wavelength, the interaction primarily involves Rayleigh scattering [44]. In this instance, each grain scatters a small amount of the ultrasound energy. Stanke and Kino described this analytically [44], where the ultrasonic
attenuation is predicted to have a power law dependence on the mean grain size and ultrasound frequency. This type of microstructure characterization involves estimating the mean grain size and grain size distribution in a polycrystalline material, and comparing that with the attenuation in an ultrasonic signal as the frequency is changed as described in Chapter 1.

Researchers have also used the backscatter of ultrasonic signals as a characterization tool for polycrystalline materials [12, 13]. In this case, each grain scatters a small portion of the energy at each grain boundary directly back to the ultrasonic sensing surface. This type of interaction is again dependent on the mean grain size and the ultrasonic frequency. In contrast to the attenuation of an ultrasonic signal due to grain scattering, for backscatter there would be an increase in signal for increased scattering processes.

The nickel based super alloy studied in this thesis is classified as an equiaxed polycrystalline microstructure, which means that the grains have axes of approximately the same length [45]. Because the LSHR material has a 'dual' microstructure, there is a change in the sizes of the grains from one region to another by approximately an order of magnitude, and there also exists a mixing region where grains are transitioning for coarse (large) to fine (small) grain sizes and both sizes exist locally within the material volume. A characterization of the local mean grain sizes and grain size distributions was chosen for this material type, where comparisons with ultrasonic attenuation and backscatter signals was accomplished.

Nickel alloys also involve a material property termed 'twin boundaries'. Twin boundaries involve a planar defect found in nickel alloys grains that needs to be accounted for in characterization of grain size dimensionality; it is when misorientation is mirrored in neighboring grains [45]. In an article correlating ultrasonic properties and nickel alloy grain sizes, it was stated that “twin boundaries are intentionally excluded, since it is believed that the angles
between the atomic planes along the twin boundaries are very small ... not significant enough to bring about any changes in the velocities and hence they do not contribute to attenuation and backscattered noise" [46]. In the present thesis work, a comparison of twin and twin-removed conditions will be compared.

The titanium alloy studied in this thesis is unique because of the micro textured regions (MTRs) within the sample. As described by Pilchak et al, MTRs behave as macro grains because of similar orientation [11], where preferred orientations of grains in the sample enhanced backscattered signals from NDE ultrasonic interactions at the MTR boundaries [9, 12, 43]. With the MTR regions typically on the millimeter size scale, and ultrasonic wavelengths of a few hundred microns, the ultrasonic interactions occur in the geometric scattering range, where ultrasonic wave reflection, refraction, and transmission occur at the macro-scale boundaries. In this case the MTR region sizes, boundary angles, and misorientation levels become important locally. This type of material characterization is typically done using pole figures, which show the distribution of crystallographic orientation for a given area, and analysis of the grain morphology characteristics [16].

Guided by previously published work, the quantification process for the LSHR Nickel alloy involved mean grain size, grain size distribution, and misorientation characterization of local regions within the samples. With the titanium alloy, a focus was placed on characterization of orientation and sizing of the local MTR features for comparison with the NDE ultrasound measurements. The 3-dimensional microstructures were characterized using 3D grain size and misorientation quantification metrics, which were compared with 2D spatial variations in the signal patterns occurring on the outer surfaces of the 3D material volume.
4.2) **LSHR Nickel Super Alloy Quantification**

The LSHR nickel super alloy was characterized with two different approaches in mind. The first approach involved characterization of the material with and without the incorporation of twin boundaries. The second approach involved characterizing the entire sample, and then characterizing segmented localized cropped regions of the microstructure.

4.2.1) **Whole Sample Analysis- LSHR**

The EDAX TSL OIM software was used to calculate statistical data of the LSHR sample with twin boundaries initially and later without the incorporation of twin boundaries. The software has the ability to identify and cluster twinned grains into a single identified grain as shown in Figure 4.1a. The unique grain ID images depicted in Figure 4.1b and c show the LSHR sample with the original twinned boundaries in certain grains, and a second image with the removal of the twinned regions. The removal of the twinned boundaries changed the total amount grains recognized from 286,704 to 230,899 or consolidation of roughly 55,000 twined grains.
In both the twin and twin-removal cases, the mean grain size, smallest grain, largest grain, average misorientation, and largest misorientation data was collected and calculated from the output files of microstructure data using the DREAM.3D software. This data for both cases was compared with each other for variations in quantified results. Data quantified for both cases is found in chart 4.1 below. When quantitatively comparing the characterized microstructure between the two cases there is a relatively small change in mean grain size, largest grain, and number of grains. On the other hand there is a twelve-fold increase in largest...
misorientation value occurrence in the sample in the twinned quantification. Also, the average misorientation value for the sample is increased four-fold in the twinned case.

<table>
<thead>
<tr>
<th></th>
<th>Twin Boundary</th>
<th>No Twin Boundary</th>
<th>% change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number Of Grains</td>
<td>286704</td>
<td>230899</td>
<td>20</td>
</tr>
<tr>
<td>Mean Grain Size</td>
<td>20.19</td>
<td>21.66</td>
<td>7</td>
</tr>
<tr>
<td>Largest Grain</td>
<td>357.09</td>
<td>415.86</td>
<td>16</td>
</tr>
<tr>
<td>Smallest Grain</td>
<td>7.98</td>
<td>7.98</td>
<td>0</td>
</tr>
<tr>
<td>Largest Misorientation</td>
<td>4.95</td>
<td>59.99</td>
<td>1200</td>
</tr>
<tr>
<td>Smallest Misorientation</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Average MisOrientation</td>
<td>0.497</td>
<td>2.09</td>
<td>400</td>
</tr>
</tbody>
</table>

**Table 4.1:** Table comparing dimensional data for the LSHR Nickel sample in the cases of twin and twin-removed boundaries present.

**4.2.2) Characterization of Segmented Regions-LSHR**

The Nickel based super alloy was segmented into 1000 um X 8000 um slices in order to approximate localized scanning probe ultrasound measurements in the sample. The 1000 um lateral size was chosen based on a typical ultrasound beam size for a focused immersion ultrasound measurement, which can approach 1 millimeter in lateral size. By segmenting the material into slices, the local material properties can be characterized and a direct correlation with the ultrasonic measurement volume and the local microstructure properties can be made. The cropped slices progress along the entire length of the sample and were offset every 500 um to a total length of the specimen, 19000 um. This 500 um offset created segments where 50% of the microstructure was present in neighboring slices. This cropping method produced in total 37 sub-sections of the microstructure as displayed in Figure 4.2.
Figure 4.2: Cropped 1000um X 8000um segment regions of the LSHR nickel super alloy sample.
Individual segmented data sets were further analyzed in order to obtain the grain size distribution with log normal fit parameter estimates, the grain size to area fraction distribution, and the misorientation distribution plots. The grain size distribution analysis was obtained for all 37 segment positions. The binning size of the distribution was set to 3um with a plot range of 0um to 420um. The area fraction plots were plotted using the same 3um binning and plot range, with the goal of correlating grain size to area fraction. Figure 4.3 shows the results for three segments representing the three key regions of the sample including: fine, coarse, and transition grains. The entire set of segmented analysis results can be found in Appendix D.

A log normal distribution curve fit was applied to the data in all of the segment positions. Origin 2015 Pro was used to plot the grain size distributions and estimate the log normal fitting parameters, where Equation 4.1 was used to describe the log normal curve. The parameters calculated in the equation include mean (xc), standard deviation (w), area (A), and offset (yo). Input parameter includes the grain size and frequency. Table 4.2 summarizes the log normal fit parameter values of the equation for each segment's grain size distribution, and Figure 4.4 plots the mean value of the equation with that of the segments actual mean grain size.

\[
y = y_o + \frac{A}{\sqrt{2\pi}wx} e^{-\frac{\left(ln\frac{x}{xc}\right)^2}{2w^2}}
\]  

(4.1)
Figure 4.3: Log normal fits of three segmented grain locations (0-1000um, 6000-7000um, 18000-19000um), for area fraction, grain size distribution, and misorientation.
<table>
<thead>
<tr>
<th>TB</th>
<th>xc (mean)</th>
<th>w (stdev)</th>
<th>A (area)</th>
<th>NO TB</th>
<th>xc (mean)</th>
<th>w (stdev)</th>
<th>A (area)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17.35</td>
<td>0.33</td>
<td>103823</td>
<td>0</td>
<td>17.94</td>
<td>0.23</td>
<td>97514</td>
</tr>
<tr>
<td>500</td>
<td>17.12</td>
<td>0.22</td>
<td>106766</td>
<td>500</td>
<td>17.69</td>
<td>0.23</td>
<td>99648</td>
</tr>
<tr>
<td>1000</td>
<td>16.95</td>
<td>0.22</td>
<td>108661</td>
<td>1000</td>
<td>17.57</td>
<td>0.23</td>
<td>101021</td>
</tr>
<tr>
<td>1500</td>
<td>16.77</td>
<td>0.21</td>
<td>109860</td>
<td>1500</td>
<td>17.38</td>
<td>0.23</td>
<td>101448</td>
</tr>
<tr>
<td>2000</td>
<td>16.56</td>
<td>0.21</td>
<td>110452</td>
<td>2000</td>
<td>17.16</td>
<td>0.22</td>
<td>100918</td>
</tr>
<tr>
<td>2500</td>
<td>16.34</td>
<td>0.2</td>
<td>109881</td>
<td>2500</td>
<td>16.96</td>
<td>0.22</td>
<td>99647</td>
</tr>
<tr>
<td>3000</td>
<td>16.14</td>
<td>0.2</td>
<td>108538</td>
<td>3000</td>
<td>16.8</td>
<td>0.22</td>
<td>97126</td>
</tr>
<tr>
<td>3500</td>
<td>15.9</td>
<td>0.2</td>
<td>103841</td>
<td>3500</td>
<td>16.57</td>
<td>0.22</td>
<td>90827</td>
</tr>
<tr>
<td>4000</td>
<td>15.7</td>
<td>0.21</td>
<td>93190</td>
<td>4000</td>
<td>16.37</td>
<td>0.23</td>
<td>79964</td>
</tr>
<tr>
<td>4500</td>
<td>15.46</td>
<td>0.21</td>
<td>76821</td>
<td>4500</td>
<td>16.2</td>
<td>0.23</td>
<td>64282</td>
</tr>
<tr>
<td>5000</td>
<td>15.3</td>
<td>0.22</td>
<td>59054</td>
<td>5000</td>
<td>16.16</td>
<td>0.25</td>
<td>47383</td>
</tr>
<tr>
<td>5500</td>
<td>15.34</td>
<td>0.23</td>
<td>44366</td>
<td>5500</td>
<td>16.38</td>
<td>0.27</td>
<td>34220</td>
</tr>
<tr>
<td>6000</td>
<td>15.46</td>
<td>0.24</td>
<td>35051</td>
<td>6000</td>
<td>16.91</td>
<td>0.3</td>
<td>26400</td>
</tr>
<tr>
<td>6500</td>
<td>15.82</td>
<td>0.27</td>
<td>30494</td>
<td>6500</td>
<td>17.78</td>
<td>0.34</td>
<td>22179</td>
</tr>
<tr>
<td>7000</td>
<td>16.1</td>
<td>0.29</td>
<td>27560</td>
<td>7000</td>
<td>18.59</td>
<td>0.38</td>
<td>19766</td>
</tr>
<tr>
<td>7500</td>
<td>16.36</td>
<td>0.31</td>
<td>24257</td>
<td>7500</td>
<td>19.58</td>
<td>0.42</td>
<td>17271</td>
</tr>
<tr>
<td>8000</td>
<td>17.21</td>
<td>0.35</td>
<td>22849</td>
<td>8000</td>
<td>21.19</td>
<td>0.46</td>
<td>15713</td>
</tr>
<tr>
<td>8500</td>
<td>17.94</td>
<td>0.38</td>
<td>22131</td>
<td>8500</td>
<td>22.57</td>
<td>0.49</td>
<td>14862</td>
</tr>
<tr>
<td>9000</td>
<td>18.42</td>
<td>0.4</td>
<td>20411</td>
<td>9000</td>
<td>23.66</td>
<td>0.51</td>
<td>13432</td>
</tr>
<tr>
<td>9500</td>
<td>19.03</td>
<td>0.42</td>
<td>18473</td>
<td>9500</td>
<td>24.54</td>
<td>0.54</td>
<td>11904</td>
</tr>
<tr>
<td>10000</td>
<td>19.58</td>
<td>0.44</td>
<td>16912</td>
<td>10000</td>
<td>25.95</td>
<td>0.55</td>
<td>10729</td>
</tr>
<tr>
<td>10500</td>
<td>20.35</td>
<td>0.47</td>
<td>16180</td>
<td>10500</td>
<td>27.41</td>
<td>0.57</td>
<td>10094</td>
</tr>
<tr>
<td>11000</td>
<td>20.65</td>
<td>0.48</td>
<td>15433</td>
<td>11000</td>
<td>28.24</td>
<td>0.59</td>
<td>9640</td>
</tr>
<tr>
<td>11500</td>
<td>20.46</td>
<td>0.47</td>
<td>14729</td>
<td>11500</td>
<td>29.04</td>
<td>0.59</td>
<td>9241</td>
</tr>
<tr>
<td>12000</td>
<td>20.83</td>
<td>0.48</td>
<td>13477</td>
<td>12000</td>
<td>30.1</td>
<td>0.59</td>
<td>8237</td>
</tr>
<tr>
<td>12500</td>
<td>20.92</td>
<td>0.49</td>
<td>13002</td>
<td>12500</td>
<td>30.55</td>
<td>0.61</td>
<td>8027</td>
</tr>
<tr>
<td>13000</td>
<td>21.51</td>
<td>0.49</td>
<td>14027</td>
<td>13000</td>
<td>31.26</td>
<td>0.62</td>
<td>8340</td>
</tr>
<tr>
<td>13500</td>
<td>21.85</td>
<td>0.51</td>
<td>12911</td>
<td>13500</td>
<td>32.24</td>
<td>0.64</td>
<td>7547</td>
</tr>
<tr>
<td>14000</td>
<td>21.89</td>
<td>0.51</td>
<td>12065</td>
<td>14000</td>
<td>33.6</td>
<td>0.66</td>
<td>7126</td>
</tr>
<tr>
<td>14500</td>
<td>22.57</td>
<td>0.54</td>
<td>11493</td>
<td>14500</td>
<td>34.89</td>
<td>0.69</td>
<td>6759</td>
</tr>
<tr>
<td>15000</td>
<td>23.44</td>
<td>0.56</td>
<td>11251</td>
<td>15000</td>
<td>35.88</td>
<td>0.68</td>
<td>6531</td>
</tr>
<tr>
<td>15500</td>
<td>22.58</td>
<td>0.53</td>
<td>11145</td>
<td>15500</td>
<td>34.8</td>
<td>0.68</td>
<td>6547</td>
</tr>
<tr>
<td>16000</td>
<td>23.44</td>
<td>0.56</td>
<td>11251</td>
<td>16000</td>
<td>37.13</td>
<td>0.7</td>
<td>6269</td>
</tr>
<tr>
<td>16500</td>
<td>23.23</td>
<td>0.55</td>
<td>10706</td>
<td>16500</td>
<td>37.92</td>
<td>0.68</td>
<td>6012</td>
</tr>
<tr>
<td>17000</td>
<td>23.24</td>
<td>0.55</td>
<td>10519</td>
<td>17000</td>
<td>37.03</td>
<td>0.69</td>
<td>5944</td>
</tr>
<tr>
<td>17500</td>
<td>23.07</td>
<td>0.55</td>
<td>10680</td>
<td>17500</td>
<td>36.79</td>
<td>0.7</td>
<td>6237</td>
</tr>
<tr>
<td>18000</td>
<td>22.44</td>
<td>0.54</td>
<td>10375</td>
<td>18000</td>
<td>36.41</td>
<td>0.71</td>
<td>6118</td>
</tr>
</tbody>
</table>

Table 4.2 Summary chart of log normal fit parameters for LSHR nickel segmented positions.
4.3) MTR Titanium Alloy Sample Quantification

The micro-texture titanium alloy was characterized using two approaches. The first approach included looking at the entire sample, where a basic understanding and characterization of the overall MTR crystallographic orientations, feature orientations, sizes, locations, and geometric shapes were studied. The second approach involved segmenting the sample into 49 vertical segments (1000 micron by 8500 microns), and characterizing the microstructure features of each segment separately.

Before the characterization process, the titanium EBSD scan data was subjected to a cleanup process to improve the microstructure property information by removing noisy data points and information. In the EDAX TSL OIM software there are data cleanup processes available, which allow for the indexing of crystallographic orientation based on the quality of the measured diffraction patterns [27]. An EBSD confidence index value is available for each measurement point, where the index falls on a scale from 0 to 1, with a value of 1 being a
measurement with high confidence and low noise in the diffraction pattern [27]. Using a confidence index threshold of 0.15, the total number of identified grains was reduced from 1,855,994 to 736,532, or 40% of the original value. Figure 4.5b shows a comparison of the original titanium IPF map and the new map with the confidence index cleanup applied.

**Figure 4.5:** a. the original sample without clean up b. Viewgraph depicts the cleaned and cropped sample used for further analysis c. IPF Color Scale.
4.3.1) Whole Sample Analysis- MTR Ti

Characterization of the entire MTR sample was based on quantifying the preferred orientations of distinguishing features of the material based on IPF map indications. The quantification process included the identification of textured regions within the sample, calculating the dominant orientation within each region, and sizing the feature based on the number of elements within the region.

Using the Inverse Pole Figure (IPF) maps, the EBSD scan was hand segmented by identifying the visual differences of orientations in the IPF mapping based on IPF color coded differences. The hand segment identified thirteen different MTR regions within the sample as shown in Figure 4.6. After identifying the thirteen different MTR regions, the sample data set was cropped using the EDAX TSL OIM software into the thirteen regions so that the pole figure for each cropped region could be obtained. The pole figure shows the crystallographic orientation of elements within the region represented by the elements Euler angles. Cropping the sample also gave an estimate of the number of point with the region as displayed in Table 4.3. The cropping is just an estimation because it is difficult to visually crop the regions to exact boundaries because of the complexity of the sample. However, the basic process did provide an initial estimation of the dominant orientations within each MTR region, which is important for understanding the elastic wave propagation and scattering processes within the material.

Using the EDAX TSL OIM software, each cropped region and its orientation and sizing properties were output into a text file for further analysis. The file contained key grain orientation data represented by Euler angles. A custom Matlab code was used to estimate which orientation was dominant for each segmented region. Table 4.4 highlights the dominant orientations for all thirteen regions and the pole figure for the whole sample.
Figure 4.6 a. Pole Figure of whole sample b. Hand segmentation of the MTR Titanium sample into thirteen key regions.
Figure 4.7 Pole figures for the thirteen segmented regions showing dominant orientations within each region.

Table 4.3 Chart detailing number of points within each hand segmented region.
Table 4.4 Chart detailing dominant crystallographic orientations for each of the thirteen regions.

4.3.2) Characterization of Segmented Regions – MTR Ti

The micro-textured titanium alloy was segmented in 1000 um X 8500 um slices in order to approximate scanning probe ultrasound measurements of localized regions within the sample. The cropped slices progress the whole length of the sample and were offset every 500 um to a total length of the specimen, 24000 um. This 500 um offset created segments where 50% of the microstructure was present in neighboring slices. This cropping method produced a total of 39 sub-sections of the microstructure as displayed in Figure 4.8. As with the LSHR sample, the reasoning for the segmentation involves the ability to be able to work with localized regions of the microstructure to do a comparative analysis of the ultrasonic interaction with a smaller, localized subset of the quantified sample.
Individual segmented regions were further analyzed in order to obtain the pole figure characteristics and dominant MTR orientations in each individual segment. A custom Matlab code was used to determine the dominant MTR orientation in terms of Euler angles based on the EDAX TSL OIM software output. Figure 4.9 provides pole figures and IPF maps for four
different segments representing four different regions within the MTR titanium sample. The positions corresponded to segments 0-1000um, 2000-4000um, 11000-12000um and 19000-20000um. Additional details and results for the other segment regions will be shown in appendix D.

![Figure 4.9 Pole figures (PF) for four specified locations (0-1000, 2000-3000, 11000-12000, 19000-2000), quantification of orientation.](image)

**Figure 4.9** Pole figures (PF) for four specified locations (0-1000, 2000-3000, 11000-12000, 19000-2000), quantification of orientation.

### 4.4) 3-Dimensional Microstructures Quantification

Both 3-dimensional microstructures were characterized on the basis of quantifying grain size and misorientation data within the full 3-dimensional volume. The synthetically generated equiaxed microstructure had a 3-dimensional volume of 200um x 200um x 200um. The calculated mean grain size was 33.30 um, with the smallest grain being 22.36 um and the largest grain 45.08 um. The total number of grains in the sample is 399. Plots for grain size distribution
of equivalent diameters, area fraction and misorientation are provided in Figure 4.10. A log normal fit distributions have been superimposed over the plots in Figure 4.10.

![Image of synthetic 3-dimensional equiaxed nickel microstructure highlighting a. grain size distribution, b. area fraction, and c. misorientation.]

**Figure 4.10** Grain quantification of synthetic 3-dimensional equiaxed nickel microstructure highlighting a. grain size distribution, b. area fraction, and c. misorientation.

The Inconel 100 sample showed more complexity in its microstructure compared to the synthetic 3-dimensional microstructure. The 3-dimensional volume of the Inconel 100 sample was 121um x 121um x 117um. There were a total of 1347 grains. The mean grain size was 2.62um with the smallest grain diameter of 0.31um and the largest grain diameter of 8.64um. The plots for grain size distribution, area fraction, and misorientation are provided in Figure 4.11.

---

72
Figure 4.11 Grain quantification of the Inconel 100 3d volume highlighting a. Grain Size Distribution b. Area Fraction c. Misorientation.
5.1) **Chapter Introduction and Background**

NDE Ultrasound models are becoming important in providing the capability to understand and assess critical attributes of advanced material systems [3]. As Dr Michael Groeber alludes to “the advancement in microstructure-property relationship, major advances in digital representations of microstructure, and computational power are yielding models with increasing complexity and accuracy [19]. Traditional representations of digital microstructures have been of nominal sizes and generalized properties of the structure [19]. With the advancement of tools such as DREAM.3D and the concept of ICME, models are becoming an effective and efficient way to develop, evaluate, and optimize advanced material systems. NDE sensing models coupled with accurate digital representations of microstructures are allowing the possibility to link physics-based NDE models with material properties and structures and understanding key relationships in a virtual environment [3, 19].

5.2) **General Model Goals and Conditions**

The general goals of the model studies were to test the feasibility of the developed ICME framework and to use it to study ultrasonic scattering phenomena of wave interactions as a result of variations and complexities within the microstructure properties. Up until recently, the incorporation of realistic microstructures with crystallographic data has been limited. The
framework created and described in Chapter 2 allows for the incorporation of 2-dimensional and 3-dimensional data from realistic and synthetic data sets to be put into computational models with increased detail. The ICME framework allows for the description of all elements within the models to be accurately represented with its polycrystalline microstructure properties. Rotated crystallographic elastic stiffness matrix, density, and spatially resolved coordinate values are all used to describe the microstructure states within the PZFlex models.

The nickel based super alloy and MTR titanium alloy were used to study two types of NDE ultrasonic models: pitch-catch and pulse echo. In each of the cases, longitudinal ultrasonic waves were propagated at 1 MHz, 5 MHz and 10 MHz frequencies. For the 3-dimensional microstructures, longitudinal ultrasonic waves were propagated through the material using the pitch-catch method, where the ultrasonic wavelength to grain interaction ratio was approximately one (stochastic scattering regime). Table 5.1 shows a breakdown of the number of models ran for each microstructure sample.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Pitch Catch Models</th>
<th>Pulse Echo Models</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 MHz</td>
<td>5 MHz</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Nickel Super alloy</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>MTR Titanium Alloy</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>3D Synthetic</td>
<td>3 Models, longitudinal wave length ( \lambda_d \sim 1 ), stochastic scattering</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3D IN 100</td>
<td>1 Models, longitudinal wave length ( \lambda_d \sim 1 ), stochastic scattering</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Grand total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 Chart detailing the amount and types of models ran on the microstructure data sets. All together 520 models were run and used in the thesis studies.

All models used a pressure loaded idealized longitudinal wave for propagation and interaction with the polycrystalline materials at three different frequencies. Discretion of the model domain was set to 20 elements per ultrasonic wavelength, which provided adequate
stability and convergence of model for representation of grain features properly [8]. The simulation time for the models varied, but was set long enough to permit the wave to travel the length of each sample. The timestep was set internally by the PZFlex model, and was set to a level that provided adequate model stability based on the ultrasonic properties and the model grid element size. At designated time steps, a screen capture image of the elastic wave propagating through the sample was taken.

The pulse-echo and pitch-catch models collected displacement data for ultrasonic signal analysis. In the pulse-echo models, the wave propagated along the y-axis and the wave was collected back at the top surface of the model where the original excitation occurred (Figure 5.1). For this style of wave interaction the top surface of the model was set to a free boundary condition, the side edges of the model were set to symmetry boundary conditions, and the bottom of the model was set to free or absorbing boundary conditions depending on the model type. In the pitch-catch models, the wave again propagated along the y-axis and the wave displacement data was collected on the bottom surface of the model. The boundary conditions and excitation for the pitch-catch models were similar to the pulse-echo models. The pitch-catch models were used to collect wave and signal information for thru-transmission displacement signals and patterns. The pulse-echo models were used to collect attenuation, backscatter signal, and pattern information for analysis.
5.3) **LSHR Nickel Specific Models and Highlighted Results**

Models were ran on all 37 segmented sections of the LSHR nickel super alloy in a scanning probe style measurement where the wave interaction was with a localized region of the microstructure. Using the ICME framework described in Chapter 2, the 21 rotated elastic stiffness matrix values were calculated for each x-y spatial location in each model using a reference orientation matrix for the nickel sample. The nickel density of 7500 kg/m³ was also used in the models. Each segmented microstructure was incorporated into three pulse-echo and three pitch-catch measurement models at 1 MHz, 5 MHz, and 10 MHz, for a total of six models per segmentation.
Figure 5.2 depicts time-resolved 5MHz ultrasound waves propagating along the y-axis through three segmented microstructures representing the coarse, fine, and transition grain regions in the microstructure. Only three variations of the models are shown in Figure 5.2 out of the 222 total models run for the LSHR nickel microstructure. Figure 5.2 depicts the pressure and displacement magnitude at certain time steps of the wave propagating through the sample. Visual observation of the model running shows subtle scattering features after the initial wave propagates through the sample. Further investigation and correlation will be presented in Chapter 6.
Figure 5.2  a. 5MHz longitudinal wave propagation through the small grain region segment at different model time steps, b. the transition region, and c. the coarse grain region. Each model shows subtle scattering occurring after the initial wave.

Pulse-echo backscatter signals and pitch-catch thru-transmission signals were collected for all 222 models ran for further analysis of the y-displacement signals (Figures 5.3 and 5.4). Also output were images of maximum x-displacement, y-displacement and pressure levels occurring within each segmented microstructure model. These maximum displacement and pressure images provide energy flow and maximum grain interaction information. Figure 5.5
highlights these output images for three specified segmented pitch-catch models at a 5 MHz frequency. A general observation of these results suggest lobes of energy are created within the bulk of the microstructure that correlate to the sizes of grains.

Figure 5.3. Representative y-displacement pulse-echo signals for the LSHR nickel ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.
Figure 5.4. Representative $y$-displacement pitch-catch signals for the LSHR nickel ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.
Figure 5.5 a. The maximum displacement of the elastic waves in x and y directions and pressure for the small grain, b. transition grain, and c. coarse grain regions.
5.4) MTR Titanium Alloy Specific Models and Highlighted Result

NDE ultrasound sensing models were ran on all 49 1000um X 8500um segmented sections of micro-textured titanium alloy in a scanning probe style measurement where the wave interaction was localized within each segmented region of the microstructure. The ICME framework was again used to calculate and input the 21 value rotated elastic stiffness matrix values for each position in the segmented microstructure data sets. The titanium density of 4430 kg/m3 and reference stiffness matrix values were used for the processing.

Each cropped segmentation was incorporated into three pulse-echo and three pitch-catch models at 1 MHz, 5 MHz, and 10 MHz ultrasound frequencies for a total of six model per segmentation. Figure 5.6 shows model results for four microstructure segments representing grains with different degrees of preferred orientation. Only four variations of the total 294 models are shown in Figure 5.6. The models shown in Figure 5.6 are for a 10 MHz ultrasound frequency with pitch-catch measurement model. Figure 5.6 shows the displacement magnitude and pressure levels at certain time steps of the wave propagating though the microstructures.
Figure 5.6 a. Wave propagation through a titanium region with minor texture shown at different time steps, b. a texture region with indications of a small macro grain, c. a large well-defined macro grain, d. and two regions of texture.

Pulse-echo backscatter and pitch-catch thru-transmission γ-displacement signals we collected for all 294 models for further analysis of the signal content (Figures 5.7 and 5.8). Also output were images of the maximum x and y displacements and pressure levels for each segmented microstructure region. Figure 5.9 highlights these output images for the four
specified segmentations of pitch-catch models at a 10 MHz frequency. General observations show elastic wave boundary interactions for the segments where the macro grain has a dominant and preferred orientation.

Figure 5.7. Representative y-displacement pulse-echo signals for the MTR titanium ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.
Figure 5.8. Representative y-displacement pitch-catch signals for the MTR titanium ultrasonic sensing models for fine grain (top), transition region (center), and coarse grain (bottom) regions, and for the three ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.
Figure 5.9 a. Maximum displacements of the wave in x and y directions and pressure for the minor texture region, b. small macro grain region, c. large preferred orientation macro grain region, d. and two textured regions.
5.5) 3-Dimensional Microstructure Samples

5.5.1) Synthetic Equiaxed Models

Models were run on the synthetic equiaxed 3-dimensional microstructure using the same idealized pressure loaded longitudinal wave, propagating in the y-direction through an xz planes of the structure. Absorbing boundary conditions were applied to the opposite face and symmetry boundary conditions applied to the four faces along the propagation direction. The longitudinal wavelength was again chosen to allow for stochastic regime scattering with $k_0d \sim 1$. Since the microstructure was synthetically created it was given density and crystallographic orientation information of a nickel structure. As the wave propagates through the 3-dimensional volume information was collected on the maximum displacements. Figure 5.10 shows instances of the wave propagating through the volume and displacement of the wave as it propagated through the y planes.
5.5.2) \textit{In100 Models}

Models were run on the Inconel 100 3-dimensional microstructure. Idealized longitudinal waves were again initiated along the entire right side of the model geometry (yz plane x-minimum), propagating towards the left as depicted in Figure 5.11, with absorbing boundary conditions applied to the opposite face and symmetry boundary conditions applied to the four faces along the propagation direction. The longitudinal wavelength was again chosen to allow for stochastic regime scattering with $k_0d \sim 1$. As the wave propagates through the 3-dimensional volume information was collected on the maximum displacements. Figure 5.11 shows instances of the wave propagating through the volume and maximum Y-displacement of the wave as on the outer surface of the 3d volume and also on a 2D internal mapping of a layer within the microstructure. Qualitative analysis of correlation of displacement and sizing features will be further explored in chapter 6.
Figure 5.11 a. Time-resolved propagation of a longitudinal wave across the x plane and peak outer surface displacement of wave, and b. maximum displacement of the wave on an internal surface with the model.
CHAPTER 6

RESULTS AND DISCUSSION

6.1) Introduction

Prior chapters highlighted the ability to take realistic microstructure data sets and create a virtual representation of the polycrystalline grains for incorporation into NDE ultrasound sensing models. Chapter 4 highlighted the approach taken to quantify meaningful material property information for correlation with sensing model output results. Chapter 5 highlighted the model conditions and types of models run for 2-dimensional LSHR dual-microstructure nickel and micro-textured (MTR) titanium samples, and 3-dimensional synthetic and realistic nickel samples. The intention of the present chapter is to show summary results for quantifying the microstructure properties, show trends in the model output summary results for the segmented microstructure regions, and lastly to make meaningful qualitative observations and correlations between microstructure variation and model signals.

6.2) LSHR Results

Measurements and characterization done with the nickel-based super alloy sample were done in a method that simulated a scanning probe measurement style, meaning that ultrasound waves interact with local microstructure segments in the transverse direction as the ultrasound sensor position is scanned along the material surface. By segmenting the microstructure, a quantification of the local microstructure properties can be made, and signals
can be compared with local grain quantification as seen in figure 6.1. The ICME framework developed as part of this thesis provides a new capability for conducting such studies in a virtual manner using computational models, which is difficult or impossible to do using traditional methods. A direct comparison of the quantified local properties with the local NDE sensing signals can then be used to develop new NDE mapping tools for material property estimation and validation.

Figure 6.1. Segmentation of the LSHR nickel sample and example segment analysis.

6.2.1) Summary of Microstructure Quantification- LSHR

Statistical data was quantified for each segmented region to include: the number of grains, as large as grain (um), smallest grain (um), mean grain size (um), average misorientation (degrees), and largest misorientation (degrees) as outline in Tables 6.1 and 6.2. The table results show a segment-by-segment comparison of the cases with twin boundaries (Table 6.1), and with
the twin boundaries removed (Table 6.2). Data in the tables were calculated in Microsoft Excel using the *.csv file output from DREAM.3D, which provided grain size and misorientation data.

<table>
<thead>
<tr>
<th>Position</th>
<th># Of Grains</th>
<th>ALA Grain Size</th>
<th>Smallest Grain Size</th>
<th>Mgs</th>
<th>Average Misorientation</th>
<th>Largest Misorientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>34672</td>
<td>54.41</td>
<td>7.98</td>
<td>16.73</td>
<td>0.447</td>
<td>3.63</td>
</tr>
<tr>
<td>500</td>
<td>35780</td>
<td>44.06</td>
<td>7.98</td>
<td>16.47</td>
<td>0.455</td>
<td>4.15</td>
</tr>
<tr>
<td>1000</td>
<td>36525</td>
<td>44.06</td>
<td>7.98</td>
<td>16.30</td>
<td>0.459</td>
<td>4.15</td>
</tr>
<tr>
<td>1500</td>
<td>37241</td>
<td>42.22</td>
<td>7.98</td>
<td>16.14</td>
<td>0.465</td>
<td>4.41</td>
</tr>
<tr>
<td>2000</td>
<td>37752</td>
<td>57.26</td>
<td>7.98</td>
<td>16.02</td>
<td>0.473</td>
<td>4.41</td>
</tr>
<tr>
<td>2500</td>
<td>38044</td>
<td>69.33</td>
<td>7.98</td>
<td>15.91</td>
<td>0.479</td>
<td>4.79</td>
</tr>
<tr>
<td>3000</td>
<td>37855</td>
<td>142.73</td>
<td>7.98</td>
<td>15.86</td>
<td>0.486</td>
<td>3.41</td>
</tr>
<tr>
<td>3500</td>
<td>36666</td>
<td>142.73</td>
<td>7.98</td>
<td>15.88</td>
<td>0.499</td>
<td>3.69</td>
</tr>
<tr>
<td>4000</td>
<td>33513</td>
<td>197.47</td>
<td>7.98</td>
<td>16.13</td>
<td>0.514</td>
<td>4.36</td>
</tr>
<tr>
<td>4500</td>
<td>28506</td>
<td>197.47</td>
<td>7.98</td>
<td>16.70</td>
<td>0.533</td>
<td>4.36</td>
</tr>
<tr>
<td>5000</td>
<td>22603</td>
<td>231.52</td>
<td>7.98</td>
<td>17.72</td>
<td>0.548</td>
<td>3.41</td>
</tr>
<tr>
<td>5500</td>
<td>17636</td>
<td>231.52</td>
<td>7.98</td>
<td>19.18</td>
<td>0.556</td>
<td>3.41</td>
</tr>
<tr>
<td>6000</td>
<td>14411</td>
<td>245.86</td>
<td>7.98</td>
<td>20.72</td>
<td>0.552</td>
<td>4.3</td>
</tr>
<tr>
<td>6500</td>
<td>12707</td>
<td>245.86</td>
<td>7.98</td>
<td>21.99</td>
<td>0.546</td>
<td>3.61</td>
</tr>
<tr>
<td>7000</td>
<td>11559</td>
<td>224.33</td>
<td>7.98</td>
<td>23.08</td>
<td>0.546</td>
<td>3.04</td>
</tr>
<tr>
<td>7500</td>
<td>10310</td>
<td>229.66</td>
<td>7.98</td>
<td>24.36</td>
<td>0.546</td>
<td>3.61</td>
</tr>
<tr>
<td>8000</td>
<td>9432</td>
<td>256.57</td>
<td>7.98</td>
<td>25.47</td>
<td>0.545</td>
<td>4.27</td>
</tr>
<tr>
<td>8500</td>
<td>8924</td>
<td>242.73</td>
<td>7.98</td>
<td>26.19</td>
<td>0.536</td>
<td>4.27</td>
</tr>
<tr>
<td>9000</td>
<td>8190</td>
<td>272.51</td>
<td>7.98</td>
<td>27.18</td>
<td>0.528</td>
<td>4.12</td>
</tr>
<tr>
<td>9500</td>
<td>7344</td>
<td>294.03</td>
<td>7.98</td>
<td>28.40</td>
<td>0.525</td>
<td>2.66</td>
</tr>
<tr>
<td>10000</td>
<td>6677</td>
<td>357.09</td>
<td>7.98</td>
<td>29.51</td>
<td>0.517</td>
<td>3.86</td>
</tr>
<tr>
<td>10500</td>
<td>6319</td>
<td>329.41</td>
<td>7.98</td>
<td>30.52</td>
<td>0.513</td>
<td>3.86</td>
</tr>
<tr>
<td>11000</td>
<td>6023</td>
<td>304.51</td>
<td>7.98</td>
<td>31.23</td>
<td>0.511</td>
<td>3.66</td>
</tr>
<tr>
<td>11500</td>
<td>5828</td>
<td>313.72</td>
<td>7.98</td>
<td>31.62</td>
<td>0.512</td>
<td>3.16</td>
</tr>
<tr>
<td>12000</td>
<td>5535</td>
<td>280.06</td>
<td>7.98</td>
<td>32.68</td>
<td>0.508</td>
<td>2.73</td>
</tr>
<tr>
<td>12500</td>
<td>5251</td>
<td>307.42</td>
<td>7.98</td>
<td>32.82</td>
<td>0.509</td>
<td>4.39</td>
</tr>
<tr>
<td>13000</td>
<td>5449</td>
<td>306.74</td>
<td>7.98</td>
<td>32.78</td>
<td>0.516</td>
<td>4.39</td>
</tr>
<tr>
<td>13500</td>
<td>4985</td>
<td>339.87</td>
<td>7.98</td>
<td>33.70</td>
<td>0.523</td>
<td>3.8</td>
</tr>
<tr>
<td>14000</td>
<td>4731</td>
<td>353.01</td>
<td>7.98</td>
<td>34.48</td>
<td>0.518</td>
<td>3.87</td>
</tr>
<tr>
<td>14500</td>
<td>4456</td>
<td>318.95</td>
<td>7.98</td>
<td>35.52</td>
<td>0.505</td>
<td>4.95</td>
</tr>
<tr>
<td>15000</td>
<td>4293</td>
<td>275.41</td>
<td>7.98</td>
<td>36.36</td>
<td>0.503</td>
<td>2.6</td>
</tr>
<tr>
<td>15500</td>
<td>4372</td>
<td>318.95</td>
<td>7.98</td>
<td>35.91</td>
<td>0.510</td>
<td>3.1</td>
</tr>
<tr>
<td>16000</td>
<td>4293</td>
<td>275.41</td>
<td>7.98</td>
<td>36.36</td>
<td>0.503</td>
<td>2.6</td>
</tr>
<tr>
<td>16500</td>
<td>4166</td>
<td>310.77</td>
<td>7.98</td>
<td>36.91</td>
<td>0.501</td>
<td>2.2</td>
</tr>
<tr>
<td>17000</td>
<td>4104</td>
<td>310.77</td>
<td>7.98</td>
<td>37.11</td>
<td>0.504</td>
<td>2.09</td>
</tr>
<tr>
<td>17500</td>
<td>4178</td>
<td>290.6</td>
<td>7.98</td>
<td>36.81</td>
<td>0.512</td>
<td>2.98</td>
</tr>
<tr>
<td>18000</td>
<td>4119</td>
<td>321.14</td>
<td>7.98</td>
<td>36.72</td>
<td>0.513</td>
<td>2.98</td>
</tr>
</tbody>
</table>

**Table 6.1:** Table showing calculated statistical data of dimensional and misorientation features for segmented LSHR nickel super alloy with twin boundaries remaining.
<table>
<thead>
<tr>
<th>Position</th>
<th># Of ALA Grain</th>
<th>Smallest Grain Mgs</th>
<th>Average Misorientation</th>
<th>Largest Misorientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32284</td>
<td>54.41</td>
<td>7.98</td>
<td>1.24</td>
</tr>
<tr>
<td>500</td>
<td>33169</td>
<td>49.18</td>
<td>7.98</td>
<td>1.31</td>
</tr>
<tr>
<td>1000</td>
<td>33618</td>
<td>49.18</td>
<td>7.98</td>
<td>1.42</td>
</tr>
<tr>
<td>1500</td>
<td>34094</td>
<td>43.7</td>
<td>7.98</td>
<td>1.48</td>
</tr>
<tr>
<td>2000</td>
<td>34351</td>
<td>61.8</td>
<td>7.98</td>
<td>1.56</td>
</tr>
<tr>
<td>2500</td>
<td>34332</td>
<td>72.25</td>
<td>7.98</td>
<td>1.66</td>
</tr>
<tr>
<td>3000</td>
<td>33737</td>
<td>163.42</td>
<td>7.98</td>
<td>1.73</td>
</tr>
<tr>
<td>3500</td>
<td>32111</td>
<td>163.42</td>
<td>7.98</td>
<td>1.84</td>
</tr>
<tr>
<td>4000</td>
<td>28752</td>
<td>283.11</td>
<td>7.98</td>
<td>1.95</td>
</tr>
<tr>
<td>4500</td>
<td>23765</td>
<td>283.11</td>
<td>7.98</td>
<td>2.03</td>
</tr>
<tr>
<td>5000</td>
<td>18096</td>
<td>234.6</td>
<td>7.98</td>
<td>2.12</td>
</tr>
<tr>
<td>5500</td>
<td>13490</td>
<td>254.39</td>
<td>7.98</td>
<td>2.21</td>
</tr>
<tr>
<td>6000</td>
<td>10539</td>
<td>259.16</td>
<td>7.98</td>
<td>2.32</td>
</tr>
<tr>
<td>6500</td>
<td>8863</td>
<td>259.16</td>
<td>7.98</td>
<td>2.45</td>
</tr>
<tr>
<td>7000</td>
<td>7855</td>
<td>279.49</td>
<td>7.98</td>
<td>2.58</td>
</tr>
<tr>
<td>7500</td>
<td>6842</td>
<td>264.15</td>
<td>7.98</td>
<td>2.72</td>
</tr>
<tr>
<td>8000</td>
<td>6085</td>
<td>312.91</td>
<td>7.98</td>
<td>2.86</td>
</tr>
<tr>
<td>8500</td>
<td>5618</td>
<td>332.39</td>
<td>7.98</td>
<td>3.00</td>
</tr>
<tr>
<td>9000</td>
<td>5030</td>
<td>278.92</td>
<td>7.98</td>
<td>3.15</td>
</tr>
<tr>
<td>9500</td>
<td>4444</td>
<td>316</td>
<td>7.98</td>
<td>3.29</td>
</tr>
<tr>
<td>10000</td>
<td>3981</td>
<td>360.6</td>
<td>7.98</td>
<td>3.44</td>
</tr>
<tr>
<td>10500</td>
<td>3722</td>
<td>332.2</td>
<td>7.98</td>
<td>3.59</td>
</tr>
<tr>
<td>11000</td>
<td>3519</td>
<td>248.86</td>
<td>7.98</td>
<td>3.75</td>
</tr>
<tr>
<td>11500</td>
<td>3370</td>
<td>356.56</td>
<td>7.98</td>
<td>3.91</td>
</tr>
<tr>
<td>12000</td>
<td>3019</td>
<td>329.85</td>
<td>7.98</td>
<td>4.07</td>
</tr>
<tr>
<td>12500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>4.23</td>
</tr>
<tr>
<td>13000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>4.39</td>
</tr>
<tr>
<td>13500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>4.55</td>
</tr>
<tr>
<td>14000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>4.72</td>
</tr>
<tr>
<td>14500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>4.88</td>
</tr>
<tr>
<td>15000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.04</td>
</tr>
<tr>
<td>15500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.21</td>
</tr>
<tr>
<td>16000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.37</td>
</tr>
<tr>
<td>16500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.53</td>
</tr>
<tr>
<td>17000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.69</td>
</tr>
<tr>
<td>17500</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>5.86</td>
</tr>
<tr>
<td>18000</td>
<td>2928</td>
<td>329.85</td>
<td>7.98</td>
<td>6.02</td>
</tr>
</tbody>
</table>

**Table 6.2:** Table showing calculated statistical data of dimensional and misorientation features for segmented LSHR nickel super alloy without twin boundaries.

The statistical data in Table’s 6.2 and 6.3 were plotted in a series of charts to see trends in the microstructure properties as the positions of segment regions were systematically moved over the length of the sample. Plots show trends for both twin and twin-removed quantified
properties on each chart. The data that was analyzed included: number of grains vs position, as-large-as grain vs position, largest misorientation value vs position, and average misorientation vs position. These plots are shown in Figure’s 4.3 and 4.4 below. General observations of the results show a relatively flat mean grain size of ~15um in the fine grain region found on the left side of the sample (0um to 4000um), and a smooth linear increase in mean grain size from 20um to 50um as the material transitions from fine to coarse grains (between 4000um and 18000um positions). The number of grains range from ~30,000 in the fine grain regions to ~3,000 in the coarse grain region with a sharp drop in the transition zone. The as-large-as results show a largest grain of ~50um in the fine grain region and maximum grain sizes of ~350um in the coarse grain region. Average misorientation increase in a smooth, linear manner from the fine to coarse region and plateau in large grain region of the sample. The values of largest misorientation shows no visual trends in data however largest values of significantly higher with the removal of twin boundaries.
Figure 6.2: a. Position vs. Number of Grains, b. Position vs. As-Large-As Grain, and c. Position vs. Mean Grain Size
6.2.2) *LSHR Nickel Model Results*

For the LSHR Sample there were 6 variations of models ran to look at backscatter and attenuation of ultrasonic elastic wave signals for each segmented model position. Figures 6.4 and 6.5 shows the peak-to-peak signal analysis of the average \(\gamma\)-displacement of propagating elastic waves for each scan position. This analysis was done for the pitch-catch thru-transmission and pulse-echo backscatter sensing model cases. Also a standard deviation analysis was done for each signal to quantify how variable the signal content was in each region due to local microstructure variations. The peak-to-peak analysis for average signal and standard deviation was done for all segmented model outputs, and is summarized in the figures below for the 1MHz, 5MHz, and 10MHz ultrasound frequencies.
Figure 6.4: Pulse-echo average signal peak-to-peak analysis and standard deviation for NDE sensing models for LSHR nickel microstructure.
6.2.3) Discussion and Correlation

Model results were compared with the LSHR nickel microstructure properties calculated without the incorporation of twin boundaries as suggested by Thompson et. al. [46]. When comparing the sensing models with microstructure characterization there are some interesting trends within the data that can be observed. For the 1 MHz pulse-echo and pitch-catch models, unfortunately little comparison can be made. The 1 MHz frequency creates an ultrasound wavelength that is much greater that the diameter of the grains resulting in Rayleigh regime scattering. Although there is a third order dependency on grain size and fourth order
dependency on ultrasound frequency for signal attenuation levels, the signal variability was somewhat random as you move from the fine to coarse grain regions.

The 5 MHz and 10 MHz signal summary offers much interesting trends in the model output versus microstructure statistics. The pitch-catch sensing models show correlation between mean grain size and as-large-as variations with segmented position and the peak-to-peak levels of backscatter signal as depicted in Figure 6.6. The computational model results are promising and show feasibility for correlating grain size features with ultrasound signal content using the ICME framework approach.

Figure 6.6: Correlation of 5 MHz and 10 MHz signal analysis with mean grain size statistics and as-large-as grain feature sizes.
The pulse-echo model results also show interesting trends, with a correlation that is inverse to the pitch-catch sensing model results. In the 5 MHz and 10 MHz sensing model results there is a general decrease in the peak-to-peak signal levels as grain features progress from fine grain to coarse grain sizes. The decrease follows standard signal attenuation trends for increasing grain sizes in the Rayleigh and stochastic scattering regimes. The 10 MHz sensing model results, in particular, show a trend and slope in the transition region that closely follows the slope of the number of grains as shown in Figure 6.7

Figure 6.7: Correlation of 5 MHz and 10 MHz signal analysis with number of grains per segment.

6.3) **F110 Micro-Textured Titanium Sample**

Measurements and characterization of the micro-textured titanium alloy were also done in a method that simulated a scanning ultrasound probe style measurement. The segmented regions were chosen to be 1 mm in transverse width, which mimics the width of a focused ultrasound transducer beam as described previously in the thesis. Figure 6.8 shows
representative segmented sections and characterization results for the right-most position, which focused on pole figure analysis of the MTR regions and their sizes and geometric features.

![Figure 6.8. Segmentation of the MTR titanium sample and example segment analysis.](image)

6.3.1) **Summary of Microstructure Quantification**

Since there are preferred orientations in the micro-textured titanium sample, a basic understanding and characterization of the overall MTR crystallographic orientation and feature sizes/shapes within each segmented region was of interest. As described in Chapter 4, 13 different MTR regions were identified in the sample, which were characterized by orientation size, position, and relative angle. Further analysis was done to determine the dominant orientation, MTR feature which is highlighted in Figure 6.9, Figure 6.10, and Table 6.3. An analysis of the intensity level of the dominant orientation in Figure 10 shows a strong, oriented MTR grain between segmented positions 10,000 – 12,000 um. A comparison of the intensity of the dominant MTR orientation for the entire microstructure is also provided in Figure 6.10.
suggesting that the MTR feature in the 10,000 – 12,000 um location is significant from an ultrasound scattering standpoint.

Figure 6.9: Identified key MTR regions in titanium microstructure and associated pole figures.

<table>
<thead>
<tr>
<th>Euler Angle 1</th>
<th>Euler Angle 2</th>
<th>Euler Angle 3</th>
<th>maxPfLine</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCW</td>
<td>Largest mass on PF</td>
<td>Set to 0</td>
<td>from code</td>
</tr>
<tr>
<td>Negative RD to largest mass on PF (angular distance from center)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>275</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>500</td>
<td>275</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>1500</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>2500</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>3000</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>3500</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>4000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>4500</td>
<td>275</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>5000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>5500</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>6000</td>
<td>185</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>6500</td>
<td>170</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>7000</td>
<td>160</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>7500</td>
<td>165</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>8000</td>
<td>170</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>8500</td>
<td>170</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>9000</td>
<td>175</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>9500</td>
<td>200</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>10000</td>
<td>225</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>10500</td>
<td>230</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11000</td>
<td>230</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11500</td>
<td>230</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 6.3
Summary table of dominant orientations for all 49 segments in the MTR titanium microstructures.

<table>
<thead>
<tr>
<th>Position</th>
<th>Theta (°)</th>
<th>Phi (°)</th>
<th>Intensity</th>
</tr>
</thead>
<tbody>
<tr>
<td>12000</td>
<td>225</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>12500</td>
<td>225</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>13000</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>13500</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>14000</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>14500</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>15000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>15500</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>16000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>16500</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>17000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>17500</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>18000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>18500</td>
<td>265</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>19000</td>
<td>265</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>19500</td>
<td>265</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>20000</td>
<td>270</td>
<td>65</td>
<td>0</td>
</tr>
<tr>
<td>20500</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>21000</td>
<td>Undefined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21500</td>
<td>85</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>22000</td>
<td>260</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>22500</td>
<td>270</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>23000</td>
<td>285</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>23500</td>
<td>Undefined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24000</td>
<td>Undefined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>whole</td>
<td>220</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 6.10** Peak intensity of dominant orientation for segmented locations.
6.3.2) **MTR Titanium Model Results**

For the micro-textured titanium microstructure there were six variations of models run to study the backscatter and attenuation of ultrasonic waves and signals for each segmented model. Figures 6.11 and 6.12 show the peak-to-peak analysis of the average y-displacement of propagating elastic wave for each scan position. This analysis was done for pitch-catch thru-transmission and pulse-echo backscatter sensing models. Also, a standard deviation analysis of the signals was done for each model. The model results included ultrasonic frequencies of 1MHz, 5MHz, and 10MHz.

![Figure 6.11: Pulse-echo average signal peak-to-peak analysis and standard deviation for NDE sensing models for the MTR titanium microstructure.](image)

106
Figure 6.12: Pitch-catch average signal peak-to-peak analysis and standard deviation for NDE sensing models for the MTR titanium microstructure.
6.3.3) Discussion and Correlation

The pitch-catch model results show interesting trends that correlate with the basic microstructure analysis. When comparing the microstructure with the signal output, the 5 MHz signals show spatial variations that correlate with the geometric MTR features for the entire sample as shown in Figure 6.13. The 10 MHz pitch-catch models also show a strong correlation with the dominant orientation intensities. The signals amplitude, position and magnitude shows a sudden change in signal when the wave is propagating in the 10,000 – 12,000 um position, a region identified as a strong macro grain position (MTR #7) as depicted in Figure 6.14. Figure 6.15 shows a comparison of the 11,000 micron segmented microstructure position, along with a depiction of the three pole figures for the key MTR regions in that segment, and a peak amplitude plot of the y-displacement levels in the ultrasonic model. The ultrasound image results show a strong scattering of energy due to misorientation levels at the key MTR boundaries, which result in strong signal variations in that segmented region. The results again show the value of the ICME framework for combining detailed microstructure property information with advanced sensing models for enhancing microstructure quantification using NDE sensing methods.
Figure 6.13: Correlation of 5 MHZ signal Analysis with geometric angles of grains and signal
Figure 6.14: Correlation of 10 MHz signal analysis with peak-to-peak signal analysis and dominate orientation intensities.
Figure 6.15: Segment in the 11,000 um region with pole figures, wave propagation for 10MHz, and peak y-displacement amplitude levels.

6.4) 3D Sample Results

6.4.1) Synthetic 3D Microstructure

Analysis, observation and correlation was done comparing Max Y-displacements of wave with size feature of the grain structure. Figure 6.16 depicts the peak amplitude response for the opposite face of y-plane propagation direction, where distinct spatial patterns occur in the propagating wave field that correlate reasonably well with the grain sizes and grain shapes of the corresponding 2D grain maps. In general, the shapes and sizes of the ultrasound patterns are within 10-20% of the actual grain sizes. Additional studies are underway to understand how the patterns change with $k_0d$ variations.
Figure 6.16: a. The 2D surface on the opposite side of the 3D grain models. b. Model Y-displacement showing lobbing of energy correlation to grain sizes. c. Grain size distribution for sample.

6.4.2) In100 3D Model

The longitudinal wavelength was again chosen to allow for stochastic regime scattering with $k_0d \sim 1$. The model results in Figure 16.7 depict one of the many unique opportunities that are now made available by the proposed methodology, where wave propagating features within the interior can be explored in detail (Figure 16.7b) to better understand complex polycrystalline scattering processes in a realistic 3 dimensional polycrystalline samples. Figure 6.17 depicts the peak amplitude response for the opposite face of y-plane, interior section propagation direction, where distinct spatial patterns occur in the propagating wave field that correlate reasonably well with the grain sizes and grain shapes of the corresponding 2D grain maps.
**Figure 6.17** a. 3D model representation b. 2D Midplane of sample. c. The 2D surface on the opposite side of the 3D grain models d. Model Y-displacement showing lobbing of energy correlation to grain sizes. e. Grain Size Distribution for Sample.
CHAPTER 7

SUMMARY AND RECOMMENDATIONS

7.1) Summary

The overall objective of research was to develop and demonstrate capabilities for characterizing and mapping 2D and 3D complex material systems. The thesis had three defined research specific tasks during the scope of the project. The first task of the research was to develop a Computational Nondestructive Evaluation (CNDE) framework that builds on the recently described concept of Integrated Computational Materials Engineering (ICME) [3 5 6]. The second task was to use and evaluate the new framework to systematically study ultrasonic wave interactions and scattering processes with two representative turbine engine microstructures – a dual-microstructure LSHR nickel and a micro-textured region (MTR) titanium. The second objective also did a proof-of-concept study of 3-dimensional synthetic and realistic microstructure evaluations and NDE sensing models for nickel super alloy materials. The third task was to try to correlate microstructure features with NDE sensing model indications, where a segmented material approach was used to approximate a scanning ultrasound probe sensing method.

Listed below are highlighted accomplishments of thesis research:
• An ICME framework was successfully developed that could take key microstructure information and input it directly into NDE sensing models, where material property information could be accounted for on an element-by-element basis in the model.

• The ICME framework was successfully used to study virtual representations of realistic 2-dimensional and 3-dimensional polycrystalline microstructures as well as a synthetic 3-dimensional microstructure representations.

• The capabilities and limitations of the current ICME framework approach were identified to include: capabilities to accurately provide element-by-element stiffness matrix values into an NDE sensing model for centimeter-sized 2-dimensional sizes, and 100-micron sized 3-dimensional sample sizes. The current framework is tedious and time-consuming to apply directly to larger 3-dimensional microstructure volumes, but the feasibility of the approach was shown in the thesis work.

• A reasonable correlation of sensing model signal results and microstructure statistics was observed for both the LSHR dual-microstructure nickel and MTR titanium materials. The ICME framework suggests much more sophisticated computational model-driven studies can be done to understand ultrasound scattering processes in a variety of material systems.

7.2) Future Work and Recommendations

Future work to research should include optimizing the ICME framework to minimize the number of steps, time, and software tools used. Also recommended would be to evaluate more complex microstructures and realistic 3-dimensional samples. Future model should incorporate larger microstructures and conditions that also model the ultrasound sensor probe, and additional realistic complexities to measurements. And lastly, further investigation and analysis
of the correlation method should be accomplished to directly compare the ground-truth microstructure statistics to the key NDE sensing physics.


[26] EDAX. Crystallographic orientation [PowerPoint slides].PDF.


[31] P. Nagy personal communication.

[32] Lovelady, K. Anisotropic Materials. PDF. Univ. of Texas at San Antonio.


APPENDIX A

DOCUMENTATION OF PROCESSES

Contents of Appendix A

A1: TSL_2_DREAM3D Pipeline Process

A2: Synthetic Generation Pipeline Process

A3: Matlab Calculation and Reconfiguration Process

A4: Consolidation of PZFlex Files Process

A1: TSL_2_DREAM3D Pipeline Process

Complete Steps going from *.ang files to DREAM3D ph & csv files

TSL_2_DREAM3D_1 pipeline 1

This pipeline takes the ang file(s) and converts them to a H5Ebsd file

- Set input directory
  - Folder where ang file are located
- Set output of h5ebsd
  - If there is more than one ang file make sure to set prefix suffix, number of files and padding digits
  - In the case of more than 1 ang file need to have some types of sequential number mechanism in order to be recognized by DREAM3D
- Run pipeline
- Below is the setup of pipeline one
TSL_2_DREAM3D_2 pipeline 2

Pipeline 2 takes the H5Ebsd file from pipeline 1 and goes through a process to calculates field data in order to find Euler angles of the grains and outputs data as a dream 3d file and also and converts it to a PH and CSV file to be used in matlab codes.

- Read in H5EBSD file
  - Input the name of particular file
  - Check cell data, field data and ensemble data boxes in this filter
- Generate name for output dream3d file and directory
- Designate an output name and directory for the ph file
- Designate an output name and directory for the csv file
- Run pipeline
- Below is a setup of pipeline 2
TSL_2_DREAM3D_3_optional pipeline 3(optional)

This pipeline takes the dream3d file for pipeline 3 and has the ability to crop the structure to a different dimension and output the PH and CSV file for particular cropped region
- Select dream3d file from pipeline 3 as input file for this pipeline line
  - Check all boxes
- Crop the dimension in the x, y, and z fields as desires
  - Uncheck renumber grains
- Set up output dream 3d file
- Runs pipeline
- Below is setup for pipeline 4

---

**A2: Synthetic Generation Pipeline Process**

Complete Steps Going From Dream 3D Synthetic Generation To .ph And .csv Inputs To Matlab

**STATS GENERATOR**
The stats generator application in the DREAM3D Package has the ability to create a synthetic data set of static based on input preferences of distribution, shape, ODF and much more

a. Open stats generator executable in dream 3d file directory
b. Grain size distribution
   i. Input values for mu, sigma, min sigma cut off value, max sigma cut off value, bin step size (e.g. 30 micron mean and 10 micron distribution example below)
   ii. Equiaxed or rolled? (e.g. equiaxed)
   iii. Press the create data button (very important or data will not be created)
   iv. Save with .dream3d file extension (e.g. dream3D2PZFlez.dream3d)

DREAM 3D

DREAM3D has the ability to analyze, reconstruct, sample, and calculate statistical data of synthetic and realistic microstructures through the filters and pipelines constructed through the software’s platform

a. SYNTHETIC GENERATION 1 pipeline
i. Designate Stats generator file as input file

![Image of Initialize Synthetic Volume](image1)

ii. Select desired voxel dimensions and spacing

![Image of Voxel Dims and Spacing](image2)

iii. Create output file and run pipeline

![Image of Initialize Synthetic Volume](image3)

iv. Can view sample in paraview through .xdmf file
   (e.g. `generating_synthetic_output.xdmf`)
   1. Go to FILE → OPEN→.xdmf file (e.g. `generating_synthetic_output.xdmf`)
   2. On side properties tool bar check all boxes then press apply
3. Click on threshold command on top tool box

4. On side tool bar, in scalars box threshold by grain ID’s and click apply

5. Go up to top tool bar an scroll down to display grain id’s
b. **SYTHETIC GENERATION 2 Pipeline**

i. Read in “generating synthetic 1” pipeline output file as input to this pipeline

ii. Check all the boxes in voxel data area of the read in filter

iii. Crop 3d volume to specific layer

   1. (e.g. x:1-199 y:1-199 z: cropped to layer 7)

iv. Depending on goal check/uncheck renumber grain box

v. Designate name for output file and run code (e.g. cropping3dfile_output.dream3d)

vi. Create output files for CSV, PH,

   1. (e.g. PH_output.ph, CSV_output.csv)

vii. Run Pipeline

viii. Can view sample in paraview through .xdmf file

   (e.g. generating_synthetic_output_2xdmf)
MATLAB A: Calculation of Crystallographic orientation

The first set of codes takes the PH and CSV files attained from the DEAM3D pipeline and takes the Euler angles of each grain id and calculate crystallographic stiffness matrix based on orientation of the grain from the reference material

DREAM3D2PZLEX M-FILE

- Make sure that all m-files are in the same directory of code will not work properly
  - Main M-File – DREAM3D2PZLEX or DREAM3D2PZLEX_loop
  - Attached M-Files – RMatOFBunge, RotateCijklReal, RotateY, Cij2Cijkl, Cijkl2Cij, CijNickel.mat, CijklTi64.mat
  - Use approach A (step a) for 2d and approach B (step b) for 3D

a. Approach A (2D): In Main M-File DREAM3D2PZLEX:
   i. Text read in file directory of PH and CSV files of the data sample

```matlab
phfile = textread('C:\Users\darius\Desktop\Dream3D to PZLEX\step 2\ph_output.ph',',',1,'headerlines',3);
phfilename = 'C:\Users\darius\Desktop\Dream3D to PZLEX\step 2\ph_output.csv',',',1,'headerlines',3);
```

ii. Specify the File out name and directory

```matlab
phfile = textread('C:\Users\darius\Desktop\Dream3D to PZLEX\step 2\ph_output.ph',',',1,'headerlines',3);
phfilename = 'C:\Users\darius\Desktop\Dream3D to PZLEX\step 2\ph_output.csv',',',1,'headerlines',3);
```

iii. continue at step c

b. Approach B: (3D) In Main M-File DREAM3D2PZLEX_loop:
   i. **** this version of the main file goes through a looping process to batch run serial sections of a 3d volume****
   ii. Specify how many layers of in the 3D sample (e.g. a = 0:1:116...116 layers)
   iii. Write the file root of CSV and PH file so the code can extract them for each specific layer. (NOTE: all file should be in the same directory for code to work properly)
   iv. Input the file root directory for all output files
v. Continue at step c

c. Continuing in either – DREAM3D2PZLEX or DREAM3D2PZLEX_loop
  i. Specify Number of voxels in X and Y direction and resolution

\[
\begin{align*}
\text{nx} &= 199; \\
\text{ny} &= 199; \\
\text{res} &= 1; \ % \text{resolution in microns}
\end{align*}
\]

ii. Load the reference material 6x6 material. (usually nickel or titanium)

```matlab
load CijNickel; 
Cij = CijNickel; 
load CijTi64; 
Cij = CijTi64; 
```

iii. In the code where it says \( \text{eul} = \text{grainfile}(i,7:9) \)' 7:9 are the columns in the CSV file which contain the Euler angles, make sure to specify the right columns for particular data set.

```
for i = 1:size(grainfile,1) 
  % get current grain id (gid) 
  gid = grainfile(i,1); 
  % eul = BungeToDMat(BMatOfQuat(grainfile(i,4:7))','rad') 
  eul = grainfile(i,7:9); 
end 
```

iv. Run Code
  i. If ran correctly code will formulate a mapping of the grain structure as such:
MATLAB B: RECONFIGURING DATA

In reconfiguring the data two files need to be generated to represent the microstructure in the FEM model. The first file is one that describes the spatial coordinates of each voxel in the microstructure volume space, the second file accounts for the $C_{ij}$ stiffness matrix unique to each grain ID of the sample. There is a set of codes to reconfigure the data for 2D data set and a set of codes for 3D data sets. Also there is specialized code to initialize a 3d volume of...
a 2d layer as well as clustering of similar voxels together for the purpose of memory of the computer.

**2 Dimensional sample**

*Cij* _Spatial Coordinates_ 2D

Takes output file from the DREAM3D2PZFLEX matlab code and reconfigure the data in the form recognized by pzflex that accounts for material identification designation for all spatially resolve elements in the material model. the spatially resold elements are identified by position in the x, y z coordinates and l explicitly written as x begin position, x end position, y begin position and y end position

- Input file and directory is the output file from Adam’s code

```
#define input and output file
/**changes for every new data set**
#include <iostream>
#include <fstream>

outFile = textread(['C:\Users\darius\Desktop\Dream3D to PZFLEX\step 3\layer 1\out',',','-1',',',',1',',]',];
```

- Specify file name and directory for the output file

```
fileWriteID = fopen(['C:\Users\darius\Desktop\Dream3D to PZFLEX\step 4\pzflex_input_regm_ij.txt',',',',w',']);
```

- Run Code

  - Code Output

```
regm mat169 1. 2. 1. 2.
regm mat169 2. 3. 1. 2.
regm mat169 3. 4. 1. 2.
regm mat169 4. 5. 1. 2.
regm mat169 5. 6. 1. 2.
regm mat169 6. 7. 1. 2.
regm mat169 7. 8. 1. 2.
regm mat169 8. 9. 1. 2.
regm mat169 9. 10. 1. 2.
regm mat169 10. 11. 1. 2.
regm mat169 11. 12. 1. 2.
regm mat169 12. 13. 1. 2.
regm mat169 14. 15. 1. 2.
regm mat169 15. 16. 1. 2.
regm mat169 16. 17. 1. 2.
regm mat169 17. 18. 1. 2.
regm mat169 18. 19. 1. 2.
```

*Cij* _Stiffness matrix_ IDs

Takes Dream3d2pzflex code output file and reconfigures the data so that each grain identifier has attached its rotated 21 elastic stiffness matrix values that are dependent on grain orientation. Also the computation of this code writes out the density of the material.

- Input file and directory is the output file from Adam’s code

```
#define input and output file
/**changes for every new data set**
#include <iostream>
#include <fstream>

outFile = textread(['C:\Users\darius\Desktop\Dream3D to PZFLEX\step 3\layer 2\out',',','-1',',',',1',',]',];
```

```
• Specify file name and directory for the output file

    ```
    fwriteID = fopen('C:\Users\darius\Desktop\Dream3D to PZFLEX\step 4\prop_function.txt','w')
    ```

• Specify density of material

    ```
    raw_grain_number = outfile(:,3);
    stiffness_matrix_values = outfile(:,1:24);
    density = 7500; %*** changes for a given material***
    ```

• Run code

    o Code output

    ```
    3 Dimensional sample

    Cij_Spatial_Coordinates_3D_brut_force_noncluster

    • Brute force way to make layers for 3D sample with vary layers through the depth of the sample
    • Input file and directory is the output file from DREAM3D2PZFLEX code
    • Input file changes for each particular layer that data need reconfiguring
    • Specify file name and directory for the output file
    • Specify Z_begin
    • Z_begin correlates to the layer depth
    • Run Code

    clear all
clc

    %define input and output file
    ...         outfile = textread('C:\Users\JohnoDRI\Desktop\20microngrains data'..'.1\headerlines.1');
    ...         fwriteID = fopen('layer_1.txt','w');

    %output file
    outfile = textread('C:\FILE ROOT out'..'.1\headerlines.1');
    fwriteID = fopen('layer_1.txt','w');

    %z-layer
    Z_begin = 1;
    Z_end = Z_begin + 1;
```
**consolidate_CijSpatialCoordinates_files**

- Strings together all the files outputs from "PzFlex_input_regn_3D_Brut_force" code
- File output names must be correct in order for model to run
- Run Code

```%
% consolidate_CijSpatialCoordinates_files
% this code strings together files produced by "Cij_Spatial_coordinates_3D_Brut_forc_noncluster"
% in an effort to consolidate individual layers into a 3Dimensional
% sample
%
% Authors: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014
%
% 50 layers
p = system('copy layer_1 bt = layer_2 bt = layer_3 bt = layer_4 bt = layer_5 bt = layer_6 bt = layer_7 bt = layer_8 bt = layer_9 bt = layer_10 bt ConsolidatedFile bt')
%
% 100 layer consolidation
%p = system('copy layer_1 bt = layer_2 bt = layer_3 bt = layer_4 bt = layer_5 bt = layer_6 bt = layer_7 bt = layer_8 bt = layer_9 bt = layer_10 bt layer_11 bt = layer_12 bt = layer_13 bt = layer_14 bt = layer_15 bt = layer_16 bt = layer_17 bt = layer_18 bt = layer_19 bt = layer_20 bt = layer_21 bt = layer_22 bt = layer_23 bt = layer_24 bt = layer_25 bt = layer_26 bt = layer_27 bt = layer_28 bt = layer_29 bt = layer_30 bt = layer_31 bt = layer_32 bt = layer_33 bt = layer_34 bt = layer_35 bt = layer_36 bt = layer_37 bt = layer_38 bt = layer_39 bt = layer_40 bt = layer_41 bt = layer_42 bt = layer_43 bt = layer_44 bt = layer_45 bt = layer_46 bt = layer_47 bt = layer_48 bt = layer_49 bt = layer_50 bt = layer_51 bt = layer_52 bt = layer_53 bt = layer_54 bt = layer_55 bt = layer_56 bt = layer_57 bt = layer_58 bt = layer_59 bt = layer_60 bt = layer_61 bt = layer_62 bt = layer_63 bt = layer_64 bt = layer_65 bt = layer_66 bt = layer_67 bt = layer_68 bt = layer_69 bt = layer_70 bt = layer_71 bt = layer_72 bt = layer_73 bt = layer_74 bt = layer_75 bt = layer_76 bt = layer_77 bt = layer_78 bt = layer_79 bt = layer_80 bt = layer_81 bt = layer_82 bt = layer_83 bt = layer_84 bt = layer_85 bt = layer_86 bt = layer_87 bt = layer_88 bt = layer_89 bt = layer_90 bt = layer_91 bt = layer_92 bt = layer_93 bt = layer_94 bt = layer_95 bt = layer_96 bt = layer_97 bt = layer_98 bt = layer_99 bt = layer_100 bt ConsolidatedFile bt')
```

**Stiffness_Matrix_IDS_3D**

Takes DREAM3D2PZFLEX code output file and reconfigures calculated data in to a format that identifies rotated stiffness matrix values for all grain ids and density of the material

- Input file and directory is the output file from DREAM3D2PZFLEX code
- Specify file name and directory for the output file
- Specify density of material
- Run code
- Code output
A4: Consolidation of PZFlex Files Process

**This code takes Pzflex files, neglects the time in each file and consolidates them into one .TXT document**

1. In the F:\ drive open the “Consolidate PzFlex files” folder
   a. Within that folder contains the matlab code, CONSOLIDATE_1D_PZFLEX_DATA, and sample data folder “Large grain pitch-catch 20”
2. Open the code in matlab
   a. Make sure the “Consolidate PzFlex files” folder is the current directory
3. Run code
4. A Select Data File box will appear
   a. Select your data (e.g. largeGrain pitch-catch 20 folder)

5. Click on folder to display contents
   a. Click on x-disp folder to display bscan data
   b. Use shift to select multiple files at a time (e.g. HighRes1-HighRes45)
6. Code will run and consolidate files  
   a. Depending on size of data may take a few minutes  
7. Save output data box will appear  
   a. Save TXT file to desired place (e.g. save as, sample.txt, saved in directory folder)
8. Open saved data and information should appear as such below:

**** larger consolidation files will not look as uniform and may be better to work with in excel as tab and space delimited data****
APPENDIX B

DREAM.3D PIPELINES

Contents of Appendix B

B1: TSL_2_DREAM3D Pipelines

B2: Synthetic Generation Pipelines

B1: TSL_2_DREAM3D Pipelines

TSL_2_DREAM3D PIPELINE 1

[PipelineBuilder]
Number_Filters=1
Name=streamline_pipeline_1

[0]
Filter_Name=EbsdToH5Ebsd
InputDir=
FilePrefix=Slice_
FileSuffix=
FileExt=ang
ZStartIndex=1
ZEndIndex=99
zSpacing=0.25
TotalDigits=4
OutputFile=Untitled.h5ebsd
StackHighToLow=false
StackLowToHigh=true
TSLchecked=false
HKLchecked=false
HEDMchecked=false
NoTranschecked=true

TSL_2_DREAM3D PIPELINE 2

[PipelineBuilder]
Number_Filters=10
Name=streamline_pipeline_2

[0]
Filter_Name=ReadH5Ebsd
InputFile=C:\\Users\\darius\\Desktop\\F110 TSL2PZFlex\\001 pipeline 1 (h5ebsd file)\\1000_8000.h5ebsd
ZStartIndex=8000
ZEndIndex=8000
UseTransformations=true
ArraySelections_VoxelCell\size=6
ArraySelections_VoxelCell\1\VoxelCell=Confidence Index
ArraySelections_VoxelCell\2\VoxelCell=EulerAngles
ArraySelections_VoxelCell\3\VoxelCell=Fit
ArraySelections_VoxelCell\4\VoxelCell=Image Quality
ArraySelections_VoxelCell\5\VoxelCell=Phases
ArraySelections_VoxelCell\6\VoxelCell=SEM Signal
ArraySelections_VoxelField\size=0
ArraySelections_VoxelEnsemble\size=3
ArraySelections_VoxelEnsemble\1\VoxelEnsemble=CrystalStructures
ArraySelections_VoxelEnsemble\2\VoxelEnsemble=LatticeConstants
ArraySelections_VoxelEnsemble\3\VoxelEnsemble=MaterialName
ArraySelections_SurfaceMeshPoint\size=0
ArraySelections_SurfaceMeshFace\size=0
ArraySelections_SurfaceMeshEdge\size=0
ArraySelections_SurfaceMeshField\size=0
ArraySelections_SurfaceMeshEnsemble\size=0
ArraySelections_SolidMeshPoint\size=0
ArraySelections_SolidMeshFace\size=0
ArraySelections_SolidMeshEdge\size=0
ArraySelections_SolidMeshField\size=0
ArraySelections_SolidMeshEnsemble\size=0

[1]
Filter_Name=FindCellQuats

[2]
Filter_Name=EBSDSegmentGrains
MisorientationTolerance=5

[3]
Filter_Name=GenerateIPFColors
ReferenceDir\size=3
ReferenceDir\1\x=0
ReferenceDir\2\y=0
ReferenceDir\3\z=1

[4]
Filter_Name=FindSizes

[5]
Filter_Name=FindNeighbors

[6]
Filter_Name=FindAvg Orientations

[7]
Filter_Name=DataContainerWriter
OutputFile=C:\\Users\\darius\\Desktop\\streamline test\\test.dream3d
WriteVoxelData=true
WriteSurfaceMeshData=true
WriteXdmfFile=true

[8]
Filter_Name=FieldDataCSVWriter
FieldDataFile=C:\\Users\\darius\\Desktop\\streamline test\\test.csv
WriteNeighborListData=false

[9]
Filter_Name=PhWriter
OutputFile=C:\\Users\\darius\\Desktop\\streamline test\\test.ph

TSL_2_DREAM3D PIPELINE 3 Optional

[PipelineBuilder]
Number_Filters=5
Name=TSL_2_DREAM3D_3

[0]
Filter_Name=DataContainerReader
InputFile=C:\\Users\\johnsodr\\Desktop\\001 original size 2516x888\\002\\REGION 1 REDO_Mod.dream3d
ReadVoxelData=true
ReadSurfaceMeshData=false
ReadSolidMeshData=false
ArraySelections_VoxelCell\size=11
ArraySelections_VoxelCell\1\VoxelCell=Confidence Index
ArraySelections_VoxelCell\2\VoxelCell=EulerAngles
ArraySelections_VoxelCell\3\VoxelCell=Fit
ArraySelections_VoxelCell\4\VoxelCell=Good Voxels
ArraySelections_VoxelCell\5\VoxelCell=GrainIds
ArraySelections_VoxelCell\6\VoxelCell=IPF Color
ArraySelections_VoxelCell\7\VoxelCell=Image Quality
ArraySelections_VoxelCell\8\VoxelCell=Phases
ArraySelections_VoxelCell\9\VoxelCell=Quats
ArraySelections_VoxelCell\10\VoxelCell=SEM Signal
ArraySelections_VoxelCell\11\VoxelCell=Surface Voxels
ArraySelections_VoxelField\size=10
ArraySelections_VoxelField\1\VoxelField=Active
ArraySelections_VoxelField\2\VoxelField=AvgQuats
ArraySelections_VoxelField\3\VoxelField=Equivalent Diameters
ArraySelections_VoxelField\4\VoxelField=EulerAngles
ArraySelections_VoxelField\5\VoxelField=Neighbor List
ArraySelections_VoxelField\6\VoxelField=Num Cells
ArraySelections_VoxelField\7\VoxelField=Num Neighbors
ArraySelections_VoxelField\8\VoxelField=Shared Surface Area List
ArraySelections_VoxelField\9\VoxelField=Surface Fields
ArraySelections_VoxelField\10\VoxelField=Volumes
ArraySelections_VoxelEnsemble\size=3
ArraySelections_VoxelEnsemble\1\VoxelEnsemble=Crystal Structures
ArraySelections_VoxelEnsemble\2\VoxelEnsemble=Lattice Constants
ArraySelections_VoxelEnsemble\3\VoxelEnsemble=Material Name
ArraySelections_SurfaceMeshPoint\size=0
ArraySelections_SurfaceMeshFace\size=0
ArraySelections_SurfaceMeshEdge\size=0
ArraySelections_SurfaceMeshField\size=0
ArraySelections_SurfaceMeshEnsemble\size=0
ArraySelections_SolidMeshPoint\size=0
ArraySelections_SolidMeshFace\size=0
ArraySelections_SolidMeshEnsemble\size=0

[1]
Filter_Name=CropVolume
XMin=0
YMin=0
ZMin=0
XMax=0
YMax=0
ZMax=0
Renumber Grains=true
Update Origin=true
B2: Synthetic Generation Pipelines

Synthetic Generation Pipeline 1

[PipelineBuilder]
Number_Filters=7
Name=Synthetic Generation 1

[0]
Filter_Name=InitializeSyntheticVolume
InputFile=
XResolution=1
YResolution=1
ZResolution=1
XPoints=200
YPoints=200
ZPoints=100

[1]
Filter_Name=PackPrimaryPhases
PeriodicBoundaries=true
WriteGoalAttributes=false
CsvOutputFile=
[2]
Filter_Name=FindNeighbors

[3]
Filter_Name=FindNumFields

[4]
Filter_Name=MatchCrystallography
MaxIterations=100000

[5]
Filter_Name=GenerateIPFColors
ReferenceDir\size=3
ReferenceDir\1\x=0
ReferenceDir\2\y=0
ReferenceDir\3\z=1

[6]
Filter_Name=DataContainerWriter
OutputFile=
WriteVoxelData=true
WriteSurfaceMeshData=false
WriteXdmfFile=true

Synthetic Generation Pipeline 2

[PipelineBuilder]
Number_Filters=5
Name=Synthetic Generation 2

[0]
Filter_Name=DataContainerReader
InputFile=
ReadVoxelData=true
ReadSurfaceMeshData=false
ReadSolidMeshData=false
ArraySelections_VoxelCell\size=0
ArraySelections_VoxelField\size=0
ArraySelections_VoxelEnsemble\size=0
ArraySelections_SurfaceMeshPoint\size=0
ArraySelections_SurfaceMeshFace\size=0
ArraySelections_SurfaceMeshEdge\size=0
ArraySelections_SurfaceMeshField\size=0
ArraySelections_SurfaceMeshEnsemble\size=0
ArraySelections_SolidMeshPoint\size=0
ArraySelections_SolidMeshFace\size=0
ArraySelections_SolidMeshEnsemble\size=0

[1]
Filter_Name=CropVolume
XMin=1
YMin=1
ZMin=7
XMax=199
YMax=199
ZMax=7
RenumberGrains=true
UpdateOrigin=true

[2]
Filter_Name=DataContainerWriter
OutputFile=
WriteVoxelData=true
WriteSurfaceMeshData=true
WriteXdmfFile=true

[3]
Filter_Name=PhWriter
OutputFile=

[4]
Filter_Name=FieldDataCSVWriter
FieldDataFile=
WriteNeighborListData=false
Contents of Appendix C

C1: Crystallographic Orientation: Elastic Stiffness Matrix Codes
C2: Data Reconfiguration Codes – 2 Dimensional
C3: Data Reconfiguration Codes – 3 Dimensional
C4: Extrapolation of dominant orientation from Pole figures Code
C5: Consolidation of PZFLEX output Data files

C1: Crystallographic Orientation: Elastic Stiffness Matrix Codes

DREAM3D2PZFLEX M-File (Main File)

% DREAM3D2PZFLEX
... DREAM3D2PZFLEX converts dream3d output files .ph and .csv to an input
... file for the FEM software PZFLEX

% Author: Adam Pilchak, AFRL/RXCM, 18 June 2013

% phfile = textread('C:\FILE ROOT.ph','','-1',''headerlines',3);
% grainfile = textread('C:\FILE ROOT.csv','','-1',''delimiter',',''headerlines',2);
% foutname = 'C:\FILE ROOT.out';

phfile = textread('G:\0009 F110 model and data\0004 segmented files\2000.ph','','-1',''headerlines',3);
grainfile = textread('G:\0009 F110 model and data\0004 segmented files\2000.csv','','-1',''delimiter',',''headerlines',2);
foutname = 'G:\0009 F110 model and data\0004 segmented files\2000.out';
% symm = 'FCC'; % crystal symmetry, a string 'HCP' or 'BCC' describing the crystal symmetry.
   % Note that 'BCC' is also appropriate for 'FCC' crystals but is called 'BCC' for historical
   % reasons in the code.
   % FCC for Ni
% ti = hcp
% ni =

% number of voxels in x and y and the resolution.
 nx = 101;
 ny = 851;
 res = 10; % resolution in microns
 %
 Cij = zeros(6,6); % preallocate an array for the 6x6 elastic stiffness tensor in matrix form

% load a .mat file containing the [6x6] matrix for nickel, or enter the matrix here manually in GPa
% load CijNickel;
% Cij = CijNickel;
load CijklTi64;
Cij = CijklTi64;

%% create the grid of spatial coordinates
 xS = 1:nx; yS = 1:ny;
 xS = repmat(xS, [ny 1])';
 yS = repmat(yS, [nx 1]);
 coords = [xS(:) yS(:)];  % for all x's, go y.
 %
%% rifle through the ph file, find associated grains in csv and apply to AngFile
 %
% reshape ph file and display as an image
 phfile = transpose(phfile);
 phmap = reshape(phfile,[ nx ny]);
 pcolor(phmap); shading flat; axis equal; axis ij; axis tight;
 %
% convert from matrix to tensor form
 Cijkl = Cij2Cijkl(Cij);
 out = zeros(nx*ny,24); % set up the output array

 for i = 1:size(grainfile,1)
   % get current grain id (gid)
   gid = grainfile(i,1);
   % eul = BungeOfRMat(RMatOfQuat(grainfile(i,4:7)),'radians');
   % eul = grainfile(i,8:10)';
   % find these locations in the phfile
   [r,c] = find(phmap == gid);
   ind = sub2ind(size(phmap),r,c);
   %
   % convert matrix to tensor notation, rotate by the Euler angle and store
   % the 21 independent components. Move back and forth between matrix /
   % vector notation for convenience
   RotatedCijkl = RotateCijklReal(Cijkl,eul,'radians');
RotatedCij = 1E9*Cijkl2Cij(RotatedCijkl);

lowerHalf = [RotatedCij(1,1) RotatedCij(1,2) RotatedCij(1,3) RotatedCij(1,4) RotatedCij(1,5) RotatedCij(1,6),...
   RotatedCij(2,2) RotatedCij(2,3) RotatedCij(2,4) RotatedCij(2,5) RotatedCij(2,6),...
   RotatedCij(3,3) RotatedCij(3,4) RotatedCij(3,5) RotatedCij(3,6),...
   RotatedCij(4,4) RotatedCij(4,5) RotatedCij(4,6),...
   RotatedCij(5,5) RotatedCij(5,6),...
   RotatedCij(6,6)]';

% lowerHalf = reshape(tril(RotatedCij,0),[36,1]);
% lowerHalf = lowerHalf(reshape(lowerHalf,[36 1])~=0);

% compile an output array
% x-position   y-position   StiffnessNumber  c11   c12  c13 c14 c15 c16, c22, c23, c24, c25, c26, c33,.....c66
out(ind,1:24) = [coords(ind,1) coords(ind,2) repmat(i,size(coords(ind,1))) repmat(
   transpose(lowerHalf), size(coords(ind,1))) ];

end
%

% write ANG file
fileWriteID = fopen(foutname,'w');
%
% set up a format string with decimal places similar to ANG files
format_Out = '%i %i %i %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e \n';
% write the data to the file
fprintf(fileWriteID,'%x-position   y-position   StiffnessNumber  c11, c12, c13, c14, c15, c16, c22, c23, c24, c25, c26, c33,.....c66
%fprint(fileWriteID, format_Out, out');
%
%fclose all;

---------------------------------------------------------------

DREAM3D2PZFLEX_LOOP M-File (Main File)

% DREAM3D2PZFLEX
... DREAM3D2PZFLEX converts dream3d output files .ph and .csv to an input -
... looping process to convert files for ech layer of a 3D specimen

% file for the FEM software PZFLEX

% Author: Adam Pilchak, AFRL/RXCM, 18 June 2013
% Modified By: Darius, AFRL/RXCA, University of Dayton, 7 June 2014

% example
%PH = ['C:\Users\dariusj\Desktop\large grain ph csv\layer_' +num2str(a)+ '.ph'];

149
%CSV = ['C:\Users\dariusj\Desktop\large grain ph csv\layer_' + num2str(a) + '.csv'];

clc
clear all
for a = 0:500:24000
%PH = ['C:\Users\dariusj\Desktop\large grain ph csv\layer_' + num2str(a) + '.ph'];
%CSV = ['C:\Users\dariusj\Desktop\large grain ph csv\layer_' + num2str(a) + '.csv'];

filename = ['C:\FILE ROOT\layer_' num2str(a) '.ph'];
filename = ['G:\0009 F110 model and data\0004 segmented files\' num2str(a) '.ph'];

filename_2 = ['C:\FILE ROOT\layer_' num2str(a) '.csv'];
filename_2 = ['G:\0009 F110 model and data\0004 segmented files\' num2str(a) '.csv'];

phfile = textread(filename,'',-1,'headerlines',3);
grainfile = textread(filename_2,'',-1,'delimiter','','headerlines',2);

foutname = ['C:\FILE ROOT\layer_' num2str(a) '.out'];
phfile = textread(filename,'',-1,'headerlines',3);
grainfile = textread(filename_2,'',-1,'delimiter','','headerlines',2);

foutname = ['G:\0009 F110 model and data\0004 segmented files\' num2str(a) '.out'];

symm = 'FCC';  % crystal symmetry, a string 'HCP' or 'BCC' describing the crystal symmetry.
% Note that 'BCC' is also appropriate for 'FCC' crystals but is called 'BCC' for historical
% reasons in the code.

% number of voxels in x and y and the resolution.
nx = 101;
yy = 851;
res = 10;  % resolution in microns

Cij = zeros(6,6);  % preallocate an array for the 6x6 elastic stiffness tensor in matrix form

% load a .mat file containing the [6x6] matrix for nickel, or enter the matrix here manually in GPa

%load CijNickel;
%Cij = CijNickel;

load CijklTi64;
Cij = CijklTi64;

% create the grid of spatial coordinates
xS = 1:nx; yS = 1:ny;
xS = repmat(xS, [ny 1])';
yS = repmat(yS, [nx 1]);
coords = [xS(:) yS(:)];  %for all x's, go y.
%
% rifle through the ph file, find associated grains in csv and apply to AngFile
% reshape ph file and display as an image
phfile = transpose(phfile);
phmap = reshape(phfile,[ nx ny]);
pcolor(phmap); shading flat; axis equal; axis ij; axis tight;
%
% convert from matrix to tensor form
Cijkl = Cij2Cijkl(Cij);
out = zeros(nx*ny,24); % set up the output array

for i = 1:size(grainfile,1)
% get current grain id (gid)
gid = grainfile(i,1);
% eul = BungeOfRMat(RMatOfQuat(grainfile(i,4:7)),'radians');
eul = grainfile(i,8:10)';
% find these locations in the phfile
[r,c] = find(phmap == gid);
ind = sub2ind(size(phmap),r,c);
%
% convert matrix to tensor notation, rotate by the Euler angle and store
% the 21 independent components. Move back and forth between matrix /
% vector notation for convenience
RotatedCijkl = RotateCijklReal(Cijkl,eul,'radians');
RotatedCij   = 1E9*Cijkl2Cij(RotatedCijkl);
lowerHalf = [RotatedCij(1,1) RotatedCij(1,2) RotatedCij(1,3) RotatedCij(1,4) RotatedCij(1,5)
RotatedCij(1,6),...
   RotatedCij(2,2) RotatedCij(2,3) RotatedCij(2,4) RotatedCij(2,5) RotatedCij(2,6),....
   RotatedCij(3,3) RotatedCij(3,4) RotatedCij(3,5) RotatedCij(3,6),....
   RotatedCij(4,4) RotatedCij(4,5) RotatedCij(4,6),....
   RotatedCij(5,5) RotatedCij(5,6),....
   RotatedCij(6,6)]';
% lowerHalf = reshape(tril(RotatedCij,0),[36,1]);
% lowerHalf = lowerHalf(reshape(lowerHalf,[36 1])~=0);

% compile an output array
% x-position   y-position   StiffnessNumber  c11   c12  c13 c14 c15 c16, c22, c23, c24, c25, c26,
c33,.....c66
out(ind,1:24) = [coords(ind,1) coords(ind,2)  repmat(i,size(coords(ind,1)))  repmat(
transpose(lowerHalf),  size(coords(ind,1)))    ];
end
%
% write ANG file
fileWriteID = fopen(foutname,'w');
%
% set up a format string with decimal places similar to ANG files
format_Out = '%i %i %i %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e %e

151
fprintf(fileWriteID, format_Out, out');

fclose all;
end

function [Cijkl] = Cij2Cijkl(Cij)

% Cijkl = Cij2Cijkl(Cij) converts Cij to Cijkl
% % Input:
% %   (1) Cij is the [6x6] stiffness tensor in the single crystal reference frame at g = {0,0,0}
% % Output:
% %   (1) Cijkl is the [3x3x3x3] array of modulii
% %

% preallocate Cijkl
Cijkl = zeros(3,3,3,3);
% the trace of the [6x6]
\begin{align*}
& C_{ijkl}(1,1,1,1) = C_{ij}(1,1); C_{ijkl}(2,2,2,2) = C_{ij}(2,2); \\
& C_{ijkl}(3,3,3,3) = C_{ij}(3,3); C_{ijkl}(2,3,2,3) = C_{ij}(4,4); \\
& C_{ijkl}(3,1,3,1) = C_{ij}(5,5); C_{ijkl}(1,2,1,2) = C_{ij}(6,6); \\
\end{align*}
%
% the first row and column of the [6x6]
\begin{align*}
& C_{ijkl}(2,2,1,1) = C_{ij}(2,1); C_{ijkl}(1,1,2,2) = C_{ij}(1,2); \\
& C_{ijkl}(3,3,1,1) = C_{ij}(3,1); C_{ijkl}(1,1,3,3) = C_{ij}(1,3); \\
& C_{ijkl}(2,3,1,1) = C_{ij}(4,1); C_{ijkl}(1,1,2,3) = C_{ij}(1,4); \\
& C_{ijkl}(3,1,1,1) = C_{ij}(5,1); C_{ijkl}(1,1,3,1) = C_{ij}(1,5); \\
& C_{ijkl}(1,2,1,1) = C_{ij}(6,1); C_{ijkl}(1,1,1,2) = C_{ij}(1,6); \\
\end{align*}
%
% the remaining part
\begin{align*}
& C_{ijkl}(3,3,2,2) = C_{ij}(3,2); C_{ijkl}(2,2,3,3) = C_{ij}(2,3); \\
& C_{ijkl}(2,3,2,2) = C_{ij}(4,2); C_{ijkl}(2,2,2,3) = C_{ij}(2,4); \\
& C_{ijkl}(3,1,2,2) = C_{ij}(5,2); C_{ijkl}(2,2,3,1) = C_{ij}(2,5); \\
& C_{ijkl}(1,2,2,2) = C_{ij}(6,2); C_{ijkl}(2,2,1,2) = C_{ij}(2,6); \\
\end{align*}
%
\begin{align*}
& C_{ijkl}(2,3,3,3) = C_{ij}(3,4); C_{ijkl}(3,3,2,3) = C_{ij}(3,4); \\
& C_{ijkl}(3,1,3,3) = C_{ij}(3,5); C_{ijkl}(3,3,3,1) = C_{ij}(3,5); \\
& C_{ijkl}(1,2,3,3) = C_{ij}(6,3); C_{ijkl}(3,3,1,2) = C_{ij}(3,6); \\
\end{align*}
%
\begin{align*}
& C_{ijkl}(3,1,2,3) = C_{ij}(4,5); C_{ijkl}(2,3,1,2) = C_{ij}(4,5); \\
& C_{ijkl}(1,2,2,3) = C_{ij}(6,4); C_{ijkl}(2,3,1,2) = C_{ij}(4,6); \\
\end{align*}
%
\begin{align*}
& C_{ijkl}(1,2,3,1) = C_{ij}(6,5); C_{ijkl}(3,1,1,2) = C_{ij}(5,6); \\
\end{align*}
% Tensor Equivalencies (1,x,y,z)
\begin{align*}
& C_{ijkl}(1,1,1,2) = C_{ijkl}(1,1,2,1); C_{ijkl}(1,3,1,3) = C_{ijkl}(3,1,3,1); \\
& C_{ijkl}(1,1,1,3) = C_{ijkl}(1,1,3,1); C_{ijkl}(1,3,2,1) = C_{ijkl}(3,1,1,2); \\
& C_{ijkl}(1,1,3,2) = C_{ijkl}(1,1,2,3); C_{ijkl}(1,3,1,3) = C_{ijkl}(3,1,2,2); \\
& C_{ijkl}(1,2,1,3) = C_{ijkl}(1,2,3,1); C_{ijkl}(1,3,2,3) = C_{ijkl}(3,1,2,3); \\
& C_{ijkl}(1,2,2,1) = C_{ijkl}(1,2,1,2); C_{ijkl}(1,3,1,2) = C_{ijkl}(3,1,1,3); \\
& C_{ijkl}(1,2,3,2) = C_{ijkl}(1,2,3,1); C_{ijkl}(1,3,2,3) = C_{ijkl}(3,1,3,3); \\
& C_{ijkl}(1,3,1,3) = C_{ijkl}(3,1,1,1); C_{ijkl}(1,3,3,3) = C_{ijkl}(3,1,3,3); \\
& C_{ijkl}(1,3,1,2) = C_{ijkl}(3,1,1,2); \\
\end{align*}
% Tensor Equivalencies (2,x,y,z)
\begin{align*}
& C_{ijkl}(2,1,1,1) = C_{ijkl}(1,2,1,1); C_{ijkl}(2,1,1,2) = C_{ijkl}(1,2,1,2); \\
& C_{ijkl}(2,1,1,3) = C_{ijkl}(1,2,1,3); C_{ijkl}(2,1,2,1) = C_{ijkl}(1,2,2,1); \\
& C_{ijkl}(2,1,2,2) = C_{ijkl}(1,2,2,2); C_{ijkl}(2,1,2,3) = C_{ijkl}(1,2,2,3); \\
& C_{ijkl}(2,1,3,1) = C_{ijkl}(1,2,3,1); C_{ijkl}(2,1,3,3) = C_{ijkl}(1,2,3,3); \\
& C_{ijkl}(2,2,1,3) = C_{ijkl}(2,2,3,1); C_{ijkl}(2,2,2,1) = C_{ijkl}(2,2,1,2); \\
& C_{ijkl}(2,2,2,2) = C_{ijkl}(2,2,2,3); C_{ijkl}(2,3,1,3) = C_{ijkl}(2,3,3,1); \\
& C_{ijkl}(2,3,2,1) = C_{ijkl}(2,3,1,2); C_{ijkl}(2,3,3,2) = C_{ijkl}(2,3,2,3); \\
\end{align*}
% Tensor Equivalencies (3,x,y,z)
\begin{align*}
& C_{ijkl}(3,1,1,3) = C_{ijkl}(1,3,1,3); C_{ijkl}(3,1,2,1) = C_{ijkl}(1,3,2,1); \\
& C_{ijkl}(3,1,3,2) = C_{ijkl}(1,3,3,2); C_{ijkl}(3,2,1,1) = C_{ijkl}(2,3,1,1); \\
& C_{ijkl}(3,2,1,2) = C_{ijkl}(2,3,1,2); C_{ijkl}(3,2,1,3) = C_{ijkl}(2,3,1,3); \\
& C_{ijkl}(3,2,2,1) = C_{ijkl}(2,3,2,1); C_{ijkl}(3,2,2,2) = C_{ijkl}(2,3,2,2); \\
& C_{ijkl}(3,2,2,3) = C_{ijkl}(2,3,2,3); C_{ijkl}(3,2,3,1) = C_{ijkl}(2,3,3,1); \\
& C_{ijkl}(3,2,3,2) = C_{ijkl}(2,3,3,2); C_{ijkl}(3,2,3,3) = C_{ijkl}(2,3,3,3); \\
& C_{ijkl}(3,3,1,3) = C_{ijkl}(3,3,3,1); C_{ijkl}(3,3,2,1) = C_{ijkl}(3,3,1,2); \\
& C_{ijkl}(3,3,3,2) = C_{ijkl}(3,3,2,3); \\
\end{align*}
Cijkl2Cij M-file Function

function [Cij] = Cijkl2Cij(Cijkl)
% Cij = Cijkl2Cij(Cijkl) converts Cij to Cijkl
%
% Input:
% (1) Cij is the [6x6] stiffness tensor in the single crystal reference frame at g = \{0,0,0\}
%
% Output:
% (1) Cijkl is the [3x3x3x3] array of modulii
%

% preallocate Cijkl
Cij= zeros(6,6);

% the trace of the [6x6]
Cij(1,1) = Cijkl(1,1,1,1); Cij(2,2) = Cijkl(2,2,2,2);
Cij(3,3) = Cijkl(3,3,3,3); Cij(4,4) = Cijkl(2,3,2,3);
Cij(5,5) = Cijkl(3,1,3,1); Cij(6,6) = Cijkl(1,2,1,2);

% the first row and column of the [6x6]
Cij(2,1) = Cijkl(2,2,1,1); Cij(1,2) = Cijkl(1,1,2,2);
Cij(3,1) = Cijkl(3,3,1,1); Cij(1,3) = Cijkl(1,1,3,3);
Cij(4,1) = Cijkl(2,3,1,1); Cij(1,4) = Cijkl(1,1,2,3);
Cij(5,1) = Cijkl(3,1,1,1); Cij(1,5) = Cijkl(1,1,3,1);
Cij(6,1) = Cijkl(1,2,1,1); Cij(1,6) = Cijkl(1,1,1,2);

% the remaining part
Cij(3,2) = Cijkl(3,3,2,2); Cij(2,3) = Cijkl(2,2,3,3);
Cij(4,2) = Cijkl(2,3,2,2); Cij(2,4) = Cijkl(2,2,2,3);
Cij(5,2) = Cijkl(3,1,2,2); Cij(2,5) = Cijkl(2,2,3,1);
Cij(6,2) = Cijkl(1,2,2,2); Cij(2,6) = Cijkl(2,2,1,2);

Cij(4,3) = Cijkl(2,3,3,3); Cij(3,4) = Cijkl(3,3,2,3);
Cij(5,3) = Cijkl(3,1,3,3); Cij(3,5) = Cijkl(3,3,3,1);
Cij(6,3) = Cijkl(1,2,3,3); Cij(3,6) = Cijkl(3,3,1,2);

Cij(5,4) = Cijkl(3,1,2,3); Cij(4,5) = Cijkl(2,3,3,1);
Cij(6,4) = Cijkl(1,2,2,3); Cij(4,6) = Cijkl(2,3,1,2);

Cij(6,5) = Cijkl(1,2,3,1); Cij(5,6) = Cijkl(3,1,1,2);

---------------------------------------------------------------

RMat of Bunge M-file Function

function rmat = RMatOfBunge(bunge, units)
% RMatOfBunge - Rotation matrix from Bunge angles.
%

% USAGE:
% rmat = RMatOfBunge(bunge, units)
% INPUT:
% bunge is 3 x n,
% the array of Bunge parameters
% units is a string,
% either 'degrees' or 'radians'
% OUTPUT:
% rmat is 3 x 3 x n,
% the corresponding rotation matrices
% if (nargin < 2)
%   error('need second argument, units: "degrees" or "radians"
% end
% if (strcmp(units, 'degrees'))
%   indeg = 1;
%   bunge = bunge*(pi/180);
% elseif (strcmp(units, 'radians'))
%   indeg = 0;
% else
%   error('angle units need to be specified: "degrees" or "radians"
% end
% n = size(bunge, 2);
% cbun = cos(bunge);
% sbun = sin(bunge);
% rmat = [
%   cbun(1, :).*cbun(3, :) - sbun(1, :).*cbun(2, :).*sbun(3, :);
%   sbun(1, :).*cbun(3, :) + cbun(1, :).*cbun(2, :).*sbun(3, :);
%   sbun(2, :).*sbun(3, :);
% -cbun(1, :).*sbun(3, :) - sbun(1, :).*cbun(2, :).*sbun(3, :);
% -sbun(1, :).*sbun(3, :) + cbun(1, :).*cbun(2, :).*sbun(3, :);
%   sbun(2, :).*cbun(3, :);
%   sbun(1, :).*sbun(2, :);
%   cbun(2, :)
% ];
% rmat = reshape(rmat, [3 3 n]);

RotateCijklReal M-file Function

function RotatedCijkl = RotateCijklReal(Cijkl,bunge,RadOrDeg)
% rotate the [6x6] Cij by a rotation described by bunge
% Inputs:
%    (1) Cij is the [6x6] stiffness tensor in the crystal reference frame
%    (2) bunge is an [nx3] array containing column vectors of Bunge angles
%    (3) RadOrDeg is a string containing either 'radians' or 'degrees'
%         describing the Bunge angle convention
%
% Outputs:
%    (1) Cij_rotated is a [6x6] array containing the elastic modulii in the
%         sample coordinate system after rotating by the Bunge angles
%
% disp('note that hex symmetry only for this function!')
% disp('C(i,j,k,l) can be easily modified for ')

% bunge = [0 45 0]; RadOrDeg = 'degrees';

% load CijNickel; Cij = CijNickel;
% disp('transposing rotate matrix!! are you sure!!');
% pause
rotate = RMatOfBunge(bunge,RadOrDeg);
% disp('rotating about sample Y are you sure??');
% pause
% rotate = RotateY(bunge);
% disp('applied rotatation matrix:')
% disp(rotate)

%% assign cijkl
% Cijkl = zeros(3,3,3,3);
RotatedCijkl = zeros(3,3,3,3);

%% rotate Cijkl to new reference frame
for i=1:3
    for j=1:3
        for k=1:3
            for l=1:3
                % Ctempijkl(i,j,k,l)=0.0;
                for p=1:3
                    for q=1:3
                        for m=1:3
                            for n=1:3
                                RotatedCijkl(i,j,k,l)=RotatedCijkl(i,j,k,l)+rotate(i,p)*rotate(j,q)*rotate(k,m)*rotate(l,n)*Cijkl(p,q,m,n);
                            end
                        end
                    end
                end
            end
        end
    end
end
end
end

end
end
end
end
RotateY M-file Function

```matlab
function [ MatrixOut ] = RotateY(angle)
%ROTATEY creates a rotation matrix about the x-axis in the sample ref.
%frame
% Detailed explanation goes here

angle = angle * pi()/180;

MatrixOut = [cos(angle) 0 sin(angle); 0 1 0; -sin(angle) 0 cos(angle)];
end
```

C2: Data Reconfiguration Codes – 2 Dimensional

Cij Spatial Coordinates 2D M-file

```matlab
% Cij Spatial Corrdinates 2D
... Code Takes ADam Pilchaks DREAM3D2PZFLEX Output file and reconfigures data in
... a form recoginized by FEM software PZFLEX for 2D samples

% Author: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014

clear all
clc

%define input and output file
%**changes for every new data set**

%input File
% outfile = textread('C:\FILE ROOT.out','','-1','headerlines',1);
outfile = textread('G:\0009 F110 model and data\0004 segmented files\010.out','','-1','headerlines',1);

%output File
% fileWriteID = fopen('C:\FILE ROOT.txt','w');
fileWriteID = fopen('G:\0009 F110 model and data\0004 segmented files\readpzflexGrain_0_new.txt','w');

%define what columns of data are needed for the particular form below
```
% Cij Stiffness Matrix IDs M-file

% Cij Stiffness Matrix IDs
... Code Takes ADam Pilchaks DREAM3D2PZFLEX Output file and reconfigures Stiffness matrix dat
... in a form recognizable by FEM software PZFLEX for 2D samples

% Author: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014
clear all
clc

%define input and output file
...**changes for every new data set**
...example
... outfile = textread('C:\Users\JohnsoDR\Desktop\test20microngrains.dat','-',-1,'headerlines',1);
... fileWriteID = fopen('C:\Users\dariusj\Desktop\pilchak\pilchak_blackshire venture samples\Reg1_Polish2\Reg1_Polish2 grain id.txt','w');

%input file
outfile = textread('C:\ FILE ROOT.out','-',-1,'headerlines',1);

%output file
fileWriteID = fopen('C:\FILE ROOT.txt','w');

%define what columns of data are needed for the particular form below
...prop mat7500.
...10.0e10  7.95e10  8.14e10  0.0   0.0   0.0  10.0e10
...8.41e10  0.0     0.0   0.0   11.7e10  0.0   0.0
...0.0      2.30e10  0.0   0.0   2.30e10  0.0   2.325e10

raw_grain_number = outfile(:,3);
stiffness_matrix_values = outfile(:,4:24);

%*** density changes for a given material***
density = 7500  %NI
... Titanium density = 4430;  
... Nickel density = 7500

%sort and refine grain number ID so it returns unique value in the order 
... it first appear in with no repititions
A = raw_grain_number;
[As,idx] = sort(A);
[U,ij] = unique(As,'first');
A(sort(idx(ij)))

refined_grain_num = A(sort(idx(ij)))

%sort and refine matrix values rows so it returns unique rows in the order 
... it first appear in with no repititions
particular_rows = (sort(idx(ij)));
refined_matrix_values = stiffness_matrix_values(particular_rows,:);

fprintf(fileWriteID,'matr
');
fprintf(fileWriteID,'        type lean 
');

% output data in format needed
for p = 1:length(refined_grain_num)
    if refined_grain_num(p)<10
        fprintf(fileWriteID,'        prop mat%d   %-d \n',refined_grain_num(p),density);
    elseif refined_grain_num(p)<100 && refined_grain_num(p) >= 10
        fprintf(fileWriteID,'        prop mat%d  %-d \n',refined_grain_num(p),density);
    elseif refined_grain_num(p)>=100
        fprintf(fileWriteID,'        prop mat%d %-d \n',refined_grain_num(p),density);
    end
    fprintf(fileWriteID,'\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e
\n',refined_matrix_values(p,1),refined_matrix_values(p,2),refined_matrix_values(p,3),refined_matrix_values(p,4),refined_matrix_values(p,5),refined_matrix_values(p,6),refined_matrix_values(p,7));
    fprintf(fileWriteID,'\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e
\n',refined_matrix_values(p,8),refined_matrix_values(p,9),refined_matrix_values(p,10),refined_matrix_values(p,11),refined_matrix_values(p,12),refined_matrix_values(p,13),refined_matrix_values(p,14));
    fprintf(fileWriteID,'\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e\t%6.4e
\n',refined_matrix_values(p,15),refined_matrix_values(p,16),refined_matrix_values(p,17),refined_matrix_values(p,18),refined_matrix_values(p,19),refined_matrix_values(p,20),refined_matrix_values(p,21));
end

fprintf(fileWriteID,'  end \n');

% end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

C3: Data Reconfiguration Codes – 3 Dimensional

Cij Spatial Coordinates 3D Brut Force non-clustering M-file

% Cij Spatial Coordinates 3D Brut Force non-clustering
... Code Takes ADam Pilchaks DREAM3D2PZFLEX Output file and reconfigures data
... in a form recognized by FEM software PZFLEX for 2D samples

% Author: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014

clear all
clc

%define input and output file
... **changes for every new data set**
... outfile = textread('C:s\Users\JohnsoDR\Desktop\test20microngrains.dat','-',1,'headerlines',1);
... fileWriteID = fopen('layer_7.txt','w');

%input file
outfile = textread('C:\FILE ROOT.out','-',1,'headerlines',1);
% output file
fileWriteID = fopen('layer_1.txt','w');

% z layer
Z_begin = 1;
Z_end = Z_begin + 1;

% define what columns of data are needed for the particular form below
% 
% %c    regn mat6 $xbegin2 $xend2 $ybegin2 $yend2 $zbegin2 $zend2
%   regn mat5 1. 21. 1. 2. 1. 2.
%   regn mat5 1. 21. 3. 4. 1. 2.
%   regn mat5 1. 21. 4. 5. 1. 2.
%   regn mat5 1. 21. 5. 6. 1. 2.
%   regn mat5 1. 21. 6. 7. 1. 2.

X_begin = outfile(:,1);
Y_begin = outfile(:,2);
grain_num = outfile(:,3);

% define x end
for p = 1:1:length(X_begin)
    X_end = X_begin(p) + 1;
end

% define y end
for z = 1:1:length(Y_begin)
    Y_end = Y_begin(z) + 1;
end

% write output file
for p = 1:1:length(grain_num)
    if grain_num(p)<10
        fprintf(fileWriteID,'        regn mat%d    %i. %i. %i. %i. %i. %i.
',grain_num(p),X_begin(p),X_end(p),Y_begin(p),Y_end(p), Z_begin, Z_end);
    elseif grain_num(p)<100 && grain_num(p) >= 10
        fprintf(fileWriteID,'        regn mat%d   %i. %i. %i. %i. %i. %i.
',grain_num(p),X_begin(p),X_end(p),Y_begin(p),Y_end(p), Z_begin, Z_end);
    elseif grain_num(p)>=100
        fprintf(fileWriteID,'        regn mat%d  %i. %i. %i. %i. %i. %i.
',grain_num(p),X_begin(p),X_end(p),Y_begin(p),Y_end(p), Z_begin, Z_end);
    end
end

******************************************************************************
consolidate_CijSpatialCoordinates_files M-file

% consolidate_CijSpatialCoordinates_files
% this codes strings together files produced by
"Cij_Spatial_coordinates_3D_brut_force_noncluster"
% in a effort to consolidates individaul layers into a 3Dimensional
% sample

% Author: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014

%10 layers
p = system('copy layer_1.txt + layer_2.txt + layer_3.txt + layer_4.txt + layer_5.txt + layer_6.txt +
layer_7.txt + layer_8.txt + layer_9.txt + layer_10.txt ConsolidatedFile.txt');

% 100 layer consildation
%p = system('copy layer_1.txt + layer_2.txt + layer_3.txt + layer_4.txt + layer_5.txt + layer_6.txt +
layer_7.txt + layer_8.txt + layer_9.txt + layer_10.txt + layer_11.txt + layer_12.txt + layer_13.txt +
layer_14.txt + layer_15.txt + layer_16.txt + layer_17.txt + layer_18.txt + layer_19.txt + layer_20.txt +
layer_21.txt + layer_22.txt + layer_23.txt + layer_24.txt + layer_25.txt + layer_26.txt +
layer_27.txt + layer_28.txt + layer_29.txt + layer_30.txt + layer_31.txt + layer_32.txt + layer_33.txt +
layer_34.txt + layer_35.txt + layer_36.txt + layer_37.txt + layer_38.txt + layer_39.txt +
layer_40.txt + layer_41.txt + layer_42.txt + layer_43.txt + layer_44.txt + layer_45.txt + layer_46.txt +
layer_47.txt + layer_48.txt + layer_49.txt + layer_50.txt + layer_51.txt + layer_52.txt +
layer_53.txt + layer_54.txt + layer_55.txt + layer_56.txt + layer_57.txt + layer_58.txt + layer_59.txt +
layer_60.txt + layer_61.txt + layer_62.txt + layer_63.txt + layer_64.txt + layer_65.txt +
layer_66.txt + layer_67.txt + layer_68.txt + layer_69.txt + layer_70.txt + layer_71.txt + layer_72.txt +
layer_73.txt + layer_74.txt + layer_75.txt + layer_76.txt + layer_77.txt + layer_78.txt +
layer_79.txt + layer_80.txt + layer_81.txt + layer_82.txt + layer_83.txt + layer_84.txt + layer_85.txt +
layer_86.txt + layer_87.txt + layer_88.txt + layer_89.txt + layer_90.txt + layer_91.txt +
layer_92.txt + layer_93.txt + layer_94.txt + layer_95.txt + layer_96.txt + layer_97.txt + layer_98.txt +
layer_99.txt ConsolidatedFile.txt');

... **** dependent on how many layers are in the sample they need to be acounted for in the
system of this code, just copy and paste into the system

Cij Stiffness Matrix IDs 3D M-file

% Cij Stiffness Matrix IDs
% Code Takes ADam Pilchaks DREAM3D2PZFLEX Output file and reconfigures Stiffness
matrix data
% in a form recognized by FEM software PZFLEX for 3D samples

% Author: Darius Johnson, AFRL/RXCA, University of Dayton, 7 June 2014

clear all
clc

%define input and output file
... **changes for every new data set**
...
fileWriteID = fopen('rolled_grain_stiff_mtx.txt','w');

%output file
fileWriteID = fopen('FILE NAME.txt', 'w');
%  *****CODE HAS TO BE IN SAME DIRECTORY AS INPUT FILES*****

% %
... **** dependent on how many layers are in the sample they need to be acounted for in the
system of this code, just copy and paste into the system

% 5 layers
% a = system('copy layer_0.out + layer_1.out + layer_2.out + layer_3.out + layer_4.out
bigfile.out');

% 200 layers
a = system('copy layer_0.out + layer_1.out + layer_2.out + layer_3.out + layer_4.out +
layer_5.out + layer_6.out + layer_7.out + layer_8.out + layer_9.out + layer_10.out + layer_11.out +
layer_12.out + layer_13.out + layer_14.out + layer_15.out + layer_16.out + layer_17.out +
layer_18.out + layer_19.out + layer_20.out + layer_21.out + layer_22.out + layer_23.out +
layer_24.out + layer_25.out + layer_26.out + layer_27.out + layer_28.out + layer_29.out +
layer_30.out + layer_31.out + layer_32.out + layer_33.out + layer_34.out + layer_35.out +
layer_36.out + layer_37.out + layer_38.out + layer_39.out + layer_40.out + layer_41.out +
layer_42.out + layer_43.out + layer_44.out + layer_45.out + layer_46.out + layer_47.out +
layer_48.out + layer_49.out + layer_50.out + layer_51.out + layer_52.out + layer_53.out +
layer_54.out + layer_55.out + layer_56.out + layer_57.out + layer_58.out + layer_59.out +
layer_60.out + layer_61.out + layer_62.out + layer_63.out + layer_64.out + layer_65.out +
layer_66.out + layer_67.out + layer_68.out + layer_69.out + layer_70.out + layer_71.out +
layer_72.out + layer_73.out + layer_74.out + layer_75.out + layer_76.out + layer_77.out +
layer_78.out + layer_79.out + layer_80.out + layer_81.out + layer_82.out + layer_83.out +
layer_84.out + layer_85.out + layer_86.out + layer_87.out + layer_88.out + layer_89.out +
layer_90.out + layer_91.out + layer_92.out + layer_93.out + layer_94.out + layer_95.out +
layer_96.out + layer_97.out + layer_98.out + layer_99.out + layer_100.out + layer_101.out +
layer_102.out + layer_103.out + layer_104.out + layer_105.out + layer_106.out + layer_107.out +
layer_108.out + layer_109.out + layer_110.out + layer_111.out + layer_112.out + layer_113.out +
layer_114.out + layer_115.out + layer_116.out + layer_117.out + layer_118.out + layer_119.out +
layer_120.out + layer_121.out + layer_122.out + layer_123.out + layer_124.out + layer_125.out +
layer_126.out + layer_127.out + layer_128.out + layer_129.out + layer_130.out + layer_131.out +
layer_132.out + layer_133.out + layer_134.out + layer_135.out + layer_136.out + layer_137.out +
layer_138.out + layer_139.out + layer_140.out + layer_141.out + layer_142.out + layer_143.out +
layer_144.out + layer_145.out + layer_146.out + layer_147.out + layer_148.out + layer_149.out +
layer_150.out + layer_151.out + layer_152.out + layer_153.out + layer_154.out + layer_155.out +
layer_156.out + layer_157.out + layer_158.out + layer_159.out + layer_160.out + layer_161.out +
layer_162.out + layer_163.out + layer_164.out + layer_165.out + layer_166.out + layer_167.out +
layer_168.out + layer_169.out + layer_170.out + layer_171.out + layer_172.out + layer_173.out +
layer_174.out + layer_175.out + layer_176.out + layer_177.out + layer_178.out + layer_179.out +
layer_180.out + layer_181.out + layer_182.out + layer_183.out + layer_184.out + layer_185.out +
layer_186.out + layer_187.out + layer_188.out + layer_189.out + layer_190.out + layer_191.out +
layer_192.out + layer_193.out + layer_194.out + layer_195.out + layer_196.out + layer_197.out +
layer_198.out + layer_199.out bigfile.out');

outfile = textread('bigfile.out');

%%

%define what columns of data are needed for the particular form below;
%prop mat7500.
%10.0e10  7.95e10  8.14e10  0.0     0.0     0.0     10.0e10
raw_grain_number = outfile(:,3);
stiffness_matrix_values = outfile(:,4:24);
density = 7500; %*** changes for a given material***

%sort and refine grain number ID so it returns unique value in the order ... it first appeared in with no repetitions
A = raw_grain_number;
[As,idx] = sort(A);
[U,ij] = unique(As,'first');
A(sort(idx(ij))) =;

refined_grain_num = A(sort(idx(ij))) ;

%sort and refine matrix values rows so it returns unique rows in the order ... it first appeared in with no repetitions
particular_rows = (sort(idx(ij)));
refined_matrix_values = stiffness_matrix_values(particular_rows,:);

% output data in format needed
for p = 1:length(refined_grain_num)
    if refined_grain_num(p)<10
        fprintf(fileWriteID,'        prop mat%d   %-d
',refined_grain_num(p),density);
    elseif refined_grain_num(p)<100 && refined_grain_num(p) >= 10
        fprintf(fileWriteID,'        prop mat%d  %-d
',refined_grain_num(p),density);
    elseif refined_grain_num(p)>=100
        fprintf(fileWriteID,'        prop mat%d %-d
',refined_grain_num(p),density);
    end
    fprintf(fileWriteID,'	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e
',refined_matrix_values(p,1),refined_matrix_values(p,2),refined_matrix_values(p,3),refined_matrix_values(p,4),refined_matrix_values(p,5),refined_matrix_values(p,6),refined_matrix_values(p,7));
    fprintf(fileWriteID,'	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e
',refined_matrix_values(p,8),refined_matrix_values(p,9),refined_matrix_values(p,10),refined_matrix_values(p,11),refined_matrix_values(p,12),refined_matrix_values(p,13),refined_matrix_values(p,14));
    fprintf(fileWriteID,'	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e	%6.4e
',refined_matrix_values(p,15),refined_matrix_values(p,16),refined_matrix_values(p,17),refined_matrix_values(p,18),refined_matrix_values(p,19),refined_matrix_values(p,20),refined_matrix_values(p,21));
end

% end

C4: Extrapolation of dominant orientation from Pole figures Code
Dominant orientation from Pole figures M-file

% PF stuff
% make sure M-file is in the same directory as the text files of code will
% not compute properly

txt_index=dir('f110_0 pf.txt');
fid2=fopen('names.txt','w+');
fprintf(fid2,'Polar\n');
fprintf(fid2,'Azimuthal\n');

for i=1:size(txt_index,1)
    fid=fopen(txt_index(i,1).name,'r+');
    fid2=fopen('names.txt','a');
    data=cell2mat(textscan(fid, '%n %n %n', 'CollectOutput',1,'CommentStyle','#'));
peak_intensity = max(data(:,3));
[r,c] = find(data(:,3)==peak_intensity);

% the orientations you care about are here:
maxPfLine = data(r,:)

% the dominant orientations are:
phi1 = data(r,2)
PHI = data(r,1)

% everything above here will read in the pole figure.
% all stuff below this you probably dont need.

%     data = data(1369:2736,:); % grab only the 011 pole fig
%     list(:,1)=[90 90 90 45 45 45 45];
%     list(:,2)=[0 90 180 270 45 135 225 315];
%
%     for j=1:size(list,1);
%         test1=data(:,1)==list(j,1);
%         test2=data(:,2)==list(j,2);
%         test3=data(:,3)==list(j,3);
%         test4=sum([test1 test2],2);
%         aptest = test1.*test2;
%         test5=find(aptest(:,1)==1);
%         test6(j,:)=data(test5,3);
%     end
%     plot(list(:,2),test6)
%     title(txt_index(i,1).name)
%     test7=[avg(test6(1),test6(2))]
%
% output(:,i+2)=test6;
% output(,:);
% fprintf(fid2,'%s\n',txt_index(i,1).name);
% fclose all;
end
% save -ascii output.txt output

C5: Consolidation of PZFLEX output Data files

CONSOLIDATE_1D-PZFLEX_DATA M-file

% CONSOLIDATE_1D-PZFLEX_DATA
% Author: Josiah Dierken, Darius Johnson,  AFRL/RXCA  2012

function CONSOLIDATE_1D_PZFLEX_DATA
% Code loads in data from a series of .dat files, removes the time data
%(first column), and consolidates the response data into a single text
% file.

clear all %Clear any existing data from MATLAB
close all %Close any existing MATLAB windows
clc %Clear text / data in the MATLAB control window

% Initializing Needed Variables
TOTAL_FILES=0;
INPUT_DATA=0;
OUTPUT_DATA=[];

% Reading in the filename and pathname from user selection
[dat_filename,dat_pathname]=uigetfile({'*.*'},'Select Data File','Multiselect','on');

% Depending on how many files are selected the filename returned by
% uigetfile()" will be stored either as a character string or a cell array.
% If only one file is selected than the filename is stored as a character
% string. This is checked by the function "ischar()"
if ischar(dat_filename)==1

% Combining the pathname and filename into a single string
dat_pathname_filename = [dat_pathname dat_filename];
TOTAL_FILES=11;

% Loading in data from the file selected
INPUT_DATA=load(dat_pathname_filename);

% OUTPUT_DATA was initialized to an empty set [ ] so we append columns
% 2->N of "INPUT_DATA" to remove the time component
OUTPUT_DATA=[OUTPUT_DATA INPUT_DATA(:,2:end)];
else
% If multiple files are selected, the filenames are stored as a cell array with a length equal to the number of files selected.
TOTAL_FILES = length(dat_filename);

% Using the for-loop to cycle through files and append the data
for i = 1:TOTAL_FILES

% Combining the pathname and filename into a single string
dat_pathname_filename = [char(dat_pathname) char(dat_filename(i))];

% Loading in data from the file selected
INPUT_DATA = load(dat_pathname_filename);

% OUTPUT_DATA was initialized to an empty set [] so we append columns 2->N of "INPUT_DATA" to remove the time component. After we append the first set of data we continue removing the time component and appending data from additional files to OUTPUT_DATA
OUTPUT_DATA = [OUTPUT_DATA INPUT_DATA(:,2:end)];
end

% Getting the size of the matrix OUTPUT_DATA vs=vertical dimmension, hs=horizontal dimmension
[vs hs] = size(OUTPUT_DATA);

% Reading in the filename and pathname from which the user selected to save the data
[Save_Filename Save_Pathname] = uiputfile('*.txt','Save Output Data');

% Setting up the file identifier FileID. This allocates memory for the file to be created and tells MATLAB that we want to be able to write and read as a text file (by using 'wt+')
FileID = fopen([Save_Pathname Save_Filename], 'wt+');

% Using a for-loop to cycle through rows of the matrix OUTPUT_DATA for j = 1:vs
  % Writing a single row of OUTPUT_DATA to the file
  fprintf(FileID,'%E ',OUTPUT_DATA(j,:));

  % Writing in a new-line character ("Pressing Enter") to move to the next line for the next row of data
  fprintf(FileID,'%n');
end

% Closing out the file identifier to close the allocated memory and save the data to the text file
fclose(FileID);
end

........................................
APPENDIX D

PZFLEX MODEL SCRIPT

Contents of Appendix D

D1: 2 Dimensional Samples Pitch-Catch Base Model
D2: 2 Dimensional Samples Pulse-Echo Base Model
D3: 3 Dimensional Samples Pitch-Catch Base Model

D1: 2 Dimensional Samples Pitch-Catch Base Model

symb x2 = 1.01e-3
symb y1 = 0.0
symb y2 = 8.51e-3
symb i1 = 1
symb i2 = $i1 + nint ( ( $x2 - $x1 ) / $box )
symb indgrd = $i2
symb j1 = 1
symb j2 = $j1 + nint ( ( $y2 - $y1 ) / $box )
symb jndgrd = $j2
grid $indgrd $jndgrd
geom
xcrd $x1 $x2 $i1 $i2
ycrd $y1 $y2 $j1 $j2
end
symb #read readpzflexCij.txt
symb #read readpzflexGrain.txt
grph
nview 1
line none
set imag tiff
revr y
plot matr
imag model.tif
end
calc
c    pres
disp
    max xdsp xmin xmax
    max ydsp ymin ymax
c    max pres pmin pmax
 pres
    max pres pmin pmax
end
data hist drv1 * function.dat
func hist drv1
pout
    hist xdsp 1 Si2 1 Sj2 Sj2 1
    hist ydsp 1 Si2 1 Sj2 Sj2 1
    hist xdsp 1 Si2 1 1 1 1
    hist ydsp 1 Si2 1 1 1 1
end
symb iplod = $i1 + $numelem
plod
    pdef pld1 func
    vctr vec1 0 -1 0
c    sdef pld1 vec1 1 $i2 $j2 $j2
    sdef pld1 vec1 1 $i2 $j1 $j1
end
boun
    side xmin symm
    side xmax symm
    side ymin absr
    side ymax absr
end
time * * 0.95

prcs
symb nloops = 50 if noexist
symb simtime = 3.0 * $y2 / $velmax
symb #get { step } timestep
symb nstepstot = nint ( $simtime / $step )
symb nsteps = nint ( $nstepstot / $nloops )
symb n = 0
data
    file out $labl.flxdata
end
do loopi 1 $nloops
exec $nsteps
data
    out disp
end
data
    math magd = ( ({ xdsp } ** 2.) + ( { ydsp } ** 2.) ) ** 0.5
end
graph
    nview 1
line none
    colr tabl data 6
    set imag tiff
    revr y
    plot magd rang 0. 1.e-15
    plot magd
    imag magd$n.tif
    plot pres rang -6.e-1 6.e-1
    plot pres
    imag pres$n.tif
symb n = $n + 1
end$ loopi
graph
    nview 1
set imag tiff
line none

plot xmax
imag xmax.tif
plot ymax
imag ymax.tif
plot pmax
imag pmax.tif

end
symb #save symb.$labl
stop

**D2: 2 Dimensional Samples Pulse-Echo Base Model**

mem 1615 100000
rest no
mp
    omp 2
end
symb #get { labl } jobname

symb velmin = 3039.
symb velmax = 6419.
symb dens = 2700.

symb freqint = 1.e6
symb freqmax = 2. * $freqint
symb wavemin = $velmin / $freqmax

symb numelem = 20
symb box = 10.0e-6
symb x1 = 0.0
symb x2 = 1.01e-3
symb y1 = 0.0

symb y2 = 8.51e-3
symb i1 = 1
symb i2 = $i1 + nint ( ( $x2 - $x1 ) / $box )

symb indgrd = $i2
symb j1 = 1
symb j2 = $j1 + nint ( ( $y2 - $y1 ) / $box )
symb jndgrd = $j2

grid $indgrd $jndgrd

geom
    xcrd $x1 $x2 $i1 $i2
    ycrd $y1 $y2 $j1 $j2
end
symb #read readpzflexCij.txt
symb #read readpzflexGrain.txt

grph
    nview 1
    line none
    set imag tiff
    revr y
c    vpnt 0 -1 0
c    vert 0 0 0
c    eye 0 0 0
plot matr
    imag model.tif
end

calc
c    pres
disp
    max xdsp xmin xmax
    max ydsp ymin ymax
    max pres pmin pmax
c    max pres pmin pmax

end
data hist drv1 * function.dat
func hist drv1
pout
  hist xdsp 1 Si2 1 Sj2 $j2 1
  hist ydsp 1 Si2 1 Sj2 $j2 1
  hist xdsp 1 Si2 1 1 1 1
  hist ydsp 1 Si2 1 1 1 1
end

symb iplod = $i1 + $numelem
plod
  pdef pld1 func
  vctr vec1 0 -1 0
  sdef pld1 vec1 1 $i2 $j2 $j2
  sdef pld1 vec1 1 $i2 $j1 $j1
end

boun
  side xmin symm
  side xmax symm
  side ymin absr
  side ymax absr
end

time * * 0.95

prcs
symb nloops = 50 if noexist
symb simtime = 3.0 * $y2 / $velmax
symb #get { step } timestep
symb nstepstot = nint ( $simtime / $step )
symb nsteps = nint ( $nstepstot / $nloops )
symb n = 0

data
  file out $labl.flxdata
end

do loopi I 1 $nloops
  exec $nsteps
  data
    out disp
  end

data
  math magd = ( ( { xdsp } ** 2. ) + ( { ydsp } ** 2. ) ) ** 0.5
end

graph
  nview 1
  line none
  colr tabl data 6
  set imag tiff
  revr y
  c vpnt 0 -1 0
  c vert 0 0 0
  c eye 0 0 0
plot magd rang 0. 1.e-15

plot magd
  imag magd$n.tif
plot pres rang -6.e-1 6.e-1
  imag pres$n.tif
symb n = $n + 1
end
end$ loopi

graph
  nview 1
  set imag tiff
  line none
plot xmax
  imag xmax.tif
plot ymax
  imag ymax.tif
plot pmax
  imag pmax.tif
plot pmax rang 0.0 8.0
symb #save symb.$labl
stop

D3: 3 Dimensional Samples Pitch-Catch
Base Model

mem 1615 1000
rest no
mp
  omp 2
end

symb #get { labl } jobname

symb velmin = 3039.
symb velmax = 6419.
symb dens = 2700.

/* Alum
symb freqint = 1.e6
Drive frequency of interest 5MHz
symb freqmax = 2. * $freqint
symb wavemin = $velmin / $freqmax

/* 324 microns wavelength
at 10MHz for steel
symb numelem = 20
  c symb box = $wavemin / $numelem
  symb box = 5.0e-5

  c symb x1 = 0.0
  c symb x2 = 25.e-3

  /* 25mm cube
  c symb y1 = 0.0
  c symb y2 = 25.e-3
  c symb z1 = 0.0
  c symb z2 = 12.5e-3
  c symb x1 = 0.0

  /* 10mm cube 200 elements X
  c symb y1 = 0.0
  c symb y2 = 10.e-3
  c symb z1 = 0.0
  c symb z2 = 5.0e-3

  symb x1 = 0.0
  symb x2 = 10.05e-3

  /* 10.005mm cube
  201 elements X
  symb y1 = 0.0
  symb y2 = 10.05e-3
  symb z1 = 0.0
  symb z2 = 10.0e-3

  symb i1 = 1
  symb i2 = $i1 + nint ( ( $x2 - $x1 ) / $box )

  /* 156 elements = 5mm size divided by
  32.4um FEM element size
  symb indgrd = $i2
  symb j1 = 1
  symb j2 = $j1 + nint ( ( $y2 - $y1 ) / $box )
  symb jndgrd = $j2
  symb k1 = 1
  symb k2 = $k1 + nint ( ( $z2 - $z1 ) / $box )
  symb kndgrd = $k2

  grid $indgrd $jndgrd $kndgrd

  geom
    xcrd $x1 $x2 $i1 $i2
    ycrd $y1 $y2 $j1 $j2
    zcrd $z1 $z2 $k1 $k2
end

symb #read readpzflexCij.txt

symb cdepth = 3
symb xbegin = ( nint ( $j2 / 2. ) )
symb xend = ( nint ( $j2 / 2. ) + 1. )
symb ybegin = ( $j2 - ( nint ( $cdepth ) ) )

/* set bottom notch or slot position
   as max Z minus the crack depth 'cdepth'
symb yend = $j2

symb #read readpzflexGrain.txt

grph
  c  nvew 1
  c  line off
  c  set imag tiff
     /* set imag output as TIFF
        image files
  c  plot matr
  c  imag model.tif

  nvew 4
  line off
  c  colr tabl data 6
     set imag tiff
     vpnt 0. 0. 0.
     vert 0. 0. 1.
     eye 1. 0. 0.
     view 4
     block b2 $i1 $i2 $j1 $j2 199 199
     plot matr block b2
     block b2 cler
     imag model.tif

  end
c  vpnt 0. 0. 0.
c  vert -1. 0. 0.
c  eye 0. 0. 1.
c  view 2
  block b2 $i1 $i2 $j1 $j2 200 200
  plot matr block b2
  block b2 cler
  imag model.tif

  end
c  vpnt 0. 0. 0.
c  vert 0. 0. 1.
c  eye 0. 1. 0.
c  view 3
  block b2 200 200 $j1 $j2 $k1 $k2
  plot matr block b2
  block b2 cler
  imag model.tif

  end

c  vpnt 0. 0. 0.
c  vert -1. 0. 0.
c  eye 0. 0. 1.
c  view 2
  plot matr

c  vpnt 0. 0. 0.
c  vert 0. 0. 1.
c  eye 0. 1. 0.
c  view 3
  plot matr

c  vpnt 0. 0. 0.
c  vert 0. 0. 1.
c  eye 0. 1. 0.
c  view 3
  plot matr

c  calc
     pres
     disp
     max xdsp xmin xmax
     max ydsp ymin ymax
     max zdsp zmin zmax
  end

data hist drv1 * function1MHz1cyc.dat
     /* read data from function.dat into 'drv1'
func hist drv1
symb xmid_5 = ( nint ( ( $i2 / 2. ) - 5. ) )
symb xmid5 = ( nint ( ( $i2 / 2. ) + 5. ) )

pout
    hist xdsp $xmid_5 $xmid5 1 $j2 $j2 1 /* save xdisplacement across entire
material surface
    hist ydsp $xmid_5 $xmid5 1 $j2 $j2 1 /* save ydisplacement across entire
materials surface
    hist xdsp 500 500 1 $j1 $j2 1 /* U(z)
    displacement at column 500
    hist ydsp 500 500 1 $j1 $j2 1 /* W(z)
    displacement at column 500
end

symb iplod = $i1 + $numelem
plod
    pdef pld1 func
    vctr vec1 1 0 0
    /* Lwave right left X
    sdef pld1 vec1 1 1 1 $j2 1 $k2
    /* Lwave right left X
    c vctr vec1 0 1 0
    /* Lwave left right Y
    c sdef pld1 vec1 1 $i2 1 1 1 $k2
    /* Lwave left right Y
    c vctr vec1 0 0 1
    /* Lwave bottom up Z
    c sdef pld1 vec1 1 $i2 1 $j2 1 1
    /* Lwave bottom up Z
    c pdef pld1 func
    c vctr vct1 1 0 0
    c sdef pld1 vct1 $niplod $niplod $nynodes
    c vctr vec1 1 0 0
    /* SAW top left
    c sdef pld1 vec1 1 2 $j2 $j2 /* SAW top left

boun
    side xmin absr /* right left
    X
    side xmax absr
    side ymin symm
    side ymax symm
    side zmin symm
    side zmax symm
    side xmin symm /* left right
    Y
    side xmax symm
    side ymin absr
    side ymax absr
    side zmin symm
    side zmax absr

end
time * * 0.95
prcs
symb nloops = 50 if noexist /* number of
data snapshots to make
symb simtime = 3.0 * $x2 / $velmax /* simulation time - set to long enough for
wave to propagate across material
symb #get { step } timestep /* find
timestep
symb nstepstot = nint ( $simtime / $step ) /* calculate total number of timesteps
required
symb nsteps = nint ( $nstepstot / $nloops )
/* time step per loop
symb n = 0

data
    file out "$labl.flxdato"
end

do loop i 1 $nloops
exec $nsteps /* run nstep timesteps
data
    out disp
end
data
    math magd = ( ( { xdsp } ** 2. ) + ( { ydsp } ** 2. ) + ( { zdsp } ** 2. ) ) ** 0.5
    math magd = ( ( { ydsp } ** 2. ) + ( { zdsp } ** 2. ) ) ** 0.5
end
graph
    nvew 1
    line off
        colr tabl data 6
            set imag tiff
/* set imag output as TIFF
image files
    c plot magd rang 0. 1.e-15
/* good for SAW, head,
SSLW, L and S
    c plot magd
    c plot pres

    nvew 4
line off
    c colr tabl data 6
        set imag tiff
        vpnt 0. 0. 0.
        vert 0. 0. 1.
        eye 1. 1. 1.
        view 1
        plot pres rang -3.e0 1.5e1
    vpnt 0. 0. 0.

vert -1. 0. 0.
eye 0. 0. 1.
view 2
plot pres rang -3.e0 1.5e1

vpnt 0. 0. 0.
vert 0. 0. 1.
eye 0. 1. 0.
view 3
plot pres rang -3.e0 1.5e1

vpnt 0. 0. 0.
vert 0. 0. 1.
eye 1. 0. 0.
view 4
plot pres rang -3.e0 1.5e1

imag test$n.tif
/* output screen to new tiff file or
avi
symb n = $n + 1
end
end$ loopi

graph
    nvew 1
    set imag tiff
ttl 1
Max xdisp through model at any time
(beam profile)
c plot pmax
    vpnt 0. 0. 0.
    vert 0. 0. 1.
eye 1. 1. 1.
plot xmax
imag xmax1.tif
plot xmin
imag xmin1.tif
plot ymax
imag ymax1.tif
plot ymin
imag ymin1.tif

vpnt 0. 0. 0.
vert 0. 0. 1.

175
eye 1. 0. 0.

plot xmax
imag xmax2.tif
plot xmin
imag xmin2.tif
plot ymax
imag ymax2.tif
plot ymin
imag ymin2.tif

depth 0. 0. 0.
vert -1. 0. 0.
eye 0. 0. 1.

plot xmax
imag xmax3.tif
plot xmin
imag xmin3.tif
plot ymax
imag ymax3.tif
plot ymin
imag ymin3.tif

end

symb #get { valsmooth } datamax maxp    /* find maximum pressure value
symb #save symb.$labl                /* save symbol variables to file
stop
APPENDIX E

MICROSTRUCTURE CHARACTERIZATION

Contents of Appendix E

E1: Nickel Based Super Alloy 1000X8000um Segmentation Characterization
E2: MTR Titanium Alloy 1000X8500um Segmentation Characterization
E1: Nickel Based Super Alloy 1000X8000um Segmentation Characterization
E2: MTR Titanium Alloy 1000X8500um Segmentation Characterization