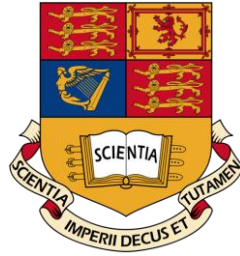


Imperial College of
Science Technology and Medicine



A Virtual Grain Structure Representation System for Micromechanics Simulations

**A Thesis submitted for the Degree of
Doctor of Philosophy of Imperial College**

by

Pan Zhang

November 2011

Department of Mechanical Engineering

Imperial College, London, SW7 2AZ

Abstract

Representing a grain structure within a combined finite element computer aided engineering environment is essential for micromechanics simulations. Methods are required to effectively generate high-fidelity virtual grain structures for accurate studies. A high-fidelity virtual grain structure means a statistically equivalent structure in conjunction with desired grain size distribution features, and must be represented with realistic grain morphology. A family of controlled Poisson Voronoi tessellation (CPVT) models have been developed in this work for systematically generating virtual grain structures with the aforementioned properties.

Three tasks have been accomplished in the development of the CPVT models: (i) defining the grain structure's regularity that specifies the uniformity of a tessellation as well as deriving a control parameter based on the regularity; (ii) modelling the mapping from a grain structure's regularity to its grain size distribution; and (iii) establishing the relation between a set of physical parameters and a distribution function. A one-gamma distribution function is used to describe a grain size distribution characteristic and a group of four physical parameters are employed to represent the metallographic measurements of a grain size distribution property. Mathematical proofs of the uniqueness of the determination of the distribution parameter from the proposed set of physical parameters have been studied, and an efficient numerical procedure is provided for computing the distribution parameter.

Based on the general scheme, two- and three-dimensional CPVT models have been formulated, which respectively define the quantities of regularity and control parameters, and model the mapping between regularity and grain size distribution. For the 2D-CPVT model, statistical tests have been carried out to validate the accuracy and robustness of regularity and grain size distribution control. In addition, micrographs with different grain size distribution features are employed to examine the capability of the 2D-CPVT model to generate virtual grain structures that meet physical measurements. A crystal plasticity finite element (CPFE) simulation of plane strain uniaxial tension has been performed to show the effect of grain size distribution on local strain distribution. For the 3D-CPVT model, a set of CPFE analyses of micro-pillar

compression have been run and the effects of both regularity and grain size on deformation responses investigated.

Further to this, a multi-zone scheme is proposed for the CPVT models to generate virtual gradient grain structures. In conjunction with the CPVT model that controls the seed generating process within individual zones, the multi-zone CPVT model has been developed by incorporating a novel mechanism of controlling the seed generation for grains spanning different zones. This model has the flexibility of generating various gradient grain structures and the natural morphology for interfacial grains between adjacent zones. Both of the 2D- and 3D-CPVT models are capable of generating a virtual grain structure with a mean grain size gradient for the grain structure domain and grain size distribution control for individual zones. A true gradient grain structure, two simulated gradient grain structure, and a true gradient grain structure with an elongated zone have been used to examine the capability of the multi-zone CPVT model.

To facilitate the CPFE analyses of inter-granular crack initiation and evolution using the cohesive zone models, a Voronoi tessellation model with non-zero thickness cohesive zone representation was developed. A grain boundary offsetting algorithm is proposed to efficiently produce the cohesive boundaries for a Voronoi tessellation. The most challenging issue of automatically meshing multiple junctions with quadrilateral elements has been resolved and a rule-based method is presented to perform the automatically partitioning of cohesive zone junctions, including data representation, edge event processing and cut-trim operations. In order to demonstrate the novelty of the proposed cohesive zone modelling and junction partitioning schemes, the CPFE simulations of plane strain uniaxial tension and three point bending have been studied.

A software system, VGRAIN, was developed to implement the proposed virtual grain structure modelling methods. *Via* user-friendly interfaces and the well-organised functional modules a virtual grain structure can be automatically generated to a very large-scale with the desired grain morphology and grain size properties. As a pre-processing grain structure representation system, VGRAIN is also capable of defining crystallographic orientations and mechanical constants for a generated grain structure. A set of additional functions has also been developed for users to study a generated grain structure and verify the feasibility of the

generated case for their simulation requirements. A well-built grain structure model in VGRAIN can be easily exported into the commercial FE/CAE platform, e.g. ABAQUS and DEFORM, *via* script input, whereby the VGRAIN system is seamlessly integrated into CPFE modelling and simulation processing.

Acknowledgement

I am sincerely and heartily grateful to my supervisors, Dr Daniel Balint and Prof Jianguo Lin, for their continual help, support and guidance during the course of this project. Without their help this work would have not been possible. I must also thank my colleagues, in particular Shiwen Wang and Morad Karimpour, for their helpful advice and information.

I owe my deep thanks to Prof Jinping Li, who taught, helped and enlightened me during the tough times in my undergraduate years. I am truly indebted and thankful to Prof Guohui Tian and Prof Lei Jia, who guided, encouraged and supported me during my postgraduate years. In addition, I must give my sincere acknowledgement to Prof Lei Jia and Shandong University for their financial support, allowing me to conduct one year's research in the University of Birmingham, where I had a chance to meet my current supervisor.

I would like to take this opportunity to show my earnest gratitude to my parents for their love and supports throughout my life. Words can never express the depth of my appreciation to their love and efforts. I am grateful to my wife for her assistance and encouragement; I owe her many holidays. I would like to extend my appreciation to all my family and friends, who have given me many unforgettable moments and plenty of memorable experiences.

This work was supported by a European Union research project on Multiscale Modelling for Multilayered Surface Systems (Project No: CP-FP 213600-2 M3-2S).

TABLE OF CONTENTS

ABSTRACT.....	II
ACKNOWLEDGEMENT	V
TABLE OF CONTENTS	VI
LIST OF FIGURES	XI
LIST OF TABLES	XVII
CHAPTER 1 INTRODUCTION	1
1.1 GRAIN STRUCTURE REPRESENTATION	1
1.2 OBJECTIVE OF THE WORK	4
1.3 THESIS ORGANISATION	5
CHAPTER 2 GRAIN STRUCTURE MODELLING	7
2.1 DIGITISED GRAIN STRUCTURES.....	8
2.1.1 2D microscopy.....	9
2.1.2 Destructive 3D microscopy.....	12
2.1.3 Non-destructive 3D microscopy	16
2.2 NUMERICAL MODELS.....	18
2.2.1 Phase-field method	19
2.2.2 Cellular Automata.....	21
2.2.3 Monte Carlo Potts model	22
2.3 TESSELLATION MODELS.....	23
2.3.1 Uniform tessellations	24
2.3.2 Random tessellations.....	28

2.3.3	<i>Summary and discussion</i>	35
CHAPTER 3	CONTROLLED POISSON VORONOI TESSELLATIONS	38
3.1	INTRODUCTION	38
3.2	THE MODELS	40
3.2.1	<i>Voronoi tessellations</i>	40
3.2.2	<i>Poisson Voronoi tessellation</i>	41
3.2.3	<i>Hard-core model</i>	42
3.2.4	<i>The CPVT models</i>	43
3.3	GRAIN SIZE DISTRIBUTION	46
3.4	PHYSICAL PARAMETERS.....	49
3.4.1	<i>Uniqueness</i>	49
3.4.2	<i>Estimation of S^*</i>	51
3.4.3	<i>Lower bound on Pr</i>	53
3.4.4	<i>Algorithm to solve the c value</i>	54
3.5	APPENDIX	55
3.5.1	<i>Proof of Lemma 1</i>	55
3.5.2	<i>Proof of Theorem 1</i>	57
3.5.3	<i>Proof of Lemma 2</i>	57
3.6	SUMMARY	60
CHAPTER 4	2D CPVT MODEL	62
4.1	INTRODUCTION	62
4.2	MODEL DEVELOPMENT	62

4.2.1	<i>Workpiece related input</i>	62
4.2.2	<i>Regularity and control parameter</i>	64
4.2.3	<i>Regularity and distribution parameter</i>	67
4.2.4	<i>Implementation</i>	69
4.3	MODEL VALIDATION	70
4.3.1	<i>Statistical analysis</i>	70
4.3.2	<i>Comparison of virtual and true grain structures</i>	74
4.4	CPFE APPLICATION	79
4.5	SUMMARY	85
CHAPTER 5	2D COHESIVE ZONE MODEL	87
5.1	MOTIVATION	87
5.2	VORONOI TESSELLATION WITH COHESIVE LAYER BOUNDARIES	92
5.2.1	<i>VT and VTclb</i>	93
5.2.2	<i>Grain boundary offsetting</i>	94
5.3	COHESIVE JUNCTION PARTITION	96
5.3.1	<i>Schemes</i>	97
5.3.2	<i>Algorithm</i>	98
5.4	CPFE APPLICATIONS	106
5.4.1	<i>System for Grain structure generation</i>	106
5.4.2	<i>Case studies</i>	108
5.5	SUMMARY	113
CHAPTER 6	3D CPVT MODELS	118

6.1	INTRODUCTION	118
6.2	MODEL DEVELOPMENT	119
6.2.1	<i>General considerations</i>	119
6.2.2	<i>Regularity and grain size distributions</i>	123
6.2.3	<i>Implementation</i>	127
6.3	CPFE APPLICATION.....	129
6.4	CONCLUSION	134
CHAPTER 7	ADVANCED MODELS	139
7.1	INTRODUCTION	139
7.2	MOTIVATIONS	139
7.3	THE MULTI-ZONE CPVT MODEL	142
7.3.1	<i>Multi-zone settings</i>	142
7.3.2	<i>Coupling multiple zones</i>	144
7.3.3	<i>Property of interface region</i>	147
7.4	CASE STUDIES	148
7.4.1	<i>Comparison of real grain structures</i>	148
7.4.2	<i>Comparison with a numerical model</i>	150
7.4.3	<i>Deformed structures</i>	153
7.5	SUMMARY	157
CHAPTER 8	DEVELOPMENT OF VGRAIN SYSTEM	158
8.1	INTRODUCTION	158
8.2	SYSTEM DEVELOPMENT	159

8.2.1	<i>System workflow</i>	159
8.2.2	<i>User interfaces</i>	162
8.2.3	<i>Code organisation</i>	163
8.3	USAGE EXAMPLE AND INTERFACE EXPLANATION.....	164
8.4	SUMMARY	170
CHAPTER 9	CONCLUSIONS AND FUTURE WORK	171
9.1	CONCLUSIONS	171
9.2	RECOMMENDATIONS FOR FUTURE WORK	174
APPENDIX A	SCRIPTS FOR ABAQUS.....	176
REFERENCE	209

LIST OF FIGURES

FIGURE 1.1. ILLUSTRATION OF AN INTEGRATED PROCESS FOR CPFE SIMULATIONS.....	2
FIGURE 2.1. ILLUSTRATION OF A GRAIN STRUCTURE GENERATION PROCESS BASED ON A DIGITISED IMAGE ACQUIRED FROM A CCD CAMERA (CF. [24]). (A) A DIGITISED IMAGE; (B) THE CORRESPONDING BINARY IMAGE; (C) THE GRAIN STRUCTURE WITH SKELETONISED BOUNDARIES.	9
FIGURE 2.2. ILLUSTRATION OF THE GENERATION OF A GRAIN STRUCTURE REPRESENTATION FOR THE FE MODEL. (A) MICROSCOPIC IMAGE OF THE MICROSTRUCTURE OF Al_2O_3 ; (B) GRAIN BOUNDARIES AFTER IMAGE PROCESSING AND SELECTED MESH SEEDS; (C) FE MESH BY TRIANGULAR ELEMENTS.	10
FIGURE 2.3. SCHEMATIC OF A 3D GRAIN STRUCTURE RECONSTRUCTION PROCESS USING A SERIAL SECTIONING EXPERIMENT (REF. [48]). (A) IMAGES TAKEN FROM THE SECTIONED SURFACES; (B) CONSTRUCTION OF AN INDIVIDUAL OBJECT; (C) ENSEMBLE OF GRAINS.	13
FIGURE 2.4. A 3D RECONSTRUCTION OF AN AL6XN STEEL MICROSTRUCTURE, OBTAINED USING OPTICAL MICROSCOPY, SEM/EBSD AND SERIAL SECTIONING TECHNIQUES (REF. [57]). (A) 3D RECONSTRUCTION OF AUSTENITE PHASE IN AL-6XN MICROSTRUCTURE. THE COLOUR KEY CORRESPONDS TO THE CRYSTALLOGRAPHIC ORIENTATION IN THE DIRECTION OF THE ARROW, WHICH IS ALSO THE NORMAL TO THE SECTIONING PLANE, (B) THE IMAGE-BASED 3D FINITE ELEMENT MESH.....	16
FIGURE 2.5. EXAMPLE OF A RECONSTRUCTED GRAIN MICROSTRUCTURE DETERMINED BY DCT FOR FE MODEL REPRESENTATION [70]. (A) THE GRAIN STRUCTURE OF A β -TITANIUM ALLOY AND A FE MESH; (B) FE SIMULATION RESULTS; (C) LEGEND CORRESPONDING TO THE VON MISES STRESS DISTRIBUTION OF (B) (MPa).	17
FIGURE 2.6. EXAMPLE OF MICROSTRUCTURAL EVOLUTION OF GRAIN GROWTH USING THE PHASE-FIELD MODEL PERFORMED ON A $180 \times 180 \times 180$ SIMPLE-CUBIC GRID (REF. [80]). (A) A GRAIN STRUCTURE WITH 5954 GRAINS; (B) THE GRAIN STRUCTURE AFTER GROWTH CONTAINING 514 GRAINS.	20
FIGURE 2.7. MICROSTRUCTURE EVOLUTION SIMULATED USING THE CA MODEL [87]. (A) THE INITIAL GRAIN STRUCTURE; (B) A SIMULATED GRAIN STRUCTURE AFTER GRAIN GROWTH.	21
FIGURE 2.8. ILLUSTRATION OF A GRAIN STRUCTURE REPRESENTED BY A HEXAGONAL LATTICE DOMAIN, WHERE EACH GRAIN IS A CLUSTER OF HEXAGONS OF IDENTICAL INDEX.	23
FIGURE 2.9. ILLUSTRATION OF THREE-DIMENSIONAL LATTICES AND THEIR ASSOCIATED VORONOI POLYHEDRA. (A) SC LATTICE, WHERE GRAINS ARE REPRESENTED BY CUBES; (B) BCC LATTICE, WHERE GRAINS ARE TRUNCATED OCTAHEDRONS; (C) FCC LATTICE, WHERE THE GRAIN SHAPE IS A RHOMBIC DODECAHEDRON.	25

FIGURE 2.10. SPACE-FILLING POLYGONS AND POLYHEDRA. (A) EQUILATERAL TRIANGLE CLUSTER; (B) SQUARE CLUSTER; (C) REGULAR HEXAGONAL CLUSTER; (D) CUBE; (E) HEXAGONAL PRISM; (F) RHOMBIC DODECAHEDRON; (G) TRUNCATED OCTAHEDRON.	26
FIGURE 2.11. SCHEMATICS OF GRAIN BOUNDARY FORMATION UNDER THE ASSUMPTION OF ISOTROPIC GROWTH, WHERE THIN CIRCLES ARE PROGRESSIVE BOUNDARIES BY CONSTANT TIME INTERVALS, THICK LINES REPRESENTS GRAIN BOUNDARIES AND CENTRE NODES SIGNIFY NUCLEI (SEEDS). (A) SIMULTANEOUS NUCLEATION AND GLOBALLY-IDENTICAL CONSTANT GROWTH VELOCITY; (B) SEQUENTIAL NUCLEATION AND GLOBALLY-IDENTICAL CONSTANT GROWTH VELOCITY; (C) SIMULTANEOUS NUCLEATION AND GLOBALLY-IDENTICAL TIME-DEPENDENT GROWTH VELOCITY; (D) SEQUENTIAL NUCLEATION AND GLOBALLY-IDENTICAL TIME-DEPENDENT GROWTH VELOCITY; (E) SEQUENTIAL NUCLEATION AND CONSTANT BUT INHOMOGENEOUS GRAIN GROWTH.	33
FIGURE 3.1. SCHEMATIC PATTERN OF A CONTROLLED POISSON VORONOI TESSELLATION, WHERE THE CONTROL PARAMETER IS THE MINIMUM SEED SPACING δ , AND THE SEED DISTANCE $dg, a = \delta$ AND $dg, s > \delta$, WHERE $s = b, c, \dots, f$	44
FIGURE 3.2. SCHEME OF THE CPVT MODEL FOR VIRTUAL GRAIN STRUCTURE GENERATION.	45
FIGURE 3.3. EXAMPLE OF A ONE-PARAMETER GAMMA DISTRIBUTION, WHERE THE LOWER AND UPPER BOUNDS OF THE INTEGRAL ARE $x_1=0.8$ AND $x_2=1.2$, RESPECTIVELY.	49
FIGURE 3.4. SCHEMATIC ILLUSTRATION OF $\varphi(c, \Delta)$ OVER THE INTERVAL OF $\Delta \in (0,1)$, WHERE $c \in [1, \infty)$ IS A CONSTANT VALUE.	53
FIGURE 4.1. ILLUSTRATION OF THE EFFECT OF EXTERNAL SEEDS ON THE NUMBER OF GRAINS. (A) A CASE ACCOUNTING FOR EXTERNAL SEEDS; (B) A CASE NEGLECTING EXTERNAL SEEDS.	63
FIGURE 4.2. VORONOI TESSELLATIONS WITH DIFFERENT REGULARITIES.	66
FIGURE 4.3. COMPARISON OF THE TWO DESCRIPTIVE MODELS, BOTH OF WHICH DESCRIBE THE RELATIONSHIP BETWEEN THE GRAIN AREA DISTRIBUTION PARAMETER c AND THE REGULARITY PARAMETER α . THE DOTTED LINE IS THE MODEL PROPOSED IN [157] AND [158], AND THE SOLID LINE IS THE IMPROVED MODEL: (A) GLOBAL QUALITY; (B) LOCAL QUALITY.	68
FIGURE 4.4. THE RELATIVE ERROR OF THE IDEAL REGULARITIES AND THOSE OF GENERATED VTs. ERROR BARS REPRESENT 95% CONFIDENCE INTERVALS.	72
FIGURE 4.5. GRAIN SIZE DISTRIBUTIONS OF VTs WITH DIFFERENT REGULARITIES AND NUMBERS OF GRAINS. ERROR BARS WITH STANDARD DEVIATION HIGHLIGHT THE DISPERSION OF FREQUENCY VARIATION OF LOCAL GRAIN DISTRIBUTIONS FROM GENERATED VTs.	76
FIGURE 4.6. CPFE SIMULATION FOR A REGULAR GRAIN STRUCTURE. (A) ORIGINAL MICROSCOPIC IMAGE (DOMAIN IS 846 MM \times 563 MM, WITH 100 MM SCALE SHOWN); (B) THE RESULT AFTER IMAGE PROCESSING WHERE THE GRAIN STRUCTURE WAS SEGMENTED INTO 181 GRAINS; (C) THE SAMPLED VIRTUAL GRAIN STRUCTURE; (D) THE GRAIN SIZE DISTRIBUTION OF THE VIRTUAL STRUCTURE (A HISTOGRAM OF THE VIRTUAL STRUCTURE IS PLOTTED BY SHADED BARS, AND THE IDEAL DISTRIBUTION GIVEN BY THE REMAINING BARS); (E) A FE MODEL WITH ASSIGNMENT OF GRAIN ORIENTATIONS; (F) THE FE MODEL AFTER MESHING.	77

FIGURE 4.7. CPFE SIMULATION FOR AN IRREGULAR GRAIN STRUCTURE. (A) ORIGINAL MICROSCOPIC IMAGE (DOMAIN IS 1000 MM × 730 MM, WITH 100 MM SCALE SHOWN); (B) THE RESULT AFTER IMAGE PROCESSING WHERE THE GRAIN STRUCTURE WAS SEGMENTED INTO 179 GRAINS; (C) THE SAMPLED VIRTUAL GRAIN STRUCTURE; (D) THE GRAIN SIZE DISTRIBUTION OF THE VIRTUAL STRUCTURE (A HISTOGRAM OF THE VIRTUAL STRUCTURE IS PLOTTED BY SHADED BARS, AND THE IDEAL DISTRIBUTION GIVEN BY THE REMAINING BARS); (E) A FE MODEL WITH ASSIGNMENT OF GRAIN ORIENTATIONS; (F) THE FE MODEL AFTER MESHING.	78
FIGURE 4.8. THE PROCEDURE FOR GENERATING A CPFE MODEL.	79
FIGURE 4.9. SCHEMATIC DIAGRAM FOR THE PLANE STRAIN CPFE MODEL.	80
FIGURE 4.10. CPFE SIMULATIONS FOR AN IRREGULAR GRAIN STRUCTURE.	83
FIGURE 4.11. CPFE SIMULATIONS FOR A REGULAR GRAIN STRUCTURE.	83
FIGURE 4.12. COMPARISON OF LOCAL DEFORMATION FOR BOTH GRAIN STRUCTURES.	84
FIGURE 5.1. AN EXAMPLE OF THE VTCLB SCHEME AND SPECIFIC CASES OF COHESIVE JUNCTIONS.	91
FIGURE 5.2. ILLUSTRATION OF OFFSETTING GRAIN BOUNDARIES TO REPRESENT COHESIVE LAYERS. (A) A GRAIN STRUCTURE BASED ON THE VT REPRESENTATION; (B) A VTCLB, WITH A DEGENERATE GRAIN SHOWN WITH ONE OF THE ORIGINAL BOUNDARIES MISSING AFTER THE OFFSETTING PROCEDURE.	93
FIGURE 5.3. ILLUSTRATION OF THE STRUCTURED OFFSET METHOD. (A) GENERATING THE MEDIAL AXIS; (B) DECOMPOSING THE GRAIN INTO A SET OF SUB-POLYGONS; (C) STRUCTURED OFFSET BOUNDARIES; (D) FORMATION OF AN INWARD-OFFSET GRAIN.	95
FIGURE 5.4. THE ALGORITHM TO GENERATE THE VTCLB STRUCTURE FOR A GIVEN VT.	96
FIGURE 5.5. ILLUSTRATION OF TWO JUNCTION PARTITION PATTERNS FOR MESHING COHESIVE ZONES.	97
FIGURE 5.6. ILLUSTRATION OF A COHESIVE LAYER UNIT IN A VTCLB.	99
FIGURE 5.7. ILLUSTRATION OF MESHING RULES.	104
FIGURE 5.8. EXAMPLE CASES OF MESHING OPERATIONS.	105
FIGURE 5.9. EXAMPLE CASES OF MESHING CZ LAYER UNITS WITH SHORT INTERFACE BOUNDARIES.	105
FIGURE 5.10. ILLUSTRATION OF A GRAIN STRUCTURE GENERATION AND CPFE MODEL DEFINITION PROCESS.	107
FIGURE 5.11. SCHEMATIC DIAGRAMS OF THE PLANE STRAIN MODEL FOR CPFE SIMULATION.	109
FIGURE 5.12. THE STRESS-STRAIN CURVES CORRESPONDING TO THE TWO DIFFERENT RATIOS OF THE NORMAL TO TANGENTIAL DAMAGE INITIATION STRESSES.	112

FIGURE 5.13. PLANE STRAIN UNIAXIAL TENSION CPFE SIMULATION WITH $tn_0/tt_0 \rightarrow 1$. (A) CRACK INITIATION; (B) COMPLETE FAILURE; (C) GRAIN BOUNDARY TRIPLE POINT BEFORE AND AFTER THE CRACK PASSING THROUGH IT, WHERE L1 - L5 ARE COHESIVE LAYER UNITS.	115
FIGURE 5.14. PLANE STRAIN UNIAXIAL TENSION CPFE SIMULATION WITH $tn_0/tt_0 \rightarrow \infty$. (A) CRACK INITIATION; (B) COMPLETE FAILURE; (C) GRAIN BOUNDARY TRIPLE POINT BEFORE AND AFTER THE CRACK PASSING THROUGH IT, WHERE L1 - L5 ARE COHESIVE LAYER UNITS.	116
FIGURE 5.15. SCHEMATIC DIAGRAMS OF THE THREE-POINT BENDING MODEL FOR CPFE SIMULATION.	117
FIGURE 6.1. VORONOI TESSELLATIONS WITH DIFFERENT DEGREES OF UNIFORMITY. (A) AN IRREGULAR STRUCTURE OF A PVT TYPE; (B) A REGULAR STRUCTURE OF A NON-PVT TYPE.	120
FIGURE 6.2. DEFINITION OF THE REGULAR DISTANCE d_{REG} , WHERE s_1 AND s_2 ARE SEEDS TAKEN FROM A BCC LATTICE.	120
FIGURE 6.3. COMPARISON OF THE STATISTICAL DATA (BARS) AND THE MODEL FITS (POINTS) FOR DIFFERENT REGULARITY VALUES.	126
FIGURE 6.4. THE DESCRIPTIVE MODEL, RELATING THE REGULARITY PARAMETER c TO THE DISTRIBUTION PARAMETER α OF A ONE-PARAMETER GAMMA DISTRIBUTION FUNCTION.	127
FIGURE 6.5. SCHEMATIC DIAGRAM FOR THE CPFE SIMULATION OF MICRO-PILLAR COMPRESSION.	130
FIGURE 6.6. CPFE SIMULATIONS OF MICRO-PILLAR COMPRESSION FOR REGULAR AND IRREGULAR GRAIN STRUCTURES.	136
FIGURE 6.7. CPFE SIMULATIONS OF MICRO-PILLAR COMPRESSION FOR GRAIN STRUCTURES WITH DIFFERENT GRAIN SIZES.	137
FIGURE 6.8. THE FORCE – DISPLACEMENT PROPERTIES DURING COMPRESSING THE MICRO-PILLARS WITH DIFFERENT REGULARITIES.	138
FIGURE 6.9. THE FORCE – DISPLACEMENT PROPERTIES DURING COMPRESSING THE MICRO-PILLARS WITH DIFFERENT GRAIN SIZES.	138
FIGURE 7.1. A GRADIENT GRAIN STRUCTURE FORMED BY PROCESSING TI-BASED BMGS VIA SURFACE MECHANICAL ATTRITION TREATMENT (SMAT) (C.F. [201] FOR DETAILS). (A) THE SPATIAL DISTRIBUTION OF GRADIENT MEAN GRAIN SIZES; (B) THE THREE MICROGRAPHS TAKEN FROM THE SURFACE TO THE DEEP MATRIX.	140
FIGURE 7.2. A GRAIN STRUCTURE OF WELD JOINT OF AN AA 5754 FSW WELD ZONE, WHERE FINE GRAINS IN THE MIDDLE AREA ARE A TRACE OF OXIDE INCLUSIONS FROM SURFACES OF A BUTTED INTERFACE [202].	141
FIGURE 7.3. ILLUSTRATION OF THE DOMAIN OF A GRADIENT STRUCTURE IN FIGURE 7.1 INTO THREE ZONES.	143
FIGURE 7.4. ILLUSTRATION OF THE DOMAIN OF A GRADIENT STRUCTURE IN FIGURE 7.2 INTO TWO ZONES.	144

FIGURE 7.5. COMPARISON OF THE CPVT MODEL WITH THE MULTI-ZONE CPVT MODEL. (A) A MULTI-ZONE GRAIN STRUCTURE GENERATED BASED ON THE CONTROL PARAMETERS $\{\delta I, \delta II\}$; (B) THE GRAIN STRUCTURE FORMED BASED ON THE SEED LATTICE IN ZONE I; (C) THE GRAIN STRUCTURE FORMED BASED ON THE SEED LATTICE IN ZONE II.	146
FIGURE 7.6. ILLUSTRATION OF GRAINS SPANNING TWO ADJACENT ZONES.	148
FIGURE 7.7. COMPARISON OF A GRAIN SIZE GRADIENT NANO-STRUCTURED Ni COATING WITH A VIRTUAL GRAIN STRUCTURE GENERATED USING THE MULTI-ZONE CPVT MODEL. (A) GRAIN STRUCTURES IN DIFFERENT ZONES OF THE COATING [205]; (B) A VIRTUAL GRADIENT GRAIN STRUCTURE GENERATED BY THE MULTI-ZONE CPVT MODEL.	150
FIGURE 7.8. COMPARISON OF 2D GRADIENT GRAIN STRUCTURES OBTAINED BY THE MC METHOD AND THE MULTI-ZONE CPVT MODEL RESPECTIVELY. (A) A GRAIN STRUCTURE USING THE MC METHOD [206]; (B) A VIRTUAL GRADIENT GRAIN STRUCTURE GENERATED BY THE MULTI-ZONE CPVT MODEL.	152
FIGURE 7.9. COMPARISON OF 3D GRADIENT GRAIN STRUCTURES OBTAINED BY THE MC METHOD AND THE MULTI-ZONE CPVT MODEL RESPECTIVELY. (A) A GRAIN STRUCTURE USING THE MC METHOD [206]; (B) A VIRTUAL GRADIENT GRAIN STRUCTURE GENERATED BY THE MULTI-ZONE CPVT MODEL.	153
FIGURE 7.10. ILLUSTRATION OF TRANSFORMING AN EQUIAXED GRAIN TO (A) A VORONOI TESSELLATION; (B) ELONGATED GRAINS.	154
FIGURE 7.11. ILLUSTRATION OF A MULTI-ZONE GRAIN STRUCTURE, IN WHICH GRAINS IN DIFFERENT ZONES HAVE DIFFERENT ASPECT RATIO. (A) A GRAIN STRUCTURE OF THERMALLY GROWN OXIDE [207]; (B) A VIRTUAL GRADIENT GRAIN STRUCTURE GENERATED BY THE MULTI-ZONE CPVT MODEL.	156
FIGURE 8.1. ILLUSTRATION OF A MICROMECHANICS SIMULATION PROCESS EMPLOYING THE VGRAIN SYSTEM.	158
FIGURE 8.2. FLOW CHART OF THE VGRAIN SYSTEM, SHOWING THE RELATIONS OF THE MODULES AND THEIR CORRESPONDING FUNCTIONALITIES.	161
FIGURE 8.3. THE APPLICATION WINDOW OF VGRAIN.	163
FIGURE 8.4. ILLUSTRATION OF THE INTERNAL DATA FLOWS AND COMMAND FLOWS.	164
FIGURE 8.5. USER INTERFACE FOR DEFINITION OF A 2D MULTI-ZONE CPVT MODEL AND THE VISUALISATION OF A RESULTANT VIRTUAL GRAIN STRUCTURE.	166
FIGURE 8.6. USER INTERFACES FOR DEFINITION OF A CPFE MODEL BASED ON A GENERATED GRAIN STRUCTURE. (A) THE CRYSTALLOGRAPHIC ORIENTATION GENERATOR; (B) ASSIGNMENT OF MATERIAL PROPERTIES; (C) THE RESULTANT VIRTUAL GRAIN STRUCTURE, WHERE COLOURS RELATE TO GRAIN ORIENTATIONS.	167
FIGURE 8.7. ILLUSTRATION OF THE GENERATION OF COHESIVE ZONES AND JUNCTION PARTITIONING RESULTS.	168

FIGURE 8.8. ILLUSTRATION OF THE INPUT OF A CPFE MODEL FROM VGRAIN TO ABAQUS.	168
FIGURE 8.9. ILLUSTRATION OF THE SUB-MODULES IN THE VGRAIN SYSTEM FOR ANALYSING MICROSTRUCTURE PROPERTIES. (A) ACCESSING RAW GEOMETRICAL DATA; (B) COMPARING THE GRAIN SIZE DISTRIBUTIONS BETWEEN USER-INPUT PHYSICAL PARAMETERS AND THE GENERATED STRUCTURES; (C) STUDYING THE CRYSTALLOGRAPHIC TEXTURE.	169
FIGURE 9.1. SCHEMATIC DIAGRAM SHOWING THE METHODOLOGY OF A CPVT MODEL.....	172

LIST OF TABLES

TABLE 2.1. PROPERTIES OF THE GEOMETRICAL ELEMENTS OF DIFFERENT GRAIN SHAPES.	26
TABLE 3.1. ASYMPTOTIC BEHAVIOUR OF $\varphi c, \Delta\Delta = 0.999$	53
TABLE 4.1. STATISTICAL RESULTS FOR THE MEAN VALUE OF RELATIVE ERROR ε AND THE STANDARD ERROR, WHERE THE MEAN GRAIN SIZE FOR EACH VT WAS CHOSEN TO BE 10 MM ²	72
TABLE 4.2. PHYSICAL PARAMETERS AND CORRESPONDING GRAIN STRUCTURE PROPERTIES FOR THE TWO GRAIN STRUCTURES.....	74
TABLE 4.3. PHYSICAL PARAMETERS AND CORRESPONDING GRAIN STRUCTURE PROPERTIES FOR THE TWO CPFE MODELS.....	81
TABLE 5.1. SUMMARY OF THE SIX TYPES OF ELEMENTARY COHESIVE LAYER UNIT. (THICK LINES REPRESENT THE BOUNDARIES OF A VTCLB, DOT-LINES ARE THE ORIGINAL GRAIN BOUNDARIES IN THE CORRESPONDING VT, AND NORMAL LINES ARE JUNCTION PARTITIONS.)	100
TABLE 5.2. SPECIAL CASE ENDS OF A COHESIVE LAYER UNIT.	101
TABLE 5.3. PHYSICAL PARAMETERS AND CORRESPONDING GRAIN STRUCTURE PROPERTIES FOR THE TWO CPFE MODELS.....	109
TABLE 6.1. THE FITTING RESULTS INCLUDING THE RELATIONS BETWEEN THE REGULARITY A AND THE DISTRIBUTION PARAMETER C, AND THE CORRESPONDING FITTING ERRORS DEFINED IN EQ. 6.6.	125
TABLE 6.2. THE VALUES OF MATERIAL CONSTANTS FOR THE CRYSTAL-PLASTICITY CONSTITUTIVE EQUATIONS IN EQS. (6.8)-(6.16).....	132
TABLE 6.3. PHYSICAL PARAMETERS AND CORRESPONDING GRAIN STRUCTURE PROPERTIES FOR THE TWO MICROSTRUCTURES.....	132
TABLE 7.1. PHYSICAL PARAMETERS FOR GRAIN STRUCTURE I AND II IN FIGURE 7.5.....	146
TABLE 7.2. PARAMETERS FOR THE MULTI-ZONE CPVT.....	149
TABLE 7.3. PARAMETERS FOR THE MULTI-ZONE CPVT TO GENERATE A 2D VIRTUAL GRAIN STRUCTURE OF A HAZ.....	151
TABLE 7.4. PARAMETERS FOR THE MULTI-ZONE CPVT TO GENERATE A 3D VIRTUAL GRAIN STRUCTURE OF A HAZ.....	151
TABLE 7.5. PARAMETERS FOR THE MULTI-ZONE CPVT TO GENERATE A 2D VIRTUAL GRAIN STRUCTURE, CONTAINING AN ELONGATED ZONE. THE PARAMETERS FOR THE ELONGATED ZONE DESCRIBE THE PROPERTIES OF THE UNDEFORMED GRAIN STRUCTURE, AND THE FINAL SHAPE IS DETERMINED IN COMBINATION WITH THE SPECIFIED ρ	155

Chapter 1

Introduction

1.1 Grain structure representation

The development towards miniaturisation of products and devices in industries such as electronics, optics, medical devices, and communications, has increased the demand for metallic parts manufactured at extremely small scale, in which some characteristic dimensions of the micro-parts are of the order of ~ 1 mm or less. Examples of micro-parts include screws, fasteners, connector pins, springs, micro-gears and micro-shafts. A variety of manufacturing processes have been employed such as machining, folding, bending, stamping, drawing and forward/backward extrusion [1, 2]. At this level, grain size of materials is comparable to that of the part being formed and the homogeneity assumption of materials in conventional metal forming processes is inappropriate. In such a context, the material anisotropy dominates the inhomogeneous deformation and mechanical properties, e.g. strength, strain-hardening, ductility and damage.

In a polycrystalline microstructure, the mechanical anisotropy mainly originates from crystallographic orientation, grain size and morphology, and their evolutions during deformation. In order to fully understand the mechanical anisotropy of a polycrystalline material and accurately predict its deformation behaviour, an explicit grain structure including crystallographic orientation and grain size, shape and organisation must be incorporated. Crystal plasticity theory is a classical physically-based method to predict the elastic and plastic deformation of polycrystalline materials at the grain level. Plastic deformation is assumed to be solely due to crystallographic dislocation slip [3-5]. The flow of dislocations along slip systems is based upon continuum plastic shear strains without immediate representation of the dynamics of the lattice and its defects. However, this theory was developed neither explicitly considering the mechanical interactions of grains in a polycrystal aggregate nor responding to complex internal or

external boundary conditions. This weakness allows the finite element methods to be involved as a vital companion to study the interactions within individual grains and between adjacent grains. These computational methods, known as the crystal plasticity finite element (CPFE) models, involve an explicit representation of a grain structure and the direct utilisation of the crystal plasticity theory to describe a constitutive behaviour of individual grains [6]. In the last decade CPFE methods have evolved as an extremely versatile tool to simulate the mechanical response of crystalline materials on all length scales from single crystals to engineering parts. The involved constitutive laws range from the empirical viscoplastic formulations [7, 5, 8] to physically-based multiscale internal-variable models [9, 10] (cf. [11] for a detailed review). In addition, the CPFE methods are capable of studying the mechanical deformation of the polycrystalline structure under complicated internal and/or external boundary conditions. This is not only essential to computing the inter-grain and sub-grain responses, but also to tackling the abrupt mechanical transitions during crack initiation and propagation.

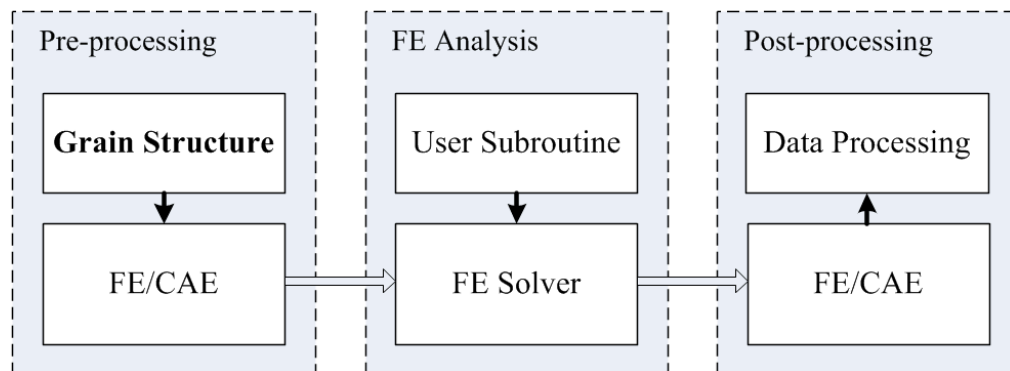


Figure 1.1. Illustration of an integrated process for CPFE simulations.

In recent years, CPFE methods have increasingly gained momentum in both small-scale and large-scale applications. For small-scale applications, they have been applied to understand the inter- and intra-grain mechanics, prediction of local damage processes and simulation of micromechanical experiments (e.g. micro-pillar compression and nano-indentation). Since stress-strain distributions are related to grain size, shape, orientation and their distributions [12-15], CPFE simulations must be based upon a grain structure

modelled within a finite element computer aided engineering (FE/CAE) environment, as shown in Figure 1.1. Explicit modelling of the grain structure is also essential to capture a range of size-dependent features of micromechanical deformation [16-18]. Besides this, in heterogeneous polycrystalline materials, crack initiation and propagation are not dominated by the mean values of stress and strain but by local maximal values around grain boundaries or interfaces [19]. By presenting a realistic grain structure the prediction of deformation and local damage evolution will correspondingly benefit.

For large-scale applications, CPFE methods have been widely used to simulate micro-forming processes and predict characteristics of a micro-part's shape, failure of materials, spring-back effects and mechanical properties of the final parts. To give a quality prediction or simulation, the large-scale simulations require a high-fidelity representation of virtual grain structure, which includes both crystallographic structure and geometrical structure, into a finite element (FE) model. Different from the small-scale simulations, in a large-scale CPFE simulation, a virtual grain structure contains a large number of grains to be represented in a FE model. Due to the number of grains, a grain structure shows a particular uniformity and grains exhibit a grain size distribution characteristic. As rigorous studies have revealed, not only the mean grain size but also the grain size distribution have significant influence on not just the macroscopic plastic flow stress and local strain-stress development [15], but also the mechanical properties such as creep rate [20], crack propagation rate [21], and creep damage evolution [13] which are all strongly affected by the grain size distribution characteristics. Therefore, there is an increasing demand for realistic virtual grain structures based upon simulation requirements for accurate prediction and studies. Note that, methods for large-scale grain structure representation must be capable of generating realistic or statistically equivalent structures effectively and efficiently at a large scale.

Voronoi tessellation models are one class of the most efficient grain structure generation methods that have been widely used for metallurgical applications [22, 23], as they provide a natural solution to represent grain structures with non-uniform grain shapes. Formation of a Voronoi tessellation can be interpreted as the result of a grain aggregate growth process, where grain seeds are simultaneously nucleated, grain growth

is isotropic and growth velocity is identical for all grains. However, as a Voronoi tessellation is disordered in terms of grain morphology, grain size and distribution, there is still a lack of effective methodology for the Voronoi tessellation models to generate virtual grain structures with the desired structural properties, e.g. a desired grain size distribution to meet the simulation requirements.

1.2 Objective of the work

The primary goal of this work is the development of a computational model that automatically generates Voronoi tessellations as virtual grain structure representation with the properties of mean grain size control and the grain size distribution control. Physical parameters are used to determine a grain size distribution of a grain structure and the structure's uniformity, by which a control parameter can be derived for the generation of a virtual grain structure. The major advantage is that a virtual grain structure generated using this model is statistically equivalent to the set of user-specified physical properties in terms of grain size distribution.

Based on this computational framework, a range of specialised models are further developed to generate a wide range of virtual grain structures including:

1. Two- and three-dimensional virtual grain structures with grain size distribution control.
2. Two- and three-dimensional gradient grain structures with mean grain size gradient control across the whole domain and grain size distribution control for individual sub-regions.
3. Two-dimensional virtual grain structures with non-zero thickness cohesive zone representation and automatic junction partitioning.

A novel pre-processing software system, VGRAIN, is further developed to implement the proposed virtual grain structure generation models. As an outstanding companion for large-scale CPFE simulations, especially micro-forming processing analyses, VGRAIN is capable of generating a complete grain structure model including grain structure,

crystallographic and mechanical properties of grains. Characteristics of a generated grain structure model can be examined and analysed by VGRAIN itself and the generated model can then be exported from VGRAIN to commercial FE/CAE platforms *via* script input.

1.3 Thesis organisation

In the following chapter, a comprehensive review on current grain structure modelling techniques and related models is presented. In Chapter 3, the controlled Poisson Voronoi tessellation (CPVT) model is introduced. A rigorous study is conducted to establish the mechanism of using the physical parameters from a given grain structure to define its grain size distribution. A mathematical proof of the uniqueness of the determination of the distribution parameter from the physical parameters is provided, and an efficient numerical procedure is proposed for computing the distribution parameter. Chapter 4 specialises the general CPVT model to a two-dimensional implementation. A series of studies are carried out to validate the accuracy of grain size distribution control and a grain structure's regularity control. Implementation of this model is demonstrated *via* a set of CPFE simulations. In Chapter 5, a novel scheme for the generation of non-zero cohesive zone representation for grain boundaries and automatic cohesive zone junction partitioning are developed. A set of CPFE simulations shows the novelty and feasibility of the proposed scheme in simulating grain boundary sliding and separation. Chapter 6 specialises the general CPVT model to a three-dimensional implementation. A group of comprehensive statistical experiments is performed to derive the relation between the distribution parameter and the grain structure's regularity. A set of CPFE simulations of micro-pillar compression are conducted to demonstrate the implementation of the 3D-CPVT model for micromechanics studies, relating to the effects of mean grain size and grain size distribution. In Chapter 7, an application of the proposed CPVT model to generation of advanced microstructural features is given. Emphasis is devoted to generating gradient virtual grain structure, where mean grain size characteristics vary across a grain structure domain. A multi-zone scheme is introduced and the multi-zone CPVT model is developed.

In Chapter 8, a novel virtual grain structure modelling software system is detailed, which implements all the proposed models. Finally, the conclusions of this study and suggestions for further work are presented in Chapter 9. An Appendix is included with example scripts for three virtual grain structure models generated by VGRAIN for ABAQUS/CAE modelling.

Chapter 2

Grain Structure Modelling

There are generally two major types of grain structure generation methods: *experimental techniques* and *numerical models*. The experimentally-determined grain structures can be achieved by processing the digital images acquired using a wide variety of metallographic microscopic techniques, such as optical or electron microscopes and electron backscatter diffraction (EBSD) techniques. The overriding advantage of using digital image based finite element analysis is the accuracy and high-fidelity in modelling the microstructure of a presented material. However, regardless of the technical difficulties, these methods are extremely time-consuming and expensive, especially for the three-dimensional grain structure reconstruction, and the achieved grain structures are also difficult to extrapolate to large specimens. Thus, it is difficult to apply the experimental approach to a large-scale CPFE simulation. Although experimental methods are beneficial in that they produce an exact representation of the microstructure of a material sample, 3D reconstruction using experimental methods may require destruction of the material sample, depending on the method used. Moreover, since experimental grain structure modelling must be based upon a present material, this is not always available during the design process for an optimised material.

Numerical models, on the other hand, are generally versatile and very flexible in generating virtual grain structures for a wide range of engineering applications. There have been many models developed in the field of grain structure modelling such as phase-field models, the Kinetic Monte Carlo Potts models, the cellular automata models and the Voronoi tessellation models. These models were initially developed to quantitatively investigate and predict microstructure evolution at the mesoscale (grain-level) based on physical or phenomenological mechanisms, which are idealised in a continuum formulation without explicit incorporation of atomic-scale dynamics. They are capable of simulation of a detailed grain boundary evolution at a realistic time cost and

for large-scale applications. Final grain structures simulated using these models can also be represented in FE models since they are obtained in a computational environment.

This chapter does not aim at a complete review of the large number of techniques and numerical models relating to grain structure modelling, but to concentrate on the most typical methods that have been applied to, or can be potentially incorporated into, a grain structure representation model for CPFE simulations. The following section focuses on a few major experimental techniques and systems that have been used to generate true grain structures. Furthermore, four types of numerical models from the field of computational mesoscale materials simulations are reviewed, namely the phase-field models, the Kinetic Monte Carlo Potts models, the cellular automata models and the Voronoi tessellation models. In addition, the feasibility of these models in representing grain structures for micromechanics simulations is addressed. Furthermore, a class of special numerical models, namely the geometrical tessellation models, are summarised. Their merits and the potential for being developed into a class of application-oriented virtual grain structure generation methods are addressed.

2.1 Digitised grain structures

Traditional quantitative metallographic techniques are utilised to characterise the size, shape, and distribution of polycrystalline grains, and some of the techniques are also capable of deriving crystallographic information from grains. By revealing these key structural features, which are linked with the constituents and the material processing history, the material mechanical properties such as hardness, strength, crack resistance, and ductility can be explained empirically. With the advent of micromechanics models, empirical studies of grain structural properties have been advanced to sophisticated numerical simulations. Inherently, traditional metallographic techniques are still essential to acquire digital images of real grain structures, which can be further incorporated into material models for micromechanical simulations. In the following, the major metallographic techniques and their applications in acquisition of digital images and applications to CPFE simulations are reviewed.

2.1.1 2D microscopy

In conventional metallographic analyses, there is a range of microscope tools available, using the transmission or reflection of either light or electrons to examine the microstructure properties over a wide span of length scales or magnification levels. One of the most commonly used tools to observe the grain morphological characteristics is the optical microscope with the ability to resolve the length of $\sim 0.2\text{ }\mu\text{m}$ or more. In combination with an etching process, planar grain structures of a polished material surface with recognisable grain boundaries can be observed. Henceforth, the morphological structure of an observed area including individual grain shape and boundaries can be examined, and images can be digitised by means of charge-coupled device (CCD) cameras or scanned-in approaches. Since a digitised image simply contains microstructure morphological information, methods such as manually sketching or digital image processing can be applied to achieve the grain boundaries and hence build an explicit grain structure for subsequent micromechanics studies.

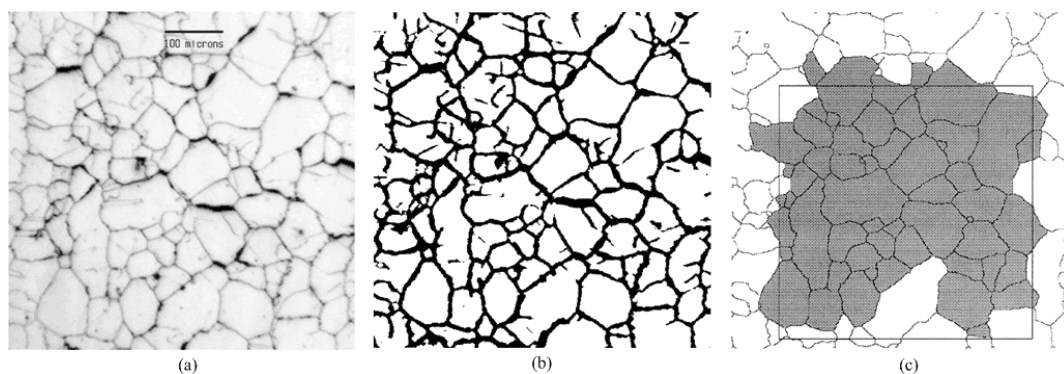


Figure 2.1. Illustration of a grain structure generation process based on a digitised image acquired from a CCD camera (cf. [24]). (a) A digitised image; (b) the corresponding binary image; (c) the grain structure with skeletonised boundaries.

For example, in [24] a QUANTIMET 570 image processing system was introduced to determine the grain morphological structure of austenitic steel, in which the surface of a polished and etched specimen was magnified by an optical system, and digital images of

an observed area were recorded by a CCD camera. After the acquisition of images, a digital image processing method proceeding with grey and binary image transformations can be applied to detect the grain boundaries. Figure 2.1 presents the results obtained from the successive operations. In addition to the binary operation, there have been other advanced mathematical morphology algorithms [25] employed to identify grain boundaries and other morphological features. For example, in [26] the watershed transformation was used to detect the grain morphological structure of a LiF thin film. Note that, after image processing, a clearly presented grain morphological structure shown in Figure 2.1 (c), is ready to be combined into a FE model. This is an early technique used to represent grain structures for micromechanics models [27, 28]. For example, a micromechanical FE study was carried out to compute the elastic behaviour of a polycrystalline thin film of aluminium oxide (Al_2O_3) in [27]. The digitised image of a micrograph was processed using the LEICA QWin image processing package to obtain the boundaries. Furthermore, boundary curves were represented by a set of selected points shown in Figure 2.2 (b), which were subsequently employed as FE mesh seeds to generate the meshes as shown in Figure 2.2 (c).

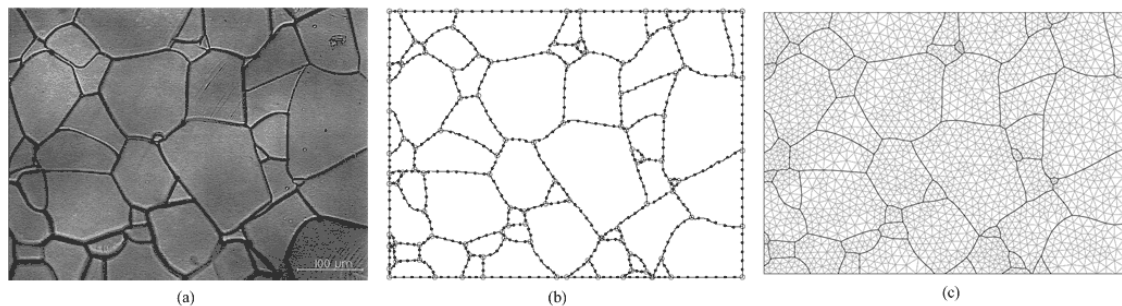


Figure 2.2. illustration of the generation of a grain structure representation for the FE model. (a) microscopic image of the microstructure of Al_2O_3 ; (b) grain boundaries after image processing and selected mesh seeds; (c) FE mesh by triangular elements.

The other commonly used metallographic tool is the scanning electron microscope (SEM). Compared to the optical microscope, the SEM expands the resolution length by more than a two-digit magnitude to approximately 4 nm in routine instruments, with ultimate values below 1 nm [29]. This magnification spans the entire range of the light microscope and covers much of the range of the transmission electron microscope (TEM). The SEM also provides a greater depth of field than the light microscope, allowing better discernment of morphological features during a microscopic investigation. Similar to the optical microscopy, the SEM can be directly used to magnify material cross-sections and detect the grain structures. Another important advantage from the SEM is its commonly built-in EBSD devices, which measure crystallographic orientations on a predefined regular grid. By automated acquisition and analysis of a backscatter diffraction pattern, the SEM/EBSD technique is capable of collecting morphological and crystallographic data simultaneously. An EBSD-based orientation microscopy allows quantitative characterisation of crystalline materials accurately, since a domain belonging to the same grain in the planar image can be easily identified by its crystallographic structure [30]. Hereby, a wealth of grain structural characteristics including crystallographic orientation, grain morphology and approximate grain boundary can be reconstructed in a sophisticated manner. This technique can also be applied to study microstructural evolution during a complex thermal-mechanical processing such as friction stir welding [31, 32]. As a by-product of microstructure characterisation using the SEM/EBSD technique, a resolved grain structure containing the grain boundaries and crystallographic orientations can be incorporated into the representation of a FE model. Generally, there are two feasible ways to formulate the grain geometry of a FE model from an EBSD image: 1) *Direct mapping*, which maps the geometry and orientation of the digital image's pixels to the mesh of the FE model, and 2) *indirect sketching*, which smoothes the grain boundary pixels by geometrically interpolated lines or curves. Note that, smoothing the pixel-based boundaries into the lines or curves based grain structure is more flexible and practical for FE/CAE meshing. Therefore, the indirect mapping is generally more preferable in 2D grain structure representation. In [33, 34], grain structures from a coarse grained aluminium sample were experimentally determined for CPFE simulations. The digital images were taken from the SEM and their crystal

orientation maps were obtained from EBSD patterns. Grain boundaries were sketched with lines and meshed using bilinear elements. This scheme allows one-to-one comparison between experiments and CPFE simulations with respect to the mechanical deformation properties of a thin film (a material with approximate columnar morphology along its thickness). Similarly, in [35] micrographs from SEM/EBSD observations on commercial purity aluminium were taken to represent the grain structure in an FE model simulating grain interactions and compared to experimental results obtained in [36]. Moreover, such comparisons can be used to validate numerical models and calibrate parameters of a model; for example the comparison study of strain localisation for a thin slice copper ingot sample under uniaxial tension in [37], the study on the influence of a subgrain misorientation distribution on the strain hardening behaviour in [38], and the evolution of surface roughening in [39].

2.1.2 Destructive 3D microscopy

The aforementioned metallographic techniques are commonly used to characterise two dimensional microstructures of a material cross-section, and are able to capture the digital images containing explicit morphological structures of grains, on which a finite element material model can be represented. In a recent development, these techniques have been further employed to record images of successive serial sections for the generation of three-dimensional grain structures.

2.1.2.1 Serial sectioning

Serial sectioning is a class of techniques applied to removal of slices of materials in sequence. The thickness of individual slice is primarily determined by the size-scale of the microstructural features of interest, for example the size of grains or other particles to be examined. The section thickness is often a compromise between the expectation of high-resolution data and the costs in collecting the data sets [40]. Approaches, such as mechanical polishing (MP) approaches or micro-milling devices [41-43], allow serial sectioning large material sample surfaces. However, when using this type of polishing it is difficult to consistently obtain removal of material with a thickness less than 1 μm . Recently, a novel ion-based micro-milling technique using focused ion beam-scanning

electron microscopes (FIB-SEM) has been used for serial sectioning [44-46]. The FIB-SEM can produce high precision milling and a much finer serial sectioning thickness than the MP approaches, but due to the limited milling speed, this technique can only be applied to a smaller volume. In conjunction with imaging devices, EBSD mapping can be coupled to the process of characterisation and reconstruction of grain structures, by which the actual crystallographic texture can be identified. Note that the acquisition of EBSD mapping data is also very slow, which significantly brings more workload to this time-consuming and laborious process.

In practice, the EBSD data is collected less frequently than the recording of images of an exposed material surface [47]. Moreover, after sectioning, especially in the MP approach, a material surface often suffers scratches. In order to accurately index EBSD patterns, additional processes are often required to minimise the surface damage. The collection of serially sectioned micrographs and EBSD mappings enable a 3D grain structure to be reconstructed *via* stacking the images, subsequent post-processing and analysing the data through feature segmentation and data clean-up [40]. Figure 2.3 schematically presents a 3D grain structure reconstruction process. The set of 2D images containing the EBSD maps were sequentially collected for every section, as illustrated in Figure 2.3. After post-processing and analysis, individual grains are identified, which can be further grouped together for reconstruction of the entire grain structure.

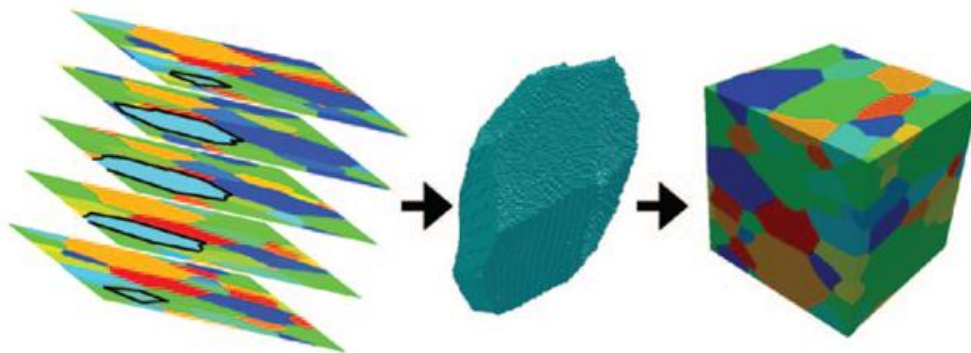


Figure 2.3. Schematic of a 3D grain structure reconstruction process using a serial sectioning experiment (Ref. [48]). (a) Images taken from the sectioned surfaces; (b) construction of an individual object; (c) ensemble of grains.

2.1.2.2 Related systems

The main challenges in performing serial sectioning are: controlling the sectioning depth, obtaining parallel flat surfaces, aligning the observation area and controlling contrast of many microscopic techniques. To avoid manually dealing with these issues, a range of automated 3D serial sectioning systems have been developed for characterisation and construction of 3D microstructures. The system in [42] consists of a micromiller (Reichert-Jung Polycut E micromiller), and etching-cleaning devices. In conjunction with the method of linear variable differential transformer (LVDT), a specimen can be automatically repositioned for photographing. In [43, 49], high-precision automated polishing equipment (Robo-Met.3D) has also been made to achieve high-precision automatic serial section and imaging. In a recent development, the fully automated 3D orientation microscopy systems by operating FIB and SEM/EBSD have been developed by Mulders and Day [50] and Zaefferer et al. [46]. These systems have significantly higher spatial resolution and more accurate image alignment functions than competing methods.

However, until recently, the milling and measurement processes for FIB-EBSD method are still very slow, from about 30 minutes to more than 1 hour per cycle [46], and the total procedure may take between 3 and 40 hours [50]. Since this method can only be used to investigate a relatively small volume, statistical results such as grain size distributions or textures cannot be sufficiently achieved. In contrast, the MP-EBSD method allows a larger observation domain, but it is extremely laborious, time-consuming and difficult to fully automate.

2.1.2.3 Digital image based FE simulations

Micromechanics studies, especially CPFE simulations, can take significant advantage of the experimentally reconstructed grain structures with true crystallographic information and morphological data. A high-fidelity 3D grain structure can be represented in a FE model allowing for rigorous analysis and scientific visualisation of the interactions between mechanical response and microstructural features, such as grain size, shape, grain boundary morphology and misorientation [51-55]. Like the 2D image-

based finite element model representation, there are mainly two methods to set up a FE model based on a reconstructed 3D grain structure; that is, by directly using the gridded experimental material data to represent the finite element mesh or by indirectly smoothing the voxel-like grain boundaries using an interpolation method such as NURBS surfaces.

In [56], a case of 3D reconstruction of an AL6XN steel microstructure was presented, which utilised serial sectioning, optical microscopy and automated EBSD techniques. In their method, a sectioning depth of 3.3 μm was used to achieve high-resolution reconstructions (i.e. 10-20 sections per grain). At each section, an individual image was obtained using optical microscopy for grain boundary recognition. Furthermore, at every 10th section, SEM/EBSD was used to measure the spatially resolved crystallographic orientation distribution. The reconstructed real microstructure was directly incorporated into a FE model to simulate mesoscale mechanical responses of the real microstructure [57]. The finite element mesh directly used the experimental material data, where the EBSD pattern of the planar structure and the successive sections divided a 3D grain structure domain into voxel data. In their simulation, the rectangular meshes consisted of a reduced-resolution experimental data with one element per voxel. Similarly, in [58], a 3D microstructure of the body centred cubic(BCC) single-phase beta-Ti alloy with 92 grains was also reconstructed and involved in simulating the initiation of plasticity using CPFE analysis.

The direct utilisation of experimentally-determined voxel data enables each grain to be represented by a collection of cubic elements with specific orientations. Note that, the mesh using a coarse grid significantly causes artificially aliased grain boundaries, while a fine grid of voxel elements leads to a large number of mesh elements in the grain structure domain, hence is computationally costly in the form of both time and memory. In contrast, a geometrically smoothed boundary network is more preferable in a FE/CAE environment. In [59, 60], surface fitting methods using parametric polynomial interpolation and non-uniform rational B-Spline (NURBS) interpolation were applied to smooth the voxels at the interface of adjacent grains. With the functional representation of the grain boundary surfaces, it enabled flexible control of the local density for meshing

and was capable of dealing with various complex grain morphology and grain junctions. However, extra geometrical algorithms were needed to remedy various complicated special cases such as the gap and overlap removal during the construction of parametric grain boundary surfaces [59].

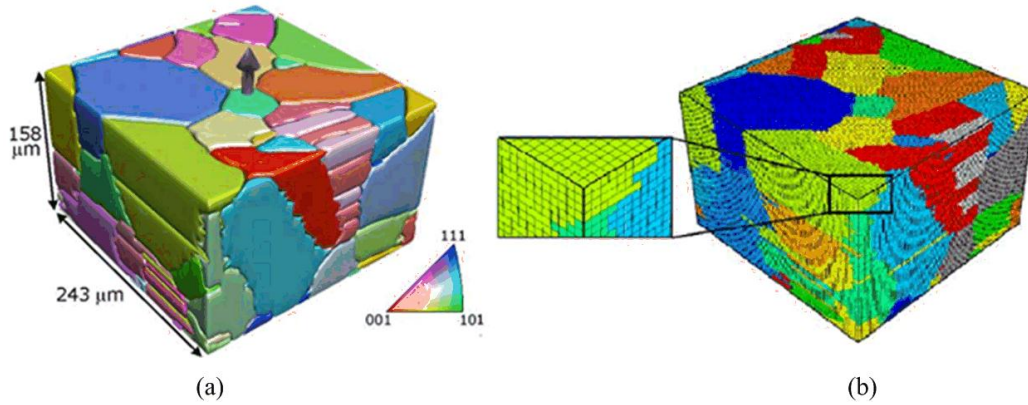


Figure 2.4. A 3D reconstruction of an AL6XN steel microstructure, obtained using optical microscopy, SEM/EBSD and serial sectioning techniques (Ref. [57]). (a) 3D reconstruction of austenite phase in AL-6XN microstructure. The colour key corresponds to the crystallographic orientation in the direction of the arrow, which is also the normal to the sectioning plane, (b) the image-based 3D finite element mesh.

2.1.3 Non-destructive 3D microscopy

Recently, a class of novel high energy X-Ray diffraction techniques, i.e. three-dimensional X-ray diffraction microscopy (3DXRD), has been developed for 3D structural characterisation and grain structure reconstruction of polycrystalline materials. This technique relies on highly penetrating hard X-rays from a synchrotron source with energies in the range of 50-100 keV [61] and applied tomographic reconstruction approaches for fast acquisition and analysis of the diffraction data [62]. It is capable of a fast and non-destructive characterization of the grain-level microstructure for specimens with the length scale from micron to centimetre. Grain morphology, organisation and crystallographic orientations can be derived for hundreds of elements simultaneously.

Moreover, various mechanical and material properties can be observed or derived using the synchrotron-based X-ray imaging and diffraction techniques. For example, the 3DXRD microscopy has been used to characterise the average stress state of each embedded grain [63], determine the three-dimensional plastic strain field [64] and the strain tensor [65]. In the most recent development, the evolution of the full stress tensor in a deformed twin was accurately determined from the 3DXRD data [66], and various algorithms, e.g. FitAllB [67], were proposed to simultaneously fit multiple grains with respect to position, orientation and stress state. Since these 3D imaging techniques are non-destructive they are versatile to study a large variety of time-dependent grain structure evolutions [68, 69].

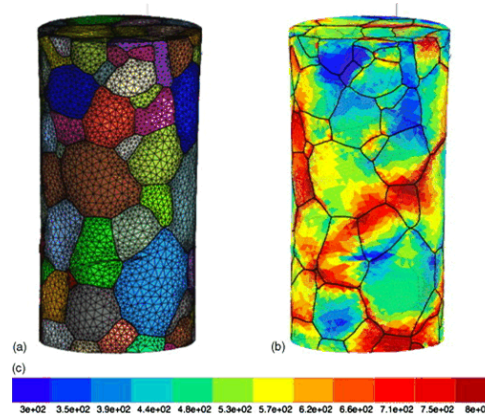


Figure 2.5. Example of a reconstructed grain microstructure determined by DCT for FE model representation [70]. (a) The grain structure of a β -Titanium alloy and a FE mesh; (b) FE simulation results; (c) legend corresponding to the von Mises stress distribution of (b) (MPa).

Among the present synchrotron-based X-ray 3D imaging techniques, the recently developed X-ray diffraction contrast tomography (DCT) offers high spatial resolution and accuracy [71] and provides the functionality of simultaneous access to 3D grain shape and crystallographic orientation distribution [72]. Since a 3D grain structure produced by the DCT contains a complete description of grain shape and crystallographic orientation, it can be used as input of material representation for CPFE simulations. Figure 2.5 (a)

presents a FE model based on a true grain structure of β -Ti alloy obtained by the DCT map; Figure 2.5 (b) shows the local stress distribution under a load of 500 MPa, where colour corresponds to stress values in the legend of Figure 2.5 (c). As mentioned above, the synchrotron-based X-ray 3D microscopy can be used to estimate the strain and stress state for individual grains. Therefore, using these non-destructive experimental techniques a direct comparison of in-situ deformation observation with the CPFЕ simulation is realistic and promising [70].

It should also be noted that whilst these synchrotron-based X-ray imaging and diffraction techniques provide a collection of promising approaches to generate true 3D grain structure and directly observe the evolutions of microstructure and mechanical deformation, the high-intensity synchrotron sources are not generally accessible. This limitation essentially constrains the applications of this class of techniques for general-purpose uses.

2.2 Numerical models

The experimental approaches are capable of constructing true grain structures; however, they are in general time-consuming, laborious and expensive for practical use. The achieved grain structures are also difficult to extrapolate to large specimens. An alternative way to generate a virtual grain structure can be achieved through mesoscale computational simulations of metallography. Microstructural development depends upon the evolution of internal interfaces in polycrystalline materials, which include the grain boundaries dividing grains with different crystallographic orientations and the phase boundaries separating regions of dissimilar thermodynamic phase. Internal interface evolution may occur due to a volumetric change *via*, for example, recrystallisation or phase transformation and proceeds by aiming to minimise the internal surface energy *via* grain growth. A successful mesoscale simulation of a microstructural evolution can provide a detailed description of grain structure geometry and crystallographic orientation, and hence can be represented in FE/CAE as a FE model.

In this section, three major types of models of grain-level material structure simulation are discussed including the phase-field models, the cellular automata, and the Monte Carlo (Potts) models. These mesoscale models assume a material as an ideal continuum without the explicit incorporation of atomic-scale dynamics. State variables in a set of governing equations are often based on a spatial grid and the governing differential equations are applied to them with respect to local or global interactions. The review of this section focuses on the models' capabilities of deriving grain boundary and/or orientation during grain structure formation and evolution, which can serve as grain structure representations for CPFE simulations.

2.2.1 Phase-field method

Recently, the phase-field method has emerged as one of the most powerful numerical approaches to modelling and predicting grain-level microstructure evolution. This method, describing the liquid-solid and solid-solid phase transformation phenomena within the framework of irreversible thermodynamics, allows many physical phenomena to be treated simultaneously. In the phase-field method, the state of a microstructure domain is represented using a set of field variables that are continuous across the interfacial regions as diffuse interfaces. In the continuum phase-field kinetic models, the field variables are governed by the Cahn-Hilliard nonlinear diffusion equation [73, 74] and the Allen-Cahn relaxation equation (time-dependent Ginzburg-Landau approach) [75]. The temporal and spatial evolution of detailed grain structural properties can be simulated without explicitly tracking the positions of the interfaces.

The phase-field method was firstly applied to model the solidification of materials, concerning dendrite formation and growth (e.g. [76, 77]), and the solid state phase transformations. Both solidification and solid-solid phase transformations focus on the reduction of bulk free energy. In the phase-field method, the free energy of a system is defined by the physical order parameters and the evolution of these parameters follows the Allen-Cahn and Cahn-Hilliard equations. A further development enabled the phase-field models to predict the microstructure evolution of grain growth and coarsening, by which the total interfacial free energy of a microstructure is reduced. The pioneering

work in [78] employed the continuum time-dependent Ginzburg-Landau model to simulate the domain-growth kinetics. In this model, the grain boundary energy was determined by the Allen-Cahn diffusion equation, and the grains with different crystallographic orientations were represented by a set of non-conserved order parameter fields. This model and its variations have become a popular tool to simulating grain growth kinetics for the two- and three-dimensional grain structure evolution [79-81]. Figure 2.6 shows two consecutive 3D grain structures during a simulated grain growth process [80]. The simulation was performed on a cubic lattice with $180 \times 180 \times 180$ grid points, using a multi-order parameter free energy model [82] with 20 order parameters. To minimise the effect of grain coalescence, grain orientations were dynamically reassigned after each time step. At the initial stage, a grain structure was a dense packing of a few thousand grains, and after a growth process, grains were eliminated *via* boundary migration and the mean grain size increased steadily as shown in Figure 2.6 (b). Note that, the resultant grain structure with detailed grain morphology and crystallographic orientations is ready to be represented in a FE/CAE environment for CPFE simulations. It is important to note that, in a phase-field model, the definition of the free energy density variation in the boundaries is arbitrary and the assumption of the existence of systematic gradients within the interface is not physically justified.

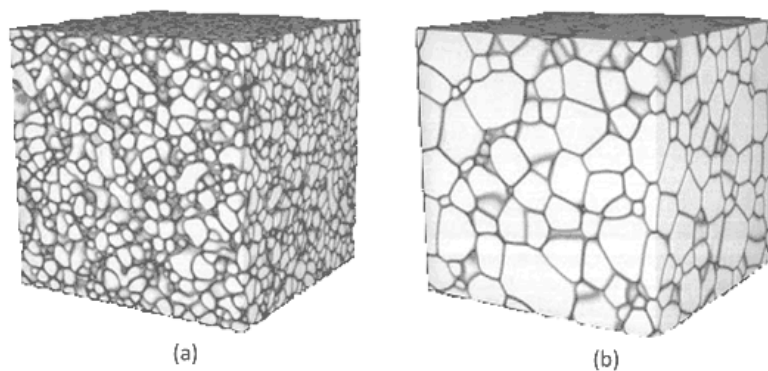


Figure 2.6. Example of microstructural evolution of grain growth using the phase-field model performed on a $180 \times 180 \times 180$ simple-cubic grid (Ref. [80]). (a) A grain structure with 5954 grains; (b) the grain structure after growth containing 514 grains.

2.2.2 Cellular Automata

Cellular automata (CA), implemented as a versatile computational framework, have been used to formulate the discrete spatial-temporal evolution of a complex system defined on a pre-defined cell lattice, which represents a grain structure. The system generally consists of a set of state variables, which relate to the individual lattice cells. In conjunction with local or global deterministic or probabilistic transformation rules to the lattice cells, the system evolves dynamically. Due to its flexibility of defining both a large variety of state variables and transformation laws, the CA method has been proposed to simulate a wide variety of phenomena of material microstructural evolution arising from recrystallisation [83, 84], dendrite growth [85], grain growth [86, 87] and phase transformation [88]. A major problem of this model is the absence of robust approaches for the treatment of nucleation including nucleation sites, rates and textures [89]. Using the CA method to simulate grain structure evolution requires an initial grain structure. This can be obtained from experiments such as those using SEM/EBSD microscopy, or from other computational models.

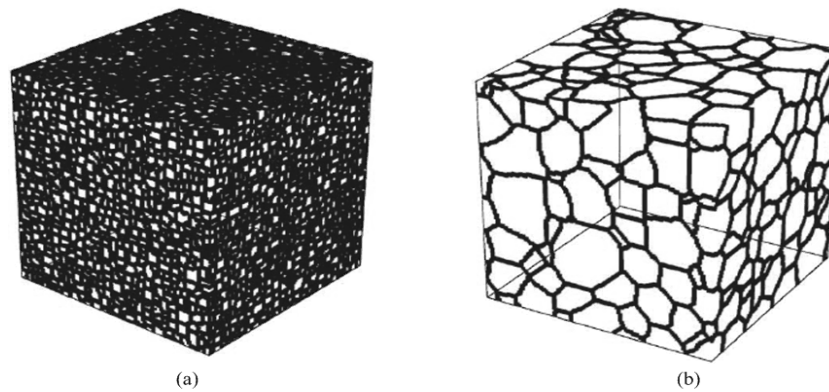


Figure 2.7. Microstructure evolution simulated using the CA model [87]. (a) The initial grain structure; (b) a simulated grain structure after grain growth.

Figure 2.7 shows a grain growth evolution simulated using the CA model reported in [87]. In this simulation, the microstructure was evolved in a lattice of $150 \times 150 \times 150$

cells. The initial grain structure presented in Figure 2.7 (a) was generated by a MC model and Figure 2.7 (b) shows a resultant grain structure after a simulated grain growth process using the CA model. In [90, 91], coupled models were proposed for cellular automaton recrystallisation simulation. In their implementation, the CPFE model was used to simulate plane strain compression of aluminium and the state variables such as dislocation density and crystal orientation were mapped to the lattice of a 2D CA model for subsequent simulation of the recrystallisation evolution phenomena.

2.2.3 Monte Carlo Potts model

Similar to the CA method, the Monte Carlo (MC) method is essentially a type of computational framework, which simulates the evolution of a physical system defined on a discrete grid domain. The simulated evolution is based on stochastic trials generated from a probability density function (PDF), in which a physical theory and the rules to decide the outcome of the stochastic trials can be incorporated. The Potts model has originally been applied to study the evolution of magnetic spin state by minimising the boundary of spin aggregates defined on a discretised domain. The Potts model was developed to evolve a grain structure domain represented by space-filling cells [92], where spin state was generalised to an arbitrary number instead of a binary state [93]. In this model, an individual grain was a cluster of lattice cells, which contained the same information such as crystallographic orientation and have the same index as shown in Figure 2.8. The energy of a system was given by a Hamiltonian, which in this case was determined based on the crystallographic misorientation between adjacent grains of different indices. Structural evolution was performed *via* sequential reorientation trials in accordance with a transition probability. The indices of candidate cells and the value of system energy were correspondingly changed.

The MC Potts model has been widely used to model the two- and three-dimensional microstructural evolution such as grain growth in single- [94, 95] and two-phase materials [96], directional grain growth [97], recrystallisation [98], solidification [99]. Similar to the CA method, a microstructure achieved from a MC Potts simulation has an explicit geometrical description and contains crystallographic information, and so it can

be applied to define the FE model for subsequent CPFE simulations. Note that, simulations using the MC Potts models are based on a pre-defined lattice. The lattice effects, resulting from the cell shape and lattice resolution, are non-physical but are the most common cause of invalid results [100]. Also, verification of a MC Potts model is difficult, since most of the basic physical features governing grain growth are not imposed on this model, for example, there is no formation resulting from the relationship between the migration rate of boundaries and the driving force. Thus, although the MC Potts model provides a fast and flexible tool for simulating grain structure evolution, this model is still a heuristic approach.

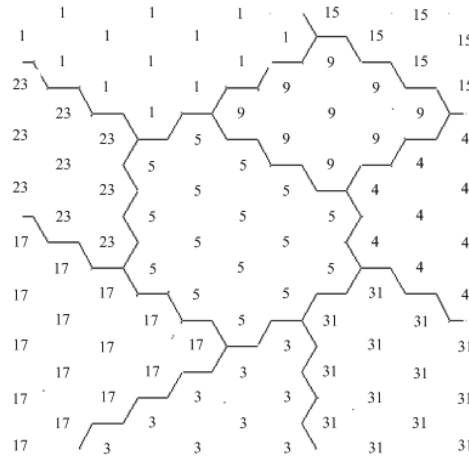


Figure 2.8. Illustration of a grain structure represented by a hexagonal lattice domain, where each grain is a cluster of hexagons of identical index.

2.3 Tessellation models

The tessellation models discussed in this section are referred to as a class of purely geometric representation of grain structures. A spatial tessellation containing space-filling cells is used to represent a grain structure, and the space-filling cells in the tessellation are defined as grains. Note that, the term of space-filling means the cells in a given domain are non-overlapping and fully occupy the given domain without gaps. In the context of grain structure modelling, the spatial tessellation is presented in a two- or three-dimensional domain, in which grains are polygons or polyhedra respectively. Note

that, the aforementioned metallographic approaches are capable of providing true grain structures corresponding to a material specimen, and the numerical models are able to produce virtual grain structures that conform to phenomenological or physical laws. However, in some circumstances, the above two approaches may not be available or suitable, for example in predicting a tailored material or theoretically studying the mechanical behaviour of a grain aggregate with special morphological features. In such context, the application of geometrical models is essential.

It is worth pointing out that the morphological and structural characteristics of virtual grain structures are oriented by the requirements of a FE analysis. An applied geometrical model is expected to effectively and efficiently generate a virtual grain structure with properties fulfilling the simulation requirements. In general studies, most of the commonly expected properties are grain size, shape and grain size distribution. The other prominent advantage of geometrical models is the high computational efficiency in generating virtual grain structures with a large number of grains and the capability of seamless integration to a FE/CAE platform.

2.3.1 Uniform tessellations

A uniform tessellation, consisting of grains of equal size and identical shape and defined on a uniform lattice, is the simplest geometrical model in representing a grain aggregate. In a regular grain lattice, the whole cluster of grains is embedded in a homogeneous matrix which has the common properties of the cluster. Each type of grain lattice corresponds to a particular type of regular space-filling cell (polygons or polyhedra). Figure 2.9 illustrates three types of widely used 3D grain lattices: simple cubic (SC) lattice, body-centred cubic (BCC) and face-centred cubic (FCC). Each individual lattice shown by the nodes specifies the locations of grain centroids, and the grain shape in the lattice is presented by the grain corresponding to the inner node (black). Note that, the nodes on the surfaces (blue) are the centroids of the adjacent grains, which can be obtained *via* parallel translations.

In geometry, there are a limited number of types of regular space-filling polygons and polyhedra [101]; the most commonly used shapes are triangles, squares and hexagons for a 2D domain, and cubes, hexagonal prisms, rhombic dodecahedrons and truncated octahedrons for a 3D domain, as shown in Figure 2.10. Each type of geometrical unit can be applied to generate a uniform tessellation *via* parallel packing. Table 2.1 presents the typical geometrical properties of the regular geometrical entities. The property of surface area per unit volume for a polyhedron or edge length per unit area corresponds to the total grain boundary energy which is a driving force for grain boundary evolution in a polycrystalline material. The larger value might lead to more mobility for grain boundaries and hence be less morphologically stable.

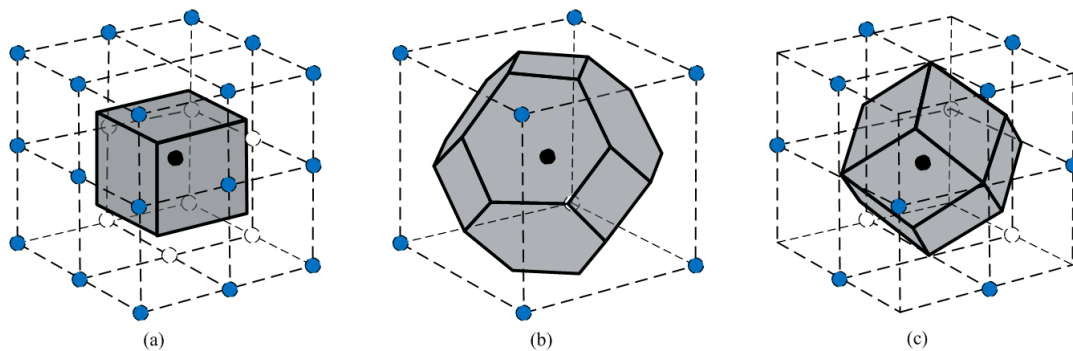


Figure 2.9. Illustration of three-dimensional lattices and their associated Voronoi polyhedra. (a) SC lattice, where grains are represented by cubes; (b) BCC lattice, where grains are truncated octahedrons; (c) FCC lattice, where the grain shape is a rhombic dodecahedron.

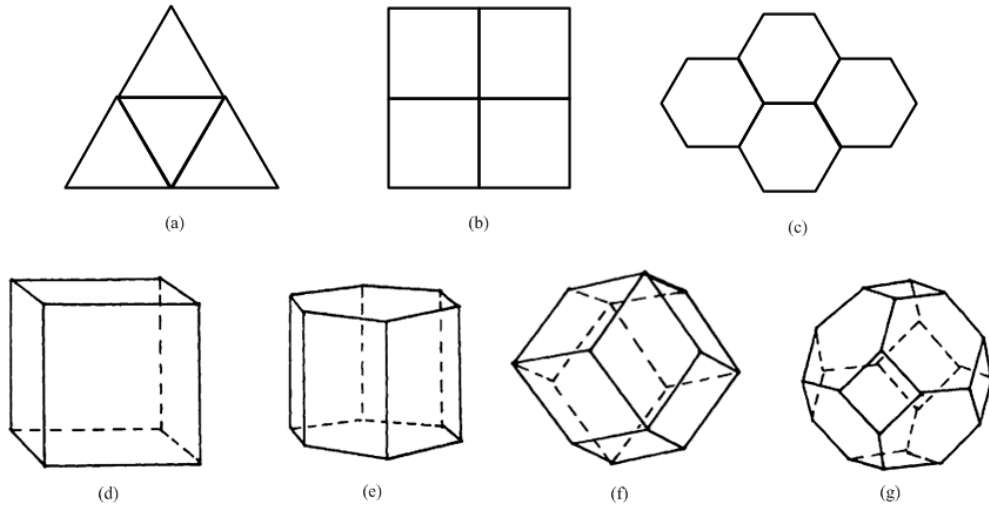


Figure 2.10. Space-filling polygons and polyhedra. (a) Equilateral triangle cluster; (b) square cluster; (c) regular hexagonal cluster; (d) cube; (e) hexagonal prism; (f) rhombic dodecahedron; (g) truncated octahedron.

Table 2.1. Properties of the geometrical elements of different grain shapes.

	Equilateral triangle	Square	Regular hexagon	Cube	Rhombic dodecahedron	Truncated octahedron
Spatial lattice	/	/	/	SC	FCC	BCC
a/v or l/a †	4.56	4	3.72	6.00	5.35	5.31
Vertices	3	4	6	8	14	24
Edges	3	4	6	12	24	36
Faces	/	/	/	6	12	14

† a/v signifies surface area per unit volume for a polyhedron, and l/a denotes edge length per unit area.

Except for the space-filling triangle and tetrahedron unit, that are usually taken as FE mesh blocks, other regular geometrical units have been widely used to act as idealised grains to represent a polycrystalline material for CPFE simulations. As in the work of [102], a comprehensive study of the evolution of crystallographic texture during large

deformation of FCC materials was conducted using a finite element implementation of a generalised Taylor-type polycrystal constitutive model. In their simulations, an aggregate of 400 grains subjected to a simple shear deformation and a plane strain compression was represented by square 2D elements, and a set of 343 cubes was used to represent a polycrystal to simulate simple compression and tension. Similar work has been done in [103] using a 2D square aggregate but a full 3D FCC slip system as a grain structure representation to simulate crystallographic texture evolution. The results showed a good agreement with experimental measurements in the simple shear and plane strain forging experiment on an oxygen free copper. In contrast, the work in [104] compared the influence of grain shape on the texture development in cold rolled ultra low carbon steel under uniaxial tension. Different grain structures were represented by cubes and truncated octahedrons respectively. As reported, the simulations using truncated octahedral grains produced improved predictions on the texture evolution compared to those with cubic grains. In [105], the CPFE method was used to investigate the deformation and failure of stent struts under uniaxial tensile loading. A regular hexagonal tessellation and a realistic random tessellation were used to examine the shape effect in predicting the deformation of 2D uniaxial tensile. The uniform tessellations of hexagonal prism and rhombic-dodecahedron were also used to model grain structures to investigate the tensile behaviours of stent struts [106]. It was found that the grain structure represented by the regular hexagonal prism tessellation yielded a significantly higher necking strain than that employing the rhombic-dodecahedron tessellation.

Note that, in polycrystalline materials, there are two major factors causing anisotropic plastic response: crystallographic texture and morphological properties of grains. Since macroscopic mechanical responses, e.g., stress and strain, can be strongly influenced by the grain size, morphology and orientation, the employment of different shape of grains inevitably has an influence on the results of material behaviours. As presented above, in early research, uniform tessellations were popular in representing grain structures for investigating the evolution of crystallographic texture and corresponding influence on mechanical behaviours. Thus, the geometrical effects of grain shape, size and arrangement were not properly taken into account. As aforementioned,

FE models based on different uniform tessellations yielded different mechanical responses, some of which might have better predictions than others. In addition, the inter-granular and intra-granular stress variations depend consistently on the grain shape, as the rigorous studies in [107, 104, 12, 108] showed. Since in reality most grain structures exhibit large variations in grain shape and size, uniform grain structures fail to account for the natural variations of a grain's morphology. Instead, a more realistic treatment of grain structure properties of grain shape and grain size distribution will improve the accuracy of CPFE simulations.

2.3.2 Random tessellations

The random tessellations, being widely used in the field of metallurgy to model grain structures, are a family of geometrical models, which mimic solidification processes by incorporating the elementary geometry of nucleation, grain growth and impingement. A solidification process of a polycrystalline material begins with grain nucleation, followed by growth and ends with all crystal surfaces impinging on adjacent crystals. The grain size and morphology of a resultant structure from a solidification process depends on the distribution of nucleation sites in both space and time, and the growth velocity of the grains after nucleation. There are basically two perspectives to categorise nucleation settings. The first type is that nuclei can be stationary, i.e. fixed at the initial nucleation location, or changeable, e.g. nuclei are floating or absorbed by others. The other type is that nucleation can be time-dependent or independent. The time-dependent nucleation means that the population of nuclei are not presented simultaneous but progressively formed.

On the other hand, there are fundamentally two categories of growth-rate relationships. Grain growth can be isotropic, that is, the growth velocities for all directions are the same, or anisotropic. For a 2D grain with anisotropic growth, its external boundary expands in an ellipse boundary. The other is grain growth velocity which may be constant or variable during growth. For an isotropic constant grain growth, grains with circular or spherical boundaries grow to impinge on other grains. As a result, the two factors of nucleation and grain growth influence the grain shape, size and grain

size distribution. Grain boundary migration also affects the final morphology and size of grains. After grain impingement on each other, microstructure evolution may be continued *via* grain boundary migration. Such a grain boundary migration attempts to reduce grain boundary energy and causes grain growth.

To model these physical solidification and grain growth processes, a variety of random tessellation models have been proposed. The term of random tessellation is named mainly because the nucleation sites are randomly generated according to a probability distribution or perturbed by a random noise. In the following discussion, random tessellations are categorised in terms of the three dominant factors: nucleation, grain growth and boundary migration; geometrical features of corresponding virtual grain structures are detailed.

As mentioned before, the nucleation condition can be time-dependent or time-independent. The time-independent nucleation means that nucleation sites (grain seeds) are simultaneously presented before grain growth occurs. In contrast, in a time-dependent nucleation process, seeds are produced sequentially with an interval of a constant period or a continuous description. Assuming the growth velocity for any grain is isotropic and constant and seeds are stationary, it yields the Voronoi tessellation (VT) model, which has been traditionally applied in metallurgy to represent polycrystalline material microstructures (e.g. [109-111, 22]). Figure 2.11 (a) schematically illustrates the grain boundary morphology of a VT where grain growth, presented by three sets of concentric circles and based on seeds s_1 , s_2 and s_3 , simultaneously starts and progresses with identical constant velocities. The resultant boundaries are straight lines, where the correlated adjacent circles (i.e. at the same level) have impinged. Furthermore, to match the metallographic observations of non-uniform grain shape and size in the microstructures, seeds in a Voronoi tessellation are usually randomly scattered in the given domain. The seeds can be randomly generated from a Poisson spatial point process, which is usually called a Poisson Voronoi tessellation model, or by perturbing a regular seed lattice with random noise [112].

For a time-dependent nucleation condition, the simplest case is the continuous nucleation where the nucleation rate, defining the seed addition rate per unit region (area or volume), remains constant during solidification. When applying this condition to a 2D event by means of sequentially generating seeds according to a Poisson spatial point process, if the growth rate is constant and isotropic for all grains, then the resultant model is the Johnson-Mehl (J-M) model [113, 114]. Detailed studies on the properties of J-M tessellations were reported in [115, 116]. Figure 2.11 (b) illustrates a schematic nucleation-growth process of a J-M tessellation. In this case, the nucleation of seeds s_1 , s_2 and s_3 occurred sequentially at times t_1 , t_2 and t_3 , and hence the final boundaries cannot be straight, but are segments of hyperbolic lines. Also, grains in a generated tessellation are not always convex.

A more general time-dependent nucleation condition is that a nucleation rate is variable during the nucleation-growth process. A typical case is that a nucleation rate declines with time. For example, provided an initial density of potential nucleation sites, a fraction of them are dismissed due to the regions being occupied by other growing grains before nucleation happens. As the given case in [117], the probability of nucleation at a given site per unit time remains constant while the region being taken by the growing grains, the expected remaining seeds N , followed an exponential decline with time: $N(t) = N_0 \exp(-at)$, where N_0 is an initially specified value, and so is the nucleation rate (dN/dt). It is worth mentioning that the limit $a = 0$ is the constant nucleation rate condition, and the limit $a = \infty$ reduces to the Poisson Voronoi tessellation model.

The other category of solidification conditions is the growth-rate relationship. The aforementioned random tessellation models assume that the grain growth is isotropic and growth velocities for all the grains are constant and identical. There are also a class of random tessellation models that concentrate on the growth effects and corresponding grain structure features. One typical setting considers that the growth rate of each grain varies according to its radius, and for simplicity, all grains have an identical growth rate relation (e.g. a growth rate proportional to a power of the grain radius [118, 119]). In addition, provided the seeds are simultaneously given, the above growth-rate relation

does not affect the final grain structure properties and the resultant tessellations are VTs as shown in Figure 2.11 (c). However, if a nucleation condition is time-dependent, then the grain morphology and grain size distribution are determined by the growth-rate of adjacent grains. In Figure 2.11 (d), seed s_2 was nucleated one time-step later than seed s_1 . Since both share an identical growth-rate relation, the boundary between them is curved. It should be noted that under the assumption of constant nucleation rate and identical radius-dependent growth rate, the random tessellations are equivalent to a family of weighted Voronoi tessellations (e.g. [120-122]).

The aforementioned growth is homogeneous, that is, growth rate is identical for all grains. There is also a scenario of inhomogeneous grain growth such that different grains have their own growth velocities for simplicity. As shown in Figure 2.11 (e), the three seeds were generated at the same time, but during grain growth, three seeds grew at their own constant speed. Since the velocity ratio between s_1 and s_3 was very large, the seed s_3 was almost embedded into seed s_1 . This type of random tessellation has been used to simulate the microstructure of metallic dual-phase polycrystalline materials in [123]. Their method accounted for the difference in grain growth velocities in different phases during solidification. The simulated grain structures showed a large variety of grain size and morphology, changing with the ratio of grain growth velocities of different phases. As the ratio tended to infinity, grains from one phase were embedded into grains of the other phase. Finally, as anisotropic grain growth has also been observed in polycrystalline materials, random tessellation models have also been developed to simulate solidification process with an anisotropic grain growth. For example, the work in [124] studied two growth modes, where grains grew based on either an elliptical or polygonal shape with a specific aspect ratio, under both simultaneous and continuous nucleation conditions.

It is worth pointing out that, despite the diversity of random tessellations in modelling solidification, the geometrical methods are easily implemented: a point on some boundary of two adjacent grains can be determined according to the following relation with respect to the impingement time t :

$$\begin{cases} t^i + t_g^i = t \\ t^j + t_g^j = t' \end{cases} \quad 2.1$$

where t^i and t^j are the nucleation time of the two adjacent grains, and t_g^i and t_g^j are time spent from nucleation to impingement. For a constant growth condition, the growing time can be calculated from $t_g^i = d_i/v_i$ and $t_g^j = d_j/v_j$, where d_i and d_j are the distances between seeds i and j to the boundary point, and v_i and v_j are the respective grow velocities of the two grains. Furthermore, a triple junction of adjacent grains i, j and k can be determined by $t^i + t_g^i = t^j + t_g^j = t^k + t_g^k$.

In addition to simulation of solidification processes, the random tessellations have also been used to model the microstructure evolution under boundary migration and grain growth [125-127]. To simulate grain boundary migration, an individual grain boundary is represented by an array of such points, that is, the boundary curve is an interpolation of them. The migration of a boundary means moving an array of points and the corresponding junctions, according to the local characteristics of a boundary, e.g. its curvature, the stress state or misorientation of two adjacent grains. Different from the aforementioned numerical models such as the Monte Carlo Potts method, the random tessellation models are capable of handling complex local geometry, especially at the junctions of grain boundaries. Moreover, it is highly efficient and committed to represent grain structures for further FE model representation.

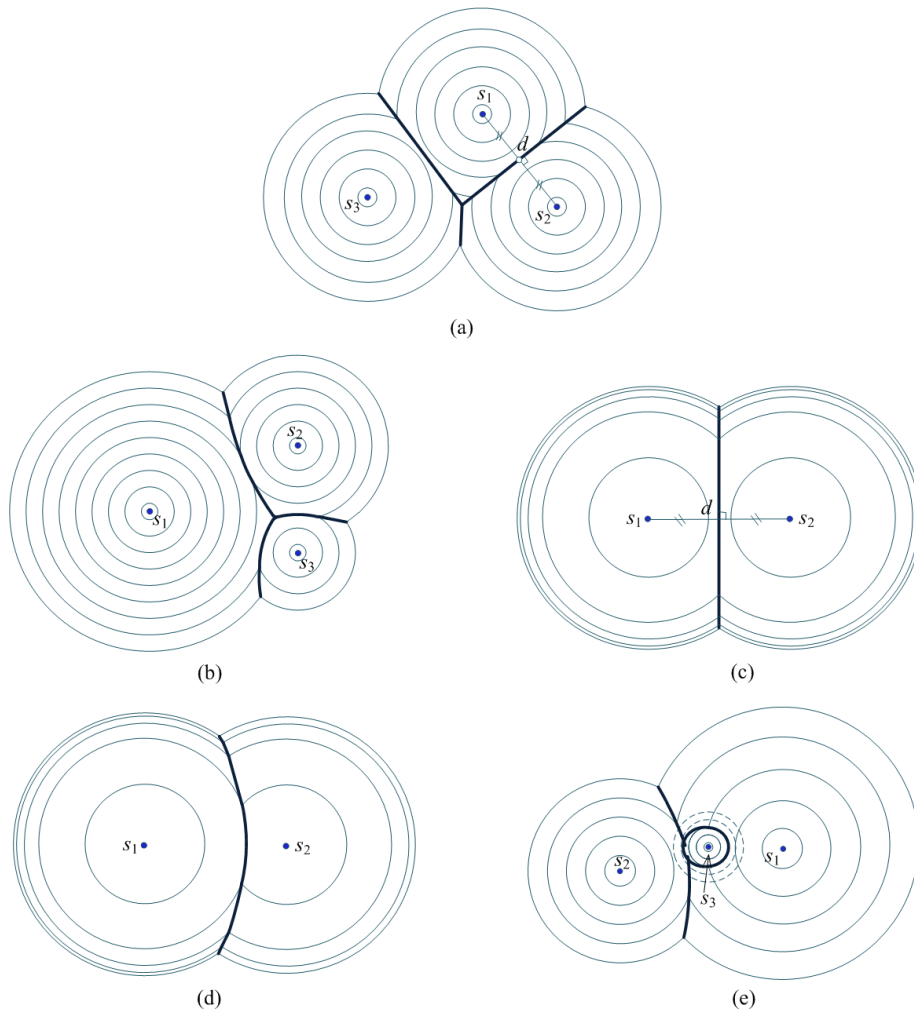


Figure 2.11. Schematics of grain boundary formation under the assumption of isotropic growth, where thin circles are progressive boundaries by constant time intervals, thick lines represents grain boundaries and centre nodes signify nuclei (seeds). (a) Simultaneous nucleation and globally-identical constant growth velocity; (b) sequential nucleation and globally-identical constant growth velocity; (c) simultaneous nucleation and globally-identical time-dependent growth velocity; (d) sequential nucleation and globally-identical time-dependent growth velocity; (e) sequential nucleation and constant but inhomogeneous grain growth.

As the random tessellations have been traditionally applied to modelling random grain structures of polycrystalline materials, they have been widely employed to

represent virtual grain structures for micromechanics simulations. The natural grain junction representation enables a CPFE simulation to easily cope with boundary geometry and further meshing, which is a significant improvement over other numerical models using a discretised representation, e.g. the CA and MC Potts models. More important is that the random tessellation model can efficiently handle the operation of large-scale grain structures for particular simulation requirements such as a micro-forming process. In addition, the concise representation of grain geometry benefits the pre-processing work being performed makes this very convenient and highly efficient. Different from the uniform tessellations, utilisation of a random tessellation to define a polycrystalline material gives a FE model more insight into the anisotropic mechanical responses resulting from the crystallographic and morphological textures. All the above advantages have promoted the random tessellation being widely utilised in CPFE applications.

For small-scale applications, random tessellations have been intensively used to understand local deformation mechanisms and study local damage initiation-propagation, with incorporation of the structural anisotropies of morphological texture and crystallographic texture. As the published results show that grain shape and organisation have a strong influence on local stress-strain fields (e.g. [107, 104, 12, 128]), the random tessellations, especially the Poisson Voronoi tessellations, have been used to study a range of localised properties of a polycrystal, e.g., shear localisation [129], micro shear banding [112] and dislocation density accumulation [128]. Having access to the high-fidelity local strain-stress fields, the micromechanics FE simulations using random tessellations can be a very powerful tool to study grain boundary sliding, crack initiation, and crack propagation (e.g. [130-132]). They also provide important local quantitative data, e.g. strain gradients, to other computational models for a variety of multi-physics simulations, e.g. recrystallisation and grain growth, or multiscale simulations. From a macro-scale perspective, the Poisson Voronoi tessellation model has been particularly applied to model virtual grain structures for various studies, mainly due to the extreme computational efficiency and natural representation of equiaxed grains.

As an indispensable tool, this model has been widely used in various large-scale micromechanics FE simulations, for example, estimation of the size of a representative volume element (RVE) of random heterogeneous materials (e.g. [133-135]), which gives a view to predicting overall properties and optimising microstructures. Moreover, as the Poisson Voronoi tessellation model can generate very large virtual grain structures with little computing efforts, the Voronoi tessellations have been employed as a standard virtual grain structure representation scheme for CPFE simulations of many micro-forming processes, e.g, hydro forming of a micro-tube [136], extrusion of a micro-pin [137], thin-sheet forming [138], and micro-mechanical tests (e.g. indentation, three-point bending, and rolling contact fatigue [139]). Note that, generation of Poisson Voronoi tessellations is based on the assumption that nucleation sites are purely randomly generated in the domain according to a Poisson point process, and hence the sophisticated features such as grain size distribution cannot be controlled.

2.3.3 Summary and discussion

Finite element (FE) methods are increasingly used for micromechanics modelling of polycrystalline materials. Polycrystalline FE has given valuable insights into grain-to-grain interaction behaviour and local deformation mechanisms, acts as an effective utility to simulate micro-forming processes for small-scale metal products, and is capable of simulating local damage processes of components in service. Since stress and strain are related to grain size, shape, orientation and their distributions, FE micromechanics simulations must be based upon a grain structure modelled within a FE/CAE computational environment. Therefore, representation of the microstructure in a realistic manner is critical for accurate simulations.

A wide range of approaches have been developed to fulfil the grain structure representation task. As this chapter outlined, the metallographic techniques, numerical models, and tessellations are the three most commonly applied methods to generate grain structures. A variety of experimentally-based methods using metallographic techniques have been developed to acquire images of actual grain structures, which can be used to define grain structures for micromechanics FE simulations. Simulations based

on the experimentally-determined geometric structure and crystallographic orientations allow for a comprehensive understanding of the relations between the structures and mechanical response in high fidelity. However, these methods are commonly very time-consuming and extremely laborious, and often require destruction of the material sample. Some of them are very expensive or cannot be accessed by the mainstream research community. More importantly, these methods have difficulty producing large-scale grain structures for large-scale FE simulation such as micro-forming of a polycrystalline material. The numerical models, on the other hand, have the advantages of repeatedly producing simulated grain structures in a large scale without experimental constraints. Successful simulations of the specific physical phenomena will achieve simulated grain structures in the computational environment, which can further be incorporated into a FE model. The geometrical models are the most efficient approaches in generating virtual grain structures in a FE/CAE platform. Although uniform tessellations have been used in prediction of crystallographic texture evolution, they paid little attention to the grain morphological anisotropy and hence the local strain-stress fields around grain boundaries and junctions. In contrast, random tessellations are capable of capturing various morphological properties meeting the requirements of advanced micromechanics simulations. In a broad sense, the random tessellation models also belong to the numerical models, which simulate physical phenomena, such as nucleation-solidification and grain growth. Defining them in a separate category is mainly due to their particular data representation features, i.e. lines or curves, which enable the generation of virtual grain structures to be accomplished highly efficiently, and allow for direct importation to a FE/CAE system for FE modelling.

Note that all the above numerical models are oriented to simulate an evolution of a physical process, e.g. solidification. They mainly emphasise the feasibility and accuracy of the related simulated process, by which the resultant grain structures are assumed to be inherently equivalent to that of the originals. However, a critical issue that has been ignored is that these models are commonly based on some simplified conditions that inevitably affect the final grain structures. For example, in generating the virtual grain structure, the Poisson Voronoi tessellation model makes the assumption that grain

nucleation is generated from a Poisson point process, i.e. purely random. Since the seed distribution dominates the final grain structure's uniformity and hence the grain size distribution [140], a lack of an effective specification of such a nucleation condition may not produce a proper grain structure model for micromechanics studies. In addition, in some simulation scenarios, original materials may not be available, e.g. when designing a special material with particular mechanical properties.

Therefore, the method to generate application-oriented (AO) virtual grain structures is very valuable for micromechanics simulations. Such an AO method should be capable of generating a virtual grain structure that is statistically equivalent to the desired grain structures based on the requirements of a micromechanics simulation. These properties can be directly taken from the metallographic measurements, or the properties can be a set of arbitrary values with respect to an imaginary grain structure. Note that, in many circumstances, grain size distribution characteristics are vital properties for accurate prediction or simulation of mechanical behaviour of a polycrystalline material. A few studies have been proposed using the ellipsoid packing method to synthesise a virtual grain structure (e.g. [141, 142]). As an AO method, it attempted to achieve grain size distribution control in generating polycrystalline material by means of a sequence of operations on optimal packing the pre-generated representative ellipsoids, and then some numerical models such as CA and MC Potts methods were used to adjust the final grain shapes. But, despite the complex geometric computations of e.g. optimal packing, overlaps and voids, the overlaps among grains and domain boundaries changed the size distribution properties. Instead of artificially producing and packing individual grains, this work develops a novel scheme self-consistently within one single tessellation framework, namely the controlled Poisson Voronoi tessellation model (CPVT), to generate virtual grain structures with grain size distribution control.

Chapter 3

Controlled Poisson Voronoi tessellations

3.1 Introduction

In large-scale CPFE analyses, e.g. micro-forming simulations, a polycrystalline material contains a large number of grains so that the size of grains exhibits a distributional feature. Namely, the size of grains varies and overall appearance of grain structures with different grain size dispersion characteristics shows different uniformity. As intensive studies revealed, not only the mean grain size but also grain size dispersion have significant impact on both of the macroscopic plastic flow stress and local stain-stress development [15]. In addition to crystal plasticity, other mechanical properties such as creep rate [20], crack propagation rate [21], and creep damage evolution [13] also strongly relate to the grain size distribution characteristics of a polycrystalline microstructure.

Voronoi tessellations (VTs) have been used as a conventional tool to generate grain structure for large-scale CPFE simulations. Grains in a VT filling the domain contiguously are non-uniform and usually exhibit large variability, as observed in natural polycrystalline microstructures. However, as VTs are disordered, there are still no clear rules for interpreting the organisation and geometrical constraints of the grains produced by the tessellations. This is mainly because the initial nucleation distribution condition is completely simplified, although it simulates a homogeneous crystallisation process.

In most applications of CPFE simulation, generating seeds for Voronoi tessellation is performed according to a Poisson spatial point process or by means of perturbing a uniform tessellation with random noises. Although there is no clear rule to verify whether a randomly generated seed lattice is appropriate to simulate a real solidification process for a particular microstructure; given any real polycrystalline microstructure, its seeds must have been nucleated based on a certain distribution pattern. This governs the

grain structural properties, e.g., the tessellation's uniformity, the mean grain size, and the grain size distribution [140]. From the perspective of generating an application-oriented virtual grain structures, one key issue is evaluating the degree of grain structural uniformity, and further using the degree of uniformity to generate the tessellation with control of the grain size distribution.

This chapter is dedicated to development of an application-oriented virtual grain structure generation model, namely the controlled Poisson Voronoi tessellation model. This model is based on the geometrical description of simple Voronoi tessellation, that is, all grains assuming an isotropic growth with an identical velocity. A virtual grain structure generated using the CPVT model has the property that its grain size distribution is statistically equivalent to the actual grain structure in term of the specified physical parameters: the mean grain size, a small grain size, a large grain size, and the percentage of grains within that range. Development of the CPVT model requires three steps:

- 1) Defining the regularity that specifies the uniformity of a tessellation, and deriving the control parameter based on the regularity.
- 2) Establishing the mapping from the regularity to the distribution parameter of a one-parameter gamma distribution.
- 3) Defining the mapping from the set of physical parameters to the distribution parameter.

Defining the quantities or mappings in the first two steps depends upon the domain's dimension. The following part of this chapter only concentrates on establishing the relation of the specific physical parameters with the grain size distribution property of a given grain structure. In the beginning, definition of related tessellation models is provided and important geometrical relations are discussed. Then, a distribution function is applied to give a mathematical description of the grain size statistics of a grain structure. The relation between physical parameters with the parameter of the distribution function is rigorously studied. A mathematical proof of the uniqueness of the determination of the distribution parameter from the proposed set of physical

parameters is provided, and an efficient numerical procedure is proposed for computing the distribution parameter.

3.2 The models

3.2.1 Voronoi tessellations

A polycrystalline grain structure can be modelled by a Voronoi tessellation, provided the physical assumptions are satisfied: 1) All nuclei (seeds) appear simultaneously and remain stationary during the grain growth process; 2) grain growth is isotropic and growth velocity is identical for all grains; and 3) grain growth ceases to form a grain boundary whenever a grain comes into contact with another growing grain boundary.

The geometrical properties of a Voronoi tessellation can be described as follows (see e.g. [143] for further details). Given a domain $\Omega \subset \mathbb{R}^2$ with a set of distinct points $S = \{s_i\}$ (i.e., no two or more points spatially coincide), where $s_i \in \Omega$ is referred to as nuclei or seeds. Each seed s_i is associated with region given by

$$V(s_i|\mathbb{R}^d) = \{t \in \mathbb{R}^d \mid ||t - s_i|| \leq ||t - s_j|| \text{ for all } s_j \in S\}, \quad 3.1$$

where $|| \cdot ||$ denotes Euclidean distance and if $d = 2$, $V(s_i)$ are planar Voronoi polygons and if $d = 3$, $V(s_i)$ are Voronoi polyhedra. Equivalently,

$$V(s_i|\mathbb{R}^d) = \cap_{j:j \neq i} H(s_i, s_j), \quad 3.2$$

where $H(s_i, s_j)$ is the closed halfspace to the seed i , that is,

$$H(s_i, s_j) = \{t \in \mathbb{R}^d \mid ||t - s_i|| \leq ||t - s_j||, \ j \neq i\}. \quad 3.3$$

Provided $d = 3$, the boundary between the two seeds $s_1(x_1, y_1, z_1)$ and $s_2(x_2, y_2, z_2)$ is a perpendicular bisector plane, which can be obtained by

$$\alpha x + \beta y + \gamma z = \theta, \quad 3.4$$

where α, β, γ and θ are coefficients, determined by the given seeds as

$$\begin{cases} \alpha = 2(x_2 - x_1) \\ \beta = 2(y_2 - y_1) \\ \gamma = 2(z_2 - z_1) \\ \theta = (x_2^2 - x_1^2) + (y_2^2 - y_1^2) + (z_2^2 - z_1^2) \end{cases} . \quad 3.5$$

For $d = 2$, the boundary between two seeds $s_1(x_1, y_1)$ and $s_2(x_2, y_2)$, are degenerated into a perpendicular bisector line. Similarly, the boundary can be decided by

$$\alpha x + \beta y = \theta, \quad 3.6$$

where the coefficients α, β and θ are determined by

$$\begin{cases} \alpha = 2(x_2 - x_1) \\ \beta = 2(y_2 - y_1) \\ \theta = (x_2^2 - x_1^2) + (y_2^2 - y_1^2) \end{cases}, \quad 3.7$$

And further, the set given by

$$V = \{V(s_i) \mid i = 1, 2, \dots, n\} \quad 3.8$$

is named as the *Voronoi diagram*. Furthermore, a finite domain $\Omega \subset \mathbb{R}^d$ is tessellated by V , that is, $s_i \in \Omega$ and $V(s_i) = V(s_i|\Omega)$, hence the Voronoi diagram V is referred to as a *Voronoi tessellation*. Note that, in this work, Voronoi tessellations are used to model grain structures, therefore, for simplicity, both polygons in a 2D tessellation and polyhedra in a 3D tessellation are denoted as grains of a Voronoi tessellation.

3.2.2 Poisson Voronoi tessellation

Most real-world Voronoi tessellations are non-uniform, and can be easily represented by a random Voronoi tessellation, in which generation of seeds are associating with a stochastic point process. A stochastic point process is a probabilistic generation of points in a given domain according to a probability distribution function

defined over the domain. That is, let a Ω be a non-empty subset of \mathbb{R}^d , i.e., $\Omega \subset \mathbb{R}^d$ and $N(A)$ be the number of points in a subset A , where $A \subseteq \Omega$. A stochastic point process is referred to as the process where points are generated in accordance with a probability distribution $P_r(N(A) = k), k = 0, 1, \dots$, for any A in Ω .

The most fundamental point process is the Poisson point process. The points are stochastically independent and the probability of the number of points $N(A)$ in a region A is given by the Poisson distribution:

$$P_r(N(A) = k) = \frac{(\lambda|A|)^k e^{-\lambda|A|}}{k!}, i = 1, 2, \dots, \quad 3.9$$

where λ is the intensity of the point giving the mean value of points in a unit area (or volume) and $|A|$ means area (or volume) of A . Obviously, the distribution of $N(A)$ does not depend on the location of A but only on its area (or volume), and also there are no interactions among the points. Accordingly, the random tessellation, in which seeds are generated in accordance to the Poisson point process, is called the Poisson Voronoi tessellation.

3.2.3 Hard-core model

If relaxing the condition that seeds are independently generated and introducing a minimum scattering distance between seeds, a hard core process [143] can be defined by the distribution P :

$$P(S) = \int \dots \int_{\{s_1, \dots, s_k\}} f(s_1, \dots, s_k) ds_1 \dots ds_k, S \in \Omega, \quad 3.10$$

where S is a set of points in the domain Ω , and the probability density function is given by

$$f(s_1, \dots, s_k) \propto \exp(-H(s_1, \dots, s_k)), \quad 3.11$$

where the function H is referred to as interaction energy function, given by

$$H(s_1, \dots, s_k) = \sum \sum_{1 \leq i \leq j \leq k} h(\|s_i - s_j\|), \quad 3.12$$

and the potential function h is,

$$h(\|s_i - s_j\|) = \begin{cases} \infty, & \text{if } \|s_i - s_j\| \leq \delta \\ -b, & \text{if } \delta < \|s_i - s_j\| \leq D \\ 0, & \text{if } \|s_i - s_j\| > D \end{cases} \quad 3.13$$

The function h is governed by the scattering distances δ and D . If the distance is no more than δ , the potential is such large that the consecutive seed is not allowed to be generated in this range, while if the distance is within $(\delta, D]$, there is an attraction ($b>0$) or repulsion ($b<0$) influence. Correspondingly, the new random Voronoi tessellation model based on the hard-core point process is named as the hard-core tessellation. Note that, introduction of the seed distance corresponds to a nucleation-exclusion zone for an individual seed, which may come from a pre-growing condition to all present seeds. If the scattering distances are set to zero and there is no propelling or attracting energy among the points, then the hard-core Voronoi tessellation model is reduced to the Poisson Voronoi tessellation model.

3.2.4 The CPVT models

Given a domain Ω and letting the seeds be generated in this domain by a hard-core process with the potential function defined by,

$$h(\|s_i - s_j\|) = \begin{cases} \infty, & \text{if } \|s_i - s_j\| < \delta \\ 0, & \text{if } \|s_i - s_j\| \geq \delta \end{cases} \quad 3.14$$

The function h is only determined by the minimum seed spacing δ , that is if the distance between a newly generated seed s_i to any of the existing seeds s_j is less than δ , the potential is so large that the new seed is not accepted. In the CPVT model, the minimum seed spacing, determined by the specified physical parameters, is employed as a control parameter, dictating the distances between it and its neighbours to generate a seed

lattice for Voronoi tessellations. In detail, when providing a control parameter δ , the seed lattice generation process proceeds as follows: Seeds are generated sequentially in a given domain by sampling their coordinates independently based on a uniform random number generator. After the first seed is produced, each subsequent seed is only accepted if it is greater than a minimum allowable distance from any existing seed. Provided that every seed has an exclusive zone with the same diameter δ , the distance between a newly generated seed i and an existing seed k satisfies the condition $d(i, k) \geq \delta, \forall k = 1, 2, \dots, i - 1$. This procedure corresponds to a propelling force being present so that the simultaneous nucleation sites are scattered under their interactions before significant grain growth begins. For example, as illustrated in Figure 3.1, the new seed g is accepted because the distances between g and the other adjacent seeds are all equal to or larger than δ .

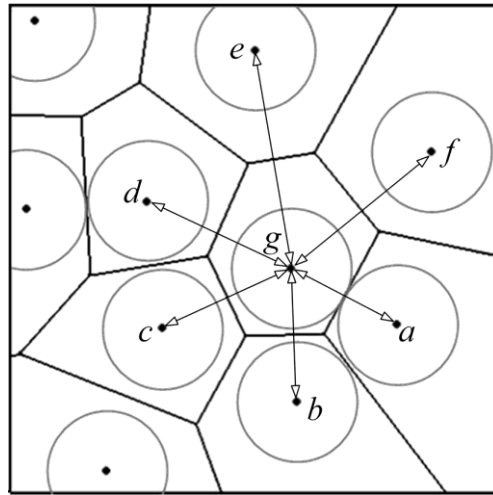


Figure 3.1. Schematic pattern of a controlled Poisson Voronoi tessellation, where the control parameter is the minimum seed spacing δ , and the seed distance $d(g, a) = \delta$ and $d(g, s) > \delta$, where $s = b, c, \dots, f$.

The core mechanism of the CPVT model is the scheme to determine the control parameter δ based on a set of physical parameters, by which any repeatedly generated Voronoi tessellations are expected to be statistically equivalent to the grain size distribution property given by the physical parameters. To be noted in the next chapter,

the control parameter dictates the seed dispersion and hence the uniformity of the related tessellation. That is, the larger the control parameter value, the more regular a tessellation appears. A global quantity of tessellation's regularity α is used to define a Voronoi tessellation's uniformity; as a global evaluation, it also relates to the grain size distribution characteristics. That is, by providing a probability distribution function to describe the grain size distribution characteristics, the global regularity can be mapped to the parameter(s) of the distribution function. The other module is established employing the physical parameters as input to specify the parameter(s) of the distribution function, which avoids directly using one or more obscure distribution function parameters to determine the control parameter.

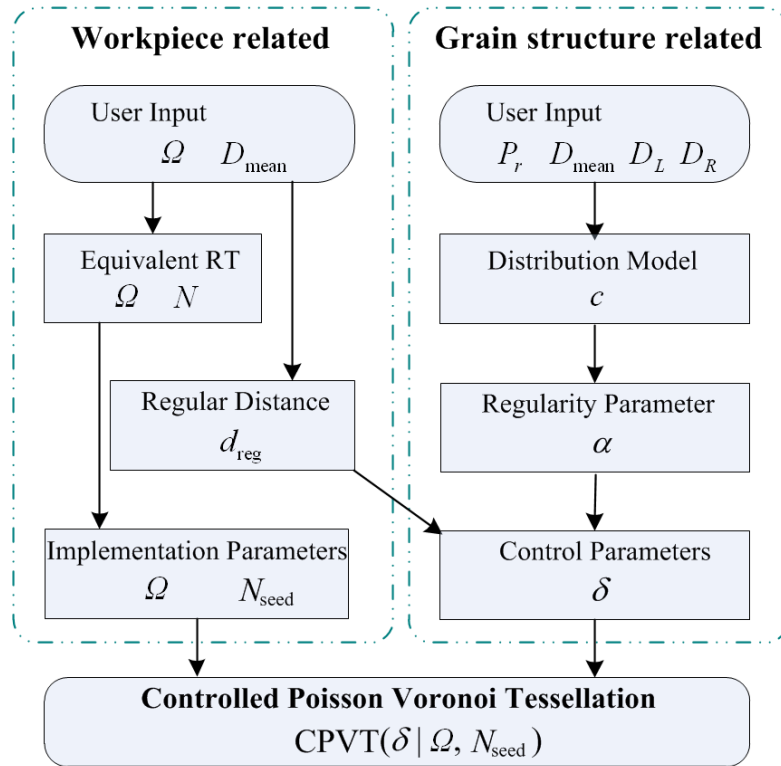


Figure 3.2. Scheme of the CPVT model for virtual grain structure generation.

In all, the CPVT model aims to fully control the tessellation's regularity and its corresponding grain size distribution using a set of physical parameter input. The control parameter is the primary link between the user-expected grain structure and the realised

virtual grain structure. Figure 3.2 shows the flow chart using CPVT to generate virtual grain structures. It is explained as follows:

- Workpiece related input includes the size of a workpiece and the mean grain size, which are used to obtain the implementation parameters including the tessellation's domain Ω and the number of grains N_{seed}
- Grain structure related input consists of four physical parameters, which describe a set of higher order grain size distribution features, to determine the control parameter δ .

In this work, the controlled Poisson Voronoi model is denoted as $CPVT(\delta|\Omega, N_{seed})$, where under the control of δ , N_{seed} seeds are sequentially generated within the domain Ω . The following part of this chapter is dedicated to define a grain size distribution function and establish the mapping between the physical parameters to the distribution function parameter(s). Details of the realisation and implementation of the CPVT model for two- and three-dimensional grain structures generation will be discussed in the remainder of this work.

3.3 Grain size distribution

Traditionally, the statistical distributions of the size parameters for Poisson Voronoi tessellations have been intensively studied. Several distribution functions have been suggested to describe the perimeter and area of grains in a planar tessellation, and surface area and volume of grains in a spatial tessellation. The log-normal distribution [144], Rayleigh law [145] and Maxwell speed distribution function [146] were proposed in early work, while, the gamma distribution functions have been widely employed to model grain size distributions (e.g. [147-153]) in terms of grain area for 2D tessellations and grain volume for 3D tessellations. Reported results include a three-parameter gamma distribution [152], a two-parameter gamma distribution [151, 148] and a translated gamma function [111]. Importantly, if the grain size is normalised by the mean, then a one-parameter gamma distribution can provide an accurate fit for the grain size distribution [153-155]. The one-parameter gamma distribution function only requires the

specification of one parameter and also provides a quality fit for a normalised grain size distribution for a 3D VT. Therefore, it is most suitable to model grain size distributions in that the single distribution parameter can be a central link from the physical parameters to the tessellation's regularity.

The one-parameter gamma distribution function takes the form of,

$$P_{x,x+dx} = \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} dx, \quad x > 0, \quad 3.15$$

where the parameter $c > 1$, and $\Gamma(c)$ is the gamma function, defined as

$$\Gamma(c) = \int_0^{\infty} x^{c-1} e^{-cx} dx. \quad 3.16$$

Note that the variance of the one-parameter gamma distribution is $1/c$. As the parameter c increases, the distribution becomes narrower, which is more suitable for modelling the tessellations having approximately similar grain sizes. In addition, there are two major advantages to using a one-parameter gamma distribution to describe the grain size distribution: Only one parameter c is involved in the relation for the tessellation's regularity, and the mean value of this distribution is one, which is the normalised mean grain size.

Consider a set of physical parameters $\{D_L, D_{\text{mean}}, D_R, P_r\}$ from the quantitative metallography, where D_{mean} is the mean grain size, defined as $D_{\text{mean}} = \frac{1}{N} \sum D_i$, D_L and D_R are two specific grain size values and P_r is the percentage of the grains with size in the range of $[D_L, D_R]$ over the total number of the grains N , that is,

$$P_r = \frac{1}{N} \sum m_i, \quad 3.17$$

where

$$m_i = \begin{cases} 1, & \text{if } D_i \in [D_L, D_R] \\ 0, & \text{otherwise} \end{cases}.$$

It should be noted that grain size D henceforth represents area in a two-dimensional tessellation and volume in a three-dimensional tessellation. When modelling grain distributions by a one-parameter gamma distribution, the set of parameters conform to the following equation:

$$P_r = \int_{x_1}^{x_2} \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} dx, \quad 3.18$$

where $x_1 = D_L/D_{\text{mean}}$, $x_2 = D_R/D_{\text{mean}}$ and $\Gamma(c)$ is a gamma function as in Eq. 3.16. Figure 3.3 presents an example of a one-parameter gamma distribution with $c = 16$, where the area of the shaded part between the lower limit $x_1 = D_L/D_{\text{mean}}$ and upper limit $x_2 = D_R/D_{\text{mean}}$ equals the percentage $P_r = 0.58$.

It can be observed that by specifying the physical parameters, the parameter c is correspondingly found by solving Eq. 3.18. However, there are still two major issues to be solved in order to utilise this scheme, which are: Conditions for which the parameter c is uniquely determined by Eq. 3.18, and an efficient numerical algorithm to solve Eq. 3.18.

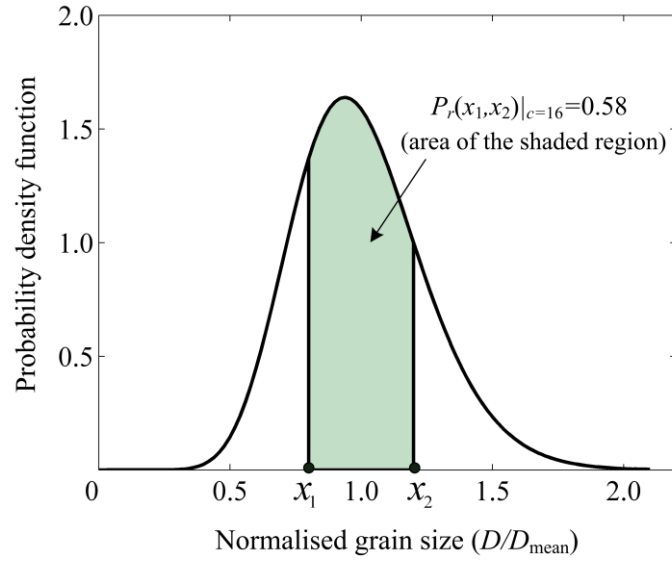


Figure 3.3. Example of a one-parameter gamma distribution, where the lower and upper bounds of the integral are $x_1=0.8$ and $x_2=1.2$, respectively.

3.4 Physical parameters

This section consists of two parts: First is the presentation of the condition for existence of a unique solution to Eq. 3.18, followed by an efficient algorithm for solving Eq. 3.18.

3.4.1 Uniqueness

Here *uniqueness* implies the uniqueness of the value of c obtained by solving Eq. 3.18. That is, given a set of physical parameters $\{D_L, D_{\text{mean}}, D_R, P_r\}$, there correspondingly exists a unique value of c characterising the one-parameter gamma distribution, describing the grain size distribution. Let,

$$\begin{cases} x_1 = 1 - \Delta_1, & 0 < \Delta_1 < 1 \\ x_2 = 1 + \Delta_2, & 0 < \Delta_2 < 1 \end{cases} \quad 3.19$$

where x_1 and x_2 are the integral limits of Eq. 3.18. For the symmetric case where $\Delta_1 = \Delta_2 = \Delta$, $\bar{x} = (x_1 + x_2)/2 = 1$, which is also the mean value of a one-parameter gamma distribution.

Lemma 1. *There exists an interval $S^* = (x_1^*, x_2^*)$, such that for any interval $S = (x_1, x_2)$, where $x_1 < 1 < x_2$, if $S \subseteq S^*$, the implicit function*

$$P_r(c) = \int_{x_1}^{x_2} \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} dx \quad 3.20$$

is strictly monotonically increasing, as the parameter c increases for any $c \geq 1$. Moreover, the interval $S^ = (x_1^*, x_2^*)$ can be estimated by*

$$\begin{cases} x_1^* \approx 1 - \frac{1}{\sqrt{c}} + \mathcal{O}(\Delta^{3/2}) \\ x_2^* \approx 1 + \frac{1}{\sqrt{c}} \end{cases}, \quad 3.21$$

where the term $\mathcal{O}(\Delta^{3/2}) > 0$.

Theorem 1. *Given the constants x_1, x_2 and the percentage value $P_r(x_1, x_2)|_c$, where $x_1 < 1 < x_2$, if the interval $S = [x_1, x_2] \subset S^* = (x_1^*, x_2^*)$, where x_1^* and x_2^* are determined from Eq. 3.21, the implicit function $P_r(c)$ in Eq. 3.20, defined over the domain $D(P_r) \subseteq [1, \infty)$ and range $R(P_r) \subseteq (0, 1)$, has the properties that:*

- $P_r(c)$ is a bijection;
- $P_r(c)$ has a continuous inverse $P_r^{-1}(c)$ on the range $R(P_r)$.

Lemma 1 and Theorem 1 explain the existence of a valid range of physical parameters, where uniqueness is satisfied. Proofs of Lemma 1 and Theorem 1 are given in Section 3.5. This is realised by choosing a small grain size D_L , a large grain size D_R and a mean grain size D_{mean} , such that $S = [x_1, x_2] \subset S^* = (x_1^*, x_2^*)$. Then, the parameter c is uniquely determined by the percentage value P_r . Although the existence of such an interval of physical parameters that uniquely defines the c values has been proved, this

interval, estimated by Eq. 3.21, is not large enough for all practical applications. In the following, efforts are made to extend the effective interval S^* .

3.4.2 Estimation of S^*

To estimate the interval S^* means to find a valid interval $S^* = (x_1^*, x_2^*)$ as large as possible such that $\frac{\partial P_r}{\partial c} > 0$. Since $S^* = (x_1^*, x_2^*) = (1 - \Delta_1^*, 1 + \Delta_2^*)$, to estimate the valid interval S^* is equivalent to finding possible large values of both Δ_1^* and Δ_2^* . For simplicity and without loss of generality, only a symmetrical situation is considered, where $\Delta_1 = \Delta_2 = \Delta$. Let the function $\varphi(\Delta)$ be

$$\varphi(c, \Delta) = \frac{\partial P_r(c, \Delta)}{\partial c} = \frac{c^c}{\Gamma(c)} \int_{1-\Delta}^{1+\Delta} x^{c-1} e^{-cx} [1 + \ln c - \psi(c) + \ln x - x] dx. \quad 3.22$$

where $\psi(c)$ is the digamma function defined as the logarithmic derivative of the gamma function,

$$\psi(c) = \frac{\Gamma'(c)}{\Gamma(c)}.$$

Furthermore, for any $c > 1$, let

$$\begin{cases} g_1(\Delta) = 1 + \ln c - \psi(c) + \ln(1 - \Delta) - (1 - \Delta) \\ g_2(\Delta) = 1 + \ln c - \psi(c) + \ln(1 + \Delta) - (1 + \Delta) \end{cases}, \quad 3.23$$

and there exist $\tilde{\Delta}_1, \tilde{\Delta}_2 \in (0, 1)$ such that

$$\begin{cases} g_1(\tilde{\Delta}_1) = 0 \\ g_2(\tilde{\Delta}_2) = 0 \end{cases}. \quad 3.24$$

Due to $0 < \tilde{\Delta}_1 < \tilde{\Delta}_2 < 1$, the property of $\varphi(c, \Delta)$ is studied with respect to three sub-intervals:

$$I_1 \cup I_2 \cup I_3 = (0, \tilde{\Delta}_1] \cup (\tilde{\Delta}_1, \tilde{\Delta}_2) \cup [\tilde{\Delta}_2, 1).$$

Lemma 2. *There exists a point $\Delta^* \in (\tilde{\Delta}_1, \tilde{\Delta}_2)$, where $\tilde{\Delta}_1$ and $\tilde{\Delta}_2$ are given by Eq. 3.24, such that for any $\Delta \in (0, \Delta^*]$, $\varphi(c, \Delta) > 0$. Moreover, the function $\varphi(c, \Delta)$ is strictly monotonically increasing for $\Delta \in (0, \Delta^*]$ and strictly monotonically decreasing for $\Delta \in [\Delta^*, 1)$.*

(Proof of Lemma 2 is given in the Appendix of this chapter.) The shape of the function $\varphi(c, \Delta)$ is schematically illustrated in Figure 3.4, where it can be observed that as Δ increases from 0 to Δ^* , $\varphi(\Delta)$ increases, starting from a positive value. But after Δ^* , as the variable Δ increases, the function value $\varphi|_{\Delta > \Delta^*}$ decreases correspondingly. Then

$$\min_{\Delta^* < \Delta < 1} \varphi(\Delta) = \lim_{\Delta \rightarrow 1} \varphi(\Delta). \quad 3.25$$

In this situation, the asymptotic value of $\lim_{\Delta \rightarrow 1} \varphi(\Delta)$ is critical, that is, if $\lim_{\Delta \rightarrow 1} \varphi(\Delta) > 0$, then the interval S^* is such that $S^* = (0, 2)$.

Table 3.1 was calculated by means of the Gaussian quadrature method using the arbitrary precision package ARPREC [156]. From the numerical results, it can be verified that for any $\Delta \in [0, 0.999]$ and $c \in [1, 90]$, $\varphi(c, \Delta) > 0$ is satisfied. In the following work of development of 2D- and 3D-CPVT model, the c value takes below 200, therefore this numerical result is sufficient for this scheme. Hence the estimation of $S^* = [0.001, 1.999]$ is achieved, where the mapping from the percentage value P_r to the distribution function parameter c is one to one.

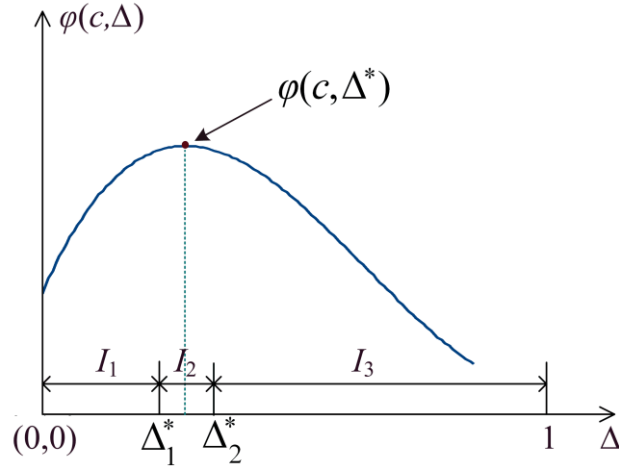


Figure 3.4. Schematic illustration of $\varphi(c, \Delta)$ over the interval of $\Delta \in (0,1)$, where $c \in [1, \infty)$ is a constant value.

Table 3.1. Asymptotic behaviour of $\varphi(c, \Delta)|_{\Delta=0.999}$.

c	φ	c	φ	c	φ	c	φ	c	φ	c	φ
1	5.61E-02	100	2.89E-47	200	8.63E-91	300	2.23E-134	400	6.23E-178	500	1.86E-221
10	6.26E-08	110	1.34E-51	210	3.77E-95	310	9.79E-139	410	2.76E-182	510	8.26E-226
20	5.76E-13	120	6.10E-56	220	1.64E-99	320	4.31E-143	420	1.22E-186	520	3.68E-230
30	1.80E-17	130	2.74E-60	230	7.18E-104	330	1.90E-147	430	5.43E-191	530	1.64E-234
40	7.70E-22	140	1.22E-64	240	3.14E-108	340	8.37E-152	440	2.41E-195	540	7.30E-239
50	5.68E-26	150	5.38E-69	250	1.37E-112	350	3.69E-156	450	1.07E-199	550	3.25E-243
60	3.86E-30	160	2.36E-73	260	6.01E-117	360	1.63E-160	460	4.75E-204	560	1.45E-247
70	2.27E-34	170	1.04E-77	270	2.63E-121	370	7.20E-165	470	2.11E-208	570	6.47E-252
80	1.21E-38	180	4.53E-82	280	1.15E-125	380	3.18E-169	480	9.38E-213	580	2.89E-256
90	6.04E-43	190	1.98E-86	290	5.07E-130	390	1.41E-173	490	4.17E-217	590	1.29E-260

3.4.3 Lower bound on P_r

The preceding discussion shows that given $x_1 = D_L/D_{\text{mean}}$ and $x_2 = D_R/D_{\text{mean}}$ within the interval given by Eq. 3.21, the function value $P_r(c, x_1, x_2)$ is monotonically increasing as c increases. On the other hand, the P_r value cannot be arbitrarily small due to the requirement of modelling grain size distributions, that is, a relaxation condition to determine the lower bound of P_r is derived based on $c \geq 1$. Therefore, the lower bound for the percentage $P_r(c, x_1, x_2)$ is given by,

$$P_r(1, x_1, x_2) = e^{-x_1} - e^{-x_2}. \quad 3.26$$

For the symmetrical case, i.e., $x_1 = 1 - \Delta$ and $x_2 = 1 + \Delta$, the lower bound is determined by

$$P_r(1, \Delta) = e^{\Delta-1} - e^{-\Delta-1}, \quad 3.27$$

where, $\Delta \in [0, 0.999]$.

3.4.4 Algorithm to solve the c value

In [157-159], the c value is determined by means of exhaustive enumeration starting from 1 with a prescribed incremental step, such as 0.001. Since each iterative step involves a series of computations including one numerical integration, this searching scheme is fairly inefficient. To improve the solution procedure, an efficient gradient search method is proposed as follows. Given the four input parameters, D_L , D_{mean} , D_R and P_r , the following equation must be solved for c ,

$$P_r = \int_{D_L/D_{\text{mean}}}^{D_R/D_{\text{mean}}} \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} dx \triangleq F(c), \quad 3.28$$

where the constant $\Gamma(c)$ is calculated by Eq. 3.16. A Newton-Raphson method can be used to obtain the c value as follows. Let

$$f(c) = \int_{D_L/D_{\text{mean}}}^{D_R/D_{\text{mean}}} \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} dx - P_r, \quad 3.29$$

then,

$$f'(c) = \int_{D_L/D_{\text{mean}}}^{D_R/D_{\text{mean}}} \frac{c^c}{\Gamma(c)} x^{c-1} e^{-cx} [1 + \ln(c) - \psi(c) + \ln(x) - x] dx, \quad 3.30$$

and the iterative root finding procedure is

$$c_{i+1} = c_i - \frac{f(c)}{f'(c)}, \quad 3.31$$

where $i = 1, 2, \dots$. Based on the preceding discussion, the denominator is always non-zero. Since all the distributions are described by $c \geq c_0$, the searching process starts at $c = c_0$. This process continues until the prescribed tolerance, ε , is achieved, where the termination condition is formulated in terms of the successive change as

$$|c_{i+1} - c_i| \leq \varepsilon. \quad 3.32$$

3.5 Appendix

3.5.1 Proof of Lemma 1

Consider a function $P_r(c, x_1, x_2)$, where $c \geq 1$ and $0 < x_1 < 1 < x_2$. The partial derivative of P_r with respect to the parameter c is computed as,

$$\frac{\partial P_r}{\partial c} = \frac{\partial}{\partial c} \left(\frac{c^c}{\Gamma(c)} \right) \int_{x_1}^{x_2} x^{c-1} e^{-cx} dx + \frac{c^c}{\Gamma(c)} \frac{\partial}{\partial c} \left(\int_{x_1}^{x_2} x^{c-1} e^{-cx} dx \right).$$

Since $(c^c)' = c^c(1 + \ln c)$,

$$\frac{\partial P_r}{\partial c} = \frac{c^c}{\Gamma(c)} \int_{x_1}^{x_2} x^{c-1} e^{-cx} [1 + \ln c - \psi(c) + \ln x - x] dx. \quad 3.33$$

Let $g(x) = 1 + \ln c - \psi(c) + \ln x - x$. Note that $x^{c-1} e^{-cx} > 0$ is always satisfied for any $x \in S$. Then there is a sufficient condition that if $g(x) > 0$, then $\frac{\partial P_r}{\partial c} > 0$.

For the term $\ln c - \psi(c)$, the inequality of

$$\frac{1}{2c} < \ln c - \psi(c) < \frac{1}{c} \quad 3.34$$

holds (see [160]). Thus, $g(x) > 1 + \frac{1}{2c} + \ln x - x$. To find a valid interval $S^* = (x_1^*, x_2^*)$ such that $g(x) > 0$, two subintervals $S_1^* = (x_1^*, 1]$ and $S_2^* = (1, x_2^*)$ are to be calculated respectively, where $x_1^* = 1 - \Delta_1^*$ and $x_2^* = 1 + \Delta_2^*$.

Note that, $\ln(1 + \Delta) = \Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \dots > \Delta - \frac{\Delta^2}{2}$, where $1 > \Delta > 0$. Thus, for $x_2 = 1 + \Delta \in S_2^*$,

$$\begin{aligned} g(x) &= 1 + \ln c - \psi(c) + \ln(1 + \Delta) - (1 + \Delta) \\ &> 1 + \frac{1}{2c} + \left(\Delta - \frac{\Delta^2}{2} \right) - (1 + \Delta) = \frac{1}{2c} - \frac{\Delta^2}{2}. \end{aligned}$$

So, $\Delta_2^* = c^{-1/2}$ and $S_2^* = (1, 1 + c^{-1/2})$ such that $g(x) > 0$, for $x \in S_2^*$.

Next for the interval S_1^* , the function $g(x)$ is differentiable. Based on Eq. 3.34, $g(1) > \ln c - \psi(c) > 0$, and hence there exists $\Delta > 0$, such that $g(x) > 0$, where $x \in (1 - \Delta, 1]$. Since $\ln(1 - \Delta) = -\Delta - \frac{\Delta^2}{2} - \frac{\Delta^3}{3} - \frac{\Delta^4}{4} - \dots$,

$$\begin{aligned} g(x) &= 1 + \ln c - \psi(c) + \ln(1 - \Delta) - (1 - \Delta) \\ &> 1 + \frac{1}{2c} + \left(-\Delta - \frac{\Delta^2}{2} - \mathcal{O}(\Delta^3) \right) - (1 - \Delta) \\ &= \frac{1}{2c} - \frac{\Delta^2}{2} - \mathcal{O}(\Delta^3). \end{aligned}$$

Thus, for $\Delta \leq c^{-1/2} - \mathcal{O}(\Delta^{3/2})$, $g(x) > 0$ and then $\Delta_1^* = c^{-1/2} - \mathcal{O}(\Delta^{3/2})$.

In summary, $\forall x \in S^* = (x_1^*, x_2^*) = S_1^* \cup S_2^*$, $g(x) > 0$, where

$$\begin{cases} x_1^* = 1 - \Delta_1^* \approx 1 - \frac{1}{\sqrt{c}} + \mathcal{O}(\Delta^{3/2}) \\ x_2^* = 1 + \Delta_2^* \approx 1 + \frac{1}{\sqrt{c}} \end{cases}. \quad 3.35$$

A conclusion, from Eq. 3.33, is the result that for any $(x_1, x_2) \subset S^*$ and $x_1 < 1 < x_2$, $\frac{\partial P_r}{\partial c} > 0$, i.e., $P_r(c, x_1, x_2)$ monotonically increases as c increases.

3.5.2 Proof of Theorem 1

Lemma 1 shows that provided $x_1, x_2 \in S^*$ where $x_1 < 1 < x_2$ and $S^* = (x_1^*, x_2^*)$ defined by Eq. 3.21, $\frac{\partial P_r}{\partial c} > 0$, that is, the percentage value $P_r(x_1, x_2)|_c$ monotonically increases as c increases. Thus, the function $P_r(c)$ is a one-to-one mapping with the domain $D(P_r) = S^*$. Also, $P_r(c)$ has a continuous inverse $P_r^{-1}(c)$ on the range $R(P_r)$.

3.5.3 Proof of Lemma 2

For functions $g_1(\Delta)$ and $g_2(\Delta)$ in Eq. 3.23, the derivatives,

$$\begin{cases} g_1'(\Delta) = \frac{-\Delta}{1-\Delta}, \\ g_2'(\Delta) = \frac{-\Delta}{1+\Delta} \end{cases}, \quad 3.36$$

are negative, that is, they are decreasing on interval $(0,1)$. Moreover, let $\tilde{\Delta}_1, \tilde{\Delta}_2 > 0$ such that Eq. 3.24 is satisfied. Note that, for $c > 1$, there are always $\tilde{\Delta}_1, \tilde{\Delta}_2 \in (0,1)$.

In the following, the property of $\varphi(c, \Delta)$ is studied with three intervals, $I_1 \cup I_2 \cup I_3 = (0, \tilde{\Delta}_1] \cup (\tilde{\Delta}_1, \tilde{\Delta}_2) \cup [\tilde{\Delta}_2, 1)$, respectively. The derivative $\varphi(\Delta)$ with respect to Δ is computed as

$$\begin{aligned} \frac{\partial \varphi(c, \Delta)}{\partial \Delta} = \frac{c^c}{\Gamma(c)} \{ & (1-\Delta)^{c-1} e^{-c(1-\Delta)} \cdot [1 + \ln c - \psi(c) + \ln(1-\Delta) \\ & - (1-\Delta)] + (1+\Delta)^{c-1} e^{-c(1+\Delta)} \cdot [1 + \ln c - \psi(c) \\ & + \ln(1+\Delta) - (1+\Delta)] \}, \end{aligned} \quad 3.37$$

and let

$$\begin{cases} m_1(\Delta) = (1-\Delta)^{c-1} e^{-c(1-\Delta)} \\ m_2(\Delta) = (1+\Delta)^{c-1} e^{-c(1+\Delta)} \end{cases}, \quad 3.38$$

where $m_1(\Delta) > 0$ and $m_2(\Delta) > 0$ for all $\Delta \in (0,1)$. Together with the notation in Eq. 3.23, it has

$$\frac{\partial \varphi(c, \Delta)}{\partial \Delta} = \frac{c^c}{\Gamma(c)} (m_1(\Delta)g_1(\Delta) + m_2(\Delta)g_2(\Delta)). \quad 3.39$$

Obviously, for $\Delta \in I_1 = (0, \tilde{\Delta}_1]$, $g_1(\Delta) \geq 0$ and $g_2(\Delta) > 0$, and hence $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} > 0$. Therefore, $\varphi(\Delta)$ is monotonically increasing on the interval $I_1 = (0, \tilde{\Delta}_1]$. In contrast, for $\Delta \in I_3 = [\tilde{\Delta}_2, 1)$, $g_1(\Delta) < 0$ and $g_2(\Delta) \leq 0$, and thus, $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} < 0$. Thus, $\varphi(\Delta)$ is monotonically decreasing on the interval I_3 .

Consider the property of $\frac{\partial \varphi(c, \Delta)}{\partial \Delta}$ on $I_2 = (\tilde{\Delta}_1, \tilde{\Delta}_2)$, where $g_1(\Delta) < 0$ and $g_2(\Delta) > 0$. $m_1(\Delta)$ and $m_2(\Delta)$ in Eq. 3.38 have relation of $m_1(\Delta) > m_2(\Delta) > 0$, for $\Delta \in (0, \tilde{\Delta}_2)$. From Eq. 3.39, it can be derived that

$$\frac{\partial \varphi(c, \Delta)}{\partial \Delta} < \frac{c^c}{\Gamma(c)} m_1(\Delta)(g_1(\Delta) + g_2(\Delta)).$$

In addition, $g_1(\tilde{\Delta}_1) + g_2(\tilde{\Delta}_1) > 0$ and $g_1(\tilde{\Delta}_2) + g_2(\tilde{\Delta}_2) < 0$. There exists $\bar{\Delta} \in (\tilde{\Delta}_1, \tilde{\Delta}_2)$, such that $g_1(\bar{\Delta}) + g_2(\bar{\Delta}) = 0$. From Eq. 3.36, it has $g'_1(\Delta) < g'_2(\Delta) < 0$, for all $\Delta \in I_2$. Therefore, for $\bar{\Delta} < \Delta < \tilde{\Delta}_2$, $g_1(\Delta) + g_2(\Delta) < 0$, and hence $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} < 0$.

For $\tilde{\Delta}_1 < \Delta \leq \bar{\Delta}$, the second derivative of $\varphi(c, \Delta)$ has

$$\begin{aligned} \frac{\partial^2 \varphi(c, \Delta)}{\partial \Delta^2} &= \frac{c^c}{\Gamma(c)} \cdot (m'_1(\Delta)g_1(\Delta) + m_1(\Delta)g'_1(\Delta) \\ &\quad + m'_2(\Delta)g_2(\Delta) + m_2(\Delta)g'_2(\Delta)). \end{aligned} \quad 3.40$$

Due to

$$m'_2(\Delta) = \frac{-1-c\Delta}{1+\Delta} m_2 < 0,$$

and hence the term

$$m'_2(\Delta)g_2(\Delta) + m_2(\Delta)g'_2(\Delta) < 0 .$$

In addition, the term

$$m'_1(\Delta)g_1(\Delta) + m_1(\Delta)g'_1(\Delta) = m_1(\Delta)\left[\frac{1}{1-\Delta}((1-c\Delta)g_1(\Delta) - \Delta)\right]$$

is negative if and only if

$$\rho(\Delta) = (1 - c\Delta)g_1(\Delta) - \Delta < 0 .$$

Note that, for $\Delta = \bar{\Delta}$, it has $g_1(\bar{\Delta}) + g_2(\bar{\Delta}) = 0$. It can be derived that

$$\ln c - \psi(c) = \frac{\bar{\Delta}^2}{2} + \frac{\bar{\Delta}^4}{4} + \dots.$$

From Eq. 3.34,

$$\frac{1}{2c} < \frac{\bar{\Delta}^2}{2} + \frac{\bar{\Delta}^4}{4} + \dots < \frac{1}{c}.$$

Then,

$$\begin{aligned} \rho(\bar{\Delta}) &= (c\bar{\Delta} - 1)\left(\frac{\bar{\Delta}^3}{3} + \frac{\bar{\Delta}^5}{5} + \dots\right) - \bar{\Delta} \\ &< \left(\frac{2}{\bar{\Delta}} - 1\right)\left(\frac{\bar{\Delta}^3}{3} + \frac{\bar{\Delta}^5}{5} + \dots\right) - \bar{\Delta} \\ &= -\bar{\Delta} + \frac{2\bar{\Delta}^2}{3} - \frac{\bar{\Delta}^3}{3} + \dots . \end{aligned}$$

Thus $\rho(\bar{\Delta}) < 0$. Moreover, since $\rho(\tilde{\Delta}_1) < 0$ and $\rho(\Delta)'' > 0$, it can be obtained that $\rho(\Delta) < 0$, for $\tilde{\Delta}_1 < \Delta \leq \bar{\Delta}$. Therefore, $\frac{\partial^2 \varphi(c, \Delta)}{\partial \Delta^2} < 0$, for $\tilde{\Delta}_1 < \Delta \leq \bar{\Delta}$, that is, $\frac{\partial \varphi(c, \Delta)}{\partial \Delta}$

decreases monotonically on the interval $(\tilde{\Delta}_1, \bar{\Delta}]$. Additionally, $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} > 0$ for $\Delta \in (0, \tilde{\Delta}_1]$ and $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} < 0$ for $\Delta \in (\tilde{\Delta}_1, \bar{\Delta}]$. Therefore, there exist $\Delta^* \in (\tilde{\Delta}_1, \bar{\Delta}] \subset (\tilde{\Delta}_1, \tilde{\Delta}_2)$ such that $\frac{\partial \varphi(c, \Delta^*)}{\partial \Delta} = 0$.

In conclusion, there exist $\Delta^* \in I_2$, such that

1. $\frac{\partial \varphi(c, \Delta^*)}{\partial \Delta} = 0, \Delta^* \in (\tilde{\Delta}_1, \tilde{\Delta}_2)$
2. $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} > 0$, for $0 < \Delta < \Delta^*$
3. $\frac{\partial \varphi(c, \Delta)}{\partial \Delta} < 0$, for $\Delta^* < \Delta < 1$.

Therefore, the function $\varphi(c, \Delta)$ strictly monotonically increases for $\Delta \in (0, \Delta^*]$ and strictly monotonically decreases for $\Delta \in [\Delta^*, 1)$.

3.6 Summary

The fundamental methodology of the CPVT model was discussed in this chapter. This model employs a control parameter, i.e. the minimum seed spacing, to monitor the seed generation process. A virtual grain structure generated using a CPVT model has the properties that its grain size distribution is statistically equivalent to that of the specified physical characteristics, in addition to possessing the appropriate morphology in simulating true grains.

Configuration of a CPVT model requires three steps: 1) defining distribution parameter from the input of physical parameters, 2) deriving the tessellation's regularity from the distribution parameter, and 3) determining the control parameter from the regularity. In CPVT models, grain size distribution is described using a one-parameter gamma distribution function. The relationship between the proposed physical parameters and the gamma distribution parameter has been studied. A mathematical proof of the uniqueness of the determination of the distribution parameter from the physical parameters was provided by identifying a small valid interval, which was then extended into a larger region that fully satisfies the application requirements.

Furthermore, a series of computational results were presented to validate the uniqueness assumption of the proposed scheme. In addition an efficient numerical procedure was proposed for computing the distribution parameter. The other two steps are to be fulfilled in the following chapters with respect to the dimension of grain structures.

Chapter 4

2D CPVT model

4.1 Introduction

In this chapter, the CPVT model is developed to generate two-dimensional virtual grain structures. A global quantity to be used to evaluate a tessellation's uniformity is given and its dual relations with control parameters and distribution parameter are presented. In combination with the results of the preceding chapter, a 2D-CPVT model has been constructed. Implementation of the entire system is explained and the corresponding computational procedures are provided.

Emphasis is devoted to examining the effectiveness and robustness of the CPVT system in generating virtual grain structures with specified properties. Two series of statistical tests were performed to validate the agreement between the prescribed regularity and that of the resultant tessellations, and to investigate the details of the overall grain size distribution. In addition, two real microscopic images with different grain size distribution features were employed to examine the capability of the system to generate virtual grain structures that meet physical measurements. To demonstrate applications of the proposed 2D-CPVT model in micromechanics simulations and further explain the virtual grain structure generation procedures, two virtual grain structures with different regularities were generated and related CPFE models have been built for 2D plane strain CPFE analyses of uniaxial tension.

4.2 Model development

4.2.1 Workpiece related input

Representation of the FE model by means of a Voronoi tessellation requires prescription of a workpiece domain Ω and the number of seeds, N_{seed} . As shown in

Figure 3.2, there are two groups of input required to configure a CPVT model: the workpiece and the grain structure related. The workpiece related input defines the domain of virtual grain structure and the number of seeds to be generated within that domain.

There are two schemes in generating the grain structure. One is to prepare a sufficiently large tessellation and cut a region from it having a specific dimension, i.e., it accounts for the influence of outer seeds on grains cut by the region boundary. In contrast, the other scheme is to generate grains within the given domain, in which the effect of outside seeds is ignored. Figure 4.1 shows tessellations with and without external seeds. Neglecting neighbouring seeds, the tessellation in Figure 4.1 (b) has lost approximately half of the grains cut by boundaries, presented in Figure 4.1 (a). However the total area of the grains cut by boundaries is correspondingly increased in Figure 4.1 (b) by a factor of ~ 2 .

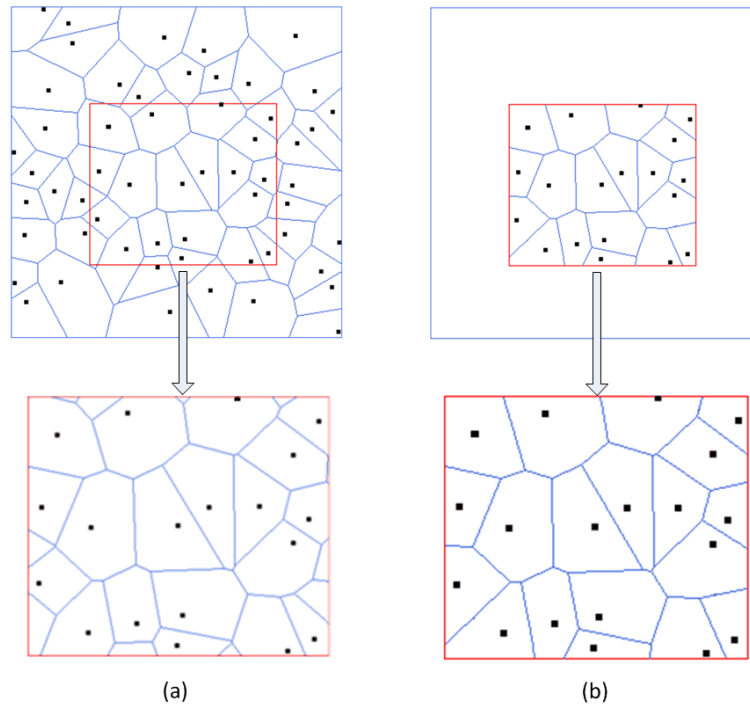


Figure 4.1. Illustration of the effect of external seeds on the number of grains. (a) A case accounting for external seeds; (b) a case neglecting external seeds.

In quantitative metallography, the number of grains in a region is based on the heuristic rule that grains cut by the region boundary are counted as half grains, which corresponds to the first scheme. Note that, ignoring the outer seeds in the second scheme results in nearly half of the grains disappearing. In such a situation, grains cut by boundaries can be approximately treated as whole grains rather than half grains. Consequently, for a VT obtained by the second scheme, the number of seeds N_{seed} to be generated is equivalent to the number of grains N , that is,

$$N_{\text{seed}} \approx N. \quad 4.1$$

In addition, the number of grains is calculated by means of the tessellation area A_0 and the mean grain size (in terms of grain area) D_{mean} , i.e.,

$$N = A_0 / D_{\text{mean}}. \quad 4.2$$

Note that, as a convention, to develop a quantitative mechanism for generating VTs, the second scheme is employed in the CPVT model, i.e., provided a mean grain size D_{mean} , N_{seed} seeds are generated in the given domain A_0 .

4.2.2 Regularity and control parameter

Whilst Voronoi tessellations are capable of capturing natural variance of grain shape and organisation, a quantitative evaluation of the degree of uniformity is critical. It is important to note, based on the comparison results of the geometrical properties of space-filling regular polygons presented in Table 2.1, the hexagon has the minimum value of edge length per unit area. Therefore, a regular hexagonal tessellation (or a honeycomb) can be deemed as the most regular tessellation among those of the same mean grain size.

For a regular tessellation (RT), consisting of regular hexagons, generated in the domain A_0 , the size of an individual hexagon is D_{mean} and the distance between any two adjacent seeds is given by [154]

$$d_{\text{reg}} = \sqrt{\frac{2}{\sqrt{3}} D_{\text{mean}}}. \quad 4.3$$

Since this RT is a fully ordered VT, and to construct a VT with a mean grain size D_{mean} in the area A_0 , the minimum distance between adjacent seeds δ should be less than the regular distance d_{reg} . In [154], Zhu et al. introduced a non-local regularity parameter, α , to evaluate a VT obtained by the second scheme of counting mean grain size. This regularity α of a VT is defined by:

$$\alpha = \delta/d_{\text{reg}}. \quad 4.4$$

For a RT, $\delta = d_{\text{reg}}$ and $\alpha = 1$, while for any other irregular VT, δ is less than d_{reg} . As tessellations become more disordered, the regularity δ decreases; if $\delta = 0$, the tessellation corresponds a completely random tessellation, i.e the Poisson Voronoi. Figure 4.2 shows a series of tessellations with varying regularity values.

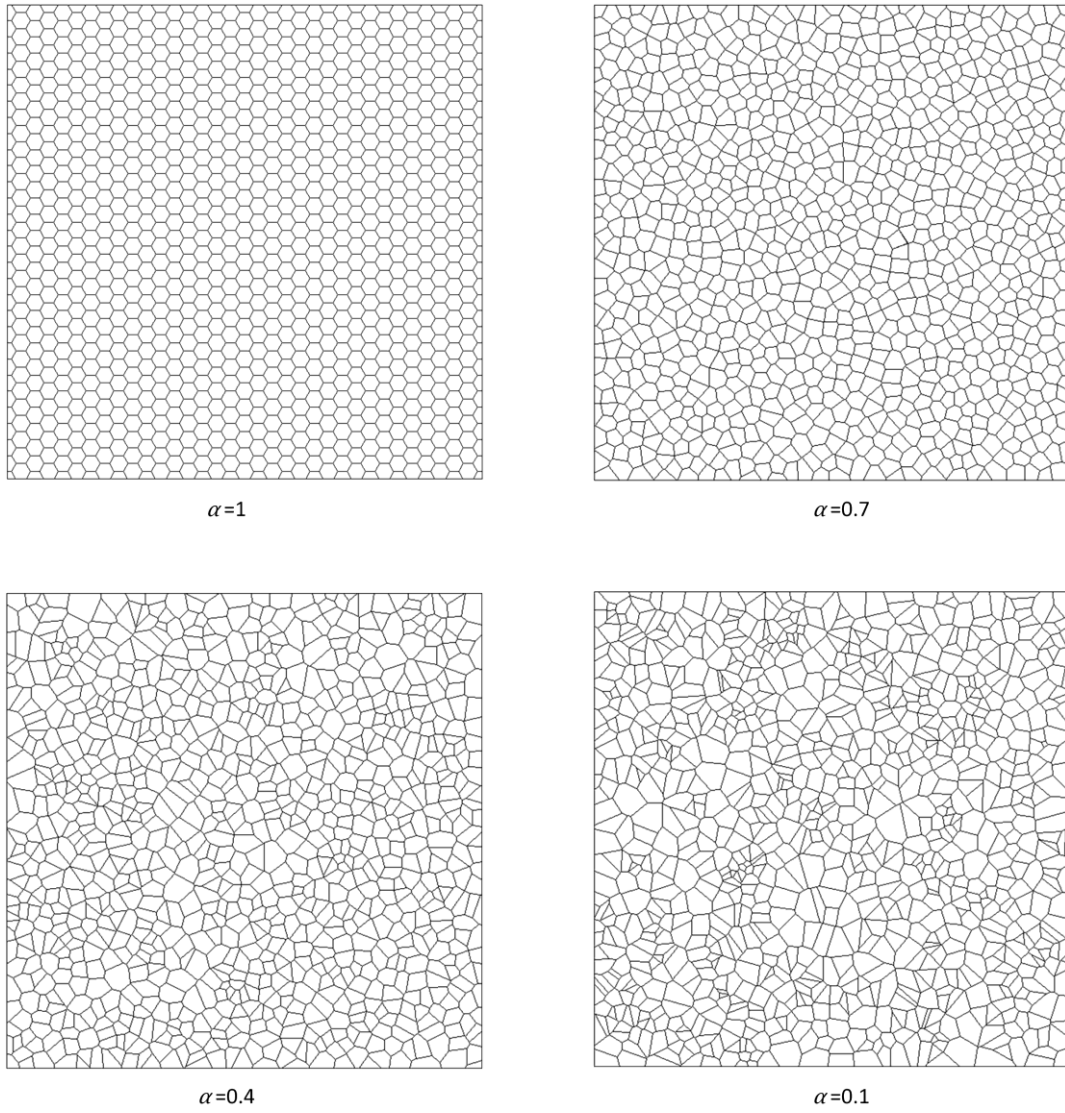


Figure 4.2. Voronoi tessellations with different regularities.

The regularity parameter was initially used to evaluate a tessellation's uniformity in terms of the minimum seed distance. This parameter could alternatively be used to monitor the seed generation process, and hence to produce a VT with a particular regularity. This mechanism is realised by introducing a control parameter δ , which reformulates the meanings of the minimum seed distance as

$$\delta = \alpha d_{\text{reg}}. \quad 4.5$$

The regularity parameter α must be derived to specify the control parameter δ (Eq. 4.5). But in practice, it is physical measurements from quantitative metallography rather than this abstract regularity parameter that are well known and widely used by engineers and scientists. In such a circumstance, the regularity parameter is too obscure for practical use. In the next section, the relationship between the regularity, α , with the distribution parameter, c , of one-parameter gamma distribution function is presented. Together with the mapping from the physical parameter to distribution parameter, the 2D-CPVT model is able to employ physical parameter input to generate virtual grain structure with statistically equivalent grain area distribution.

4.2.3 Regularity and distribution parameter

As preceding discussion, a one-parameter distribution function (Eqs. 3.15-3.16) is used to represent a normalised grain size distribution in terms of grain area for 2D grain structure. Note that the variance of a one-parameter gamma distribution is $1/c$. As the distribution parameter c increases, the distribution function becomes narrower, which is more suitable for modelling regular VTs; smaller c values are more suitable for irregular VTs. In addition, there are two major advantages to using the one-parameter gamma distribution to describe the grain area distribution: 1) Only one parameter, c , is involved in the relation with the regularity parameter δ , and 2) the mean value of this distribution function is 1, which makes the distribution capable of modelling the normalised grain area distribution of a VT.

In [154], a series of statistical tests have been conducted to summarise the relationship between the fitting parameter c and the regularity α , where α ranges from 0 to 0.8. In [157] and [158], a descriptive model was proposed based on statistical data provided in [154]:

$$\alpha(c) = A(z(c) - z_0)^{k+nz(c)}, \quad c_0 \leq c \quad 4.6$$

where $z(c) = c/c_m$, $z_0 = c_0/c_m$, $c_0 = 3.555$, $c_m = 47.524$, $A = 0.738895$, $k = 0.323911$ and $n = -0.414367$.

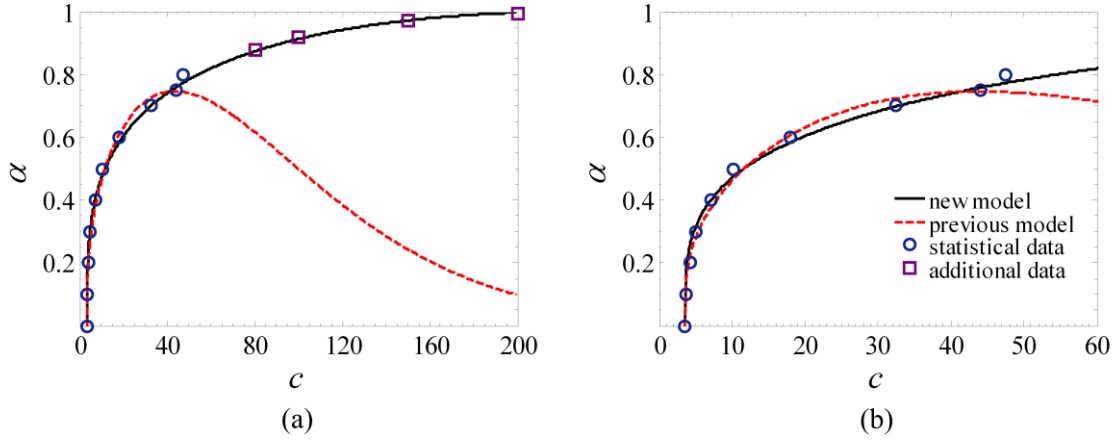


Figure 4.3. Comparison of the two descriptive models, both of which describe the relationship between the grain area distribution parameter c and the regularity parameter α . The dotted line is the model proposed in [157] and [158], and the solid line is the improved model: (a) Global quality; (b) local quality.

The dotted lines in Figure 4.3 show that this model provides good predictions for $c < 30$. However, as c increases beyond 40, the related regularity α decreases, which fails to describe the tendency of the tessellation's regularity to increase as the distribution function becomes narrower. In order to fix this problem, a set of newly calibrated values are provided. This has been done by adding four supporting data points to the experimental data set to aid in formulating a sound objective function for optimisation. The CMA-ES algorithm [161] is applied here to find the model's parameters. The new parameter values are $c_0 = 3.555$, $c_m = 230$, $A = 0.98$, $k = 0.2$ and $n = -0.3$. In Figure 4.3, the new result of $\alpha(c)$ is given by the solid lines. Figure 4.3 (a) compares the overall quality of both descriptions. It can be seen that the newly proposed model provides a consistent description of the relationship between the regularity and the distribution parameter, i.e., $\alpha(c)$ is monotonically increasing. In contrast, the previous model failed to capture this feature. Figure 4.3 (b) also reveals that the fit quality of the new model is as good as the previous model for small c values.

4.2.4 Implementation

The 2D-CPVT model is the two dimensional implementation of the general CPVT model proposed in Chapter 3, as the flow chart shown in Figure 3.2. There are two groups of use-inputs: a workpiece related input and the grain structure related input. The implementation procedure is explained as follows:

The workpiece related parameters are input to the model, including the domain Ω and the mean grain size D_{mean} . This is followed by calculation of:

- a. The number of seeds N_{seed} (using Eq. 4.1 and Eq. 4.2);
- b. The regular distance d_{reg} (using Eq. 4.3);

The physical parameters are input, followed by operations to:

- a. Compute the distribution parameter c by the Newton-Raphson method (using Eqs. 3.28-3.32).
- b. Derive the regularity parameter α from the empirical model (using Eq.4.6).
- c. Combine c and α with the obtained regular distance d_{reg} to compute the control parameter δ (Eq.4.5).

In summary, a virtual grain structure generation process can be implemented when given a domain Ω with area A_0 and a mean grain size D_{mean} , the number of seeds N_{seed} , the regular distance d_{reg} and the control parameter δ are correspondingly defined by Eqs. 4.1-4.5. Seeds are to be generated sequentially in this domain by sampling their x and y coordinates independently based on a uniform random number generator. It is assumed that a newly generated seed can only be accepted if its distance from any existing seeds is greater than or equal to the control parameter δ . The resultant Voronoi tessellation is referred to as a two-dimensional controlled Poisson Voronoi tessellation (2D-CPVT) with the control parameter δ , also denoted by $\text{CPVT}(\delta|\Omega, D_{\text{mean}})$. As a seed lattice is generated from the 2D-CPVT model the corresponding VT geometry can be constructed using the results of Section 3.2.1.

Note that, if $\delta = 0$, the 2D-CPVT is reduced to a planar Poisson Voronoi tessellation, and if $\delta = 1$, the 2D-CPVT is a regular hexagonal tessellation with grain area D_{mean} . In addition, a tessellation generated by the 2D-CPVT model has a regularity value that converges to the prescribed α with a small and allowable asymptotic error. When sampling a small number of seeds, the resultant VT tends to be slightly more regular than the prescribed regularity. This effect is to be discussed later.

4.3 Model validation

4.3.1 Statistical analysis

The CVPT model works in sequence, mapping the user-defined physical parameters to the distribution parameter c , then from the distribution parameter c to the regularity α , and finally from the regularity α to the control parameter δ . VTs are subsequently produced. The ultimate objective is to produce VTs with two properties: 1) Regularity of a generated tessellation similar to the value derived from the physical parameters input, and 2) the mean grain size and the overall grain size distribution of the tessellation conforming to the gamma distribution derived from the physical parameters. It is important to note that the equation linking the obtained tessellation's regularity to the distribution parameter is based on statistical data from virtual grain structures. Additionally, there is an asymptotic error between the regularity value derived from the physical parameters input and the resultant value of a VT. Thus, a comprehensive investigation of the effectiveness and robustness of the CPVT model for regularity control and grain size distribution control is necessary.

The number of grains in an individual tessellation is a factor that strongly influences evaluation of the statistical results. For a VT with fewer grains, statistical variation is greater. Therefore, all statistical tests were conducted for 200, 1000 and 5000 grains, respectively. The mean grain size used for all tessellations was $10 \mu\text{m}^2$. For the VT having 200 grains, the domain Ω was $40 \mu\text{m} \times 50 \mu\text{m}$; for the VT having 1000 grains, Ω was $100 \mu\text{m} \times 100 \mu\text{m}$; and for the VT having 5000 grains, Ω was $200 \mu\text{m} \times 250 \mu\text{m}$. The

following statistical results were all based on 1000 independent generations of VTs for each individual case.

4.3.1.1 Regularity analysis

Regularity of a VT is determined by its minimum seed distance and mean grain size. In the 2D-CPVT model, regularity is employed as an intermediate parameter to derive the control parameter δ . The small asymptotic error in the regularity mentioned previously is to be investigated in this section. In order to compare the difference between an ideal regularity and the regularity of a resultant VT, a relative error ε is defined as

$$\varepsilon = \frac{\hat{\alpha} - \alpha}{\alpha} = \frac{\hat{d}_{\min} - \delta}{\delta} \quad 4.7$$

where $\hat{\alpha}$ and \hat{d}_{\min} are resultant values of the regularity and minimum seed distance of a VT, respectively. To estimate the accuracy of the regularity, the mean and standard error of ε , given by

$$\bar{\varepsilon} = \sum_{i=1}^n \varepsilon_i, \quad 4.8$$

$$se = \sqrt{\frac{\sum_{i=1}^n (\varepsilon_i - \bar{\varepsilon})^2}{n(n-1)}}, \quad 4.9$$

respectively, are calculated for a set of $n = 1000$ independent runs.

Values of $\bar{\varepsilon}$ and se were calculated for a range of regularity values and for three different numbers of grains and a fixed domain, as described previously, shown in Table 4.1 and Figure 4.4. It can be observed that for a VT having 1000 or more grains, $\bar{\varepsilon}$ is less than 3%, which can be ignored in practical usage. For a VT that has roughly 200 grains, if $\alpha \leq 0.25$, the mean asymptotic error $\bar{\varepsilon}$ becomes significant, greater than or equal to 5%. This occurs because when the regularity is sufficiently small, the value of the control parameter is significantly decreased, hence the acceptable interval of grain seed distances is enlarged. Furthermore, given a certain number of randomly generated seeds,

the larger the domain, the more uncertainty there will be in the regularity. In all cases, the standard error of ε is very small, indicating that the error ε is consistent from one generation to the next.

Table 4.1. Statistical results for the mean value of relative error $\bar{\varepsilon}$ and the standard error, where the mean grain size for each VT was chosen to be $10 \mu\text{m}^2$

α	$N_{\text{seed}} = 200$		$N_{\text{seed}} = 1000$		$N_{\text{seed}} = 5000$	
	$\bar{\varepsilon}$	se	$\bar{\varepsilon}$	se	$\bar{\varepsilon}$	se
0.1	1.01E-01	3.36E-03	2.45E-02	7.54E-04	5.30E-03	1.60E-04
0.2	3.13E-02	9.96E-04	6.23E-03	2.02E-04	1.32E-03	4.16E-05
0.3	1.23E-02	3.96E-04	2.57E-03	7.87E-05	5.59E-04	1.77E-05
0.4	6.55E-03	2.07E-04	1.26E-03	4.01E-05	2.55E-04	7.56E-06
0.5	3.38E-03	9.89E-05	6.60E-04	1.97E-05	1.25E-04	3.96E-06
0.6	1.61E-03	4.97E-05	2.85E-04	8.91E-06	5.50E-05	1.80E-06
0.7	4.83E-04	1.56E-05	7.53E-05	2.33E-06	1.29E-05	4.33E-07

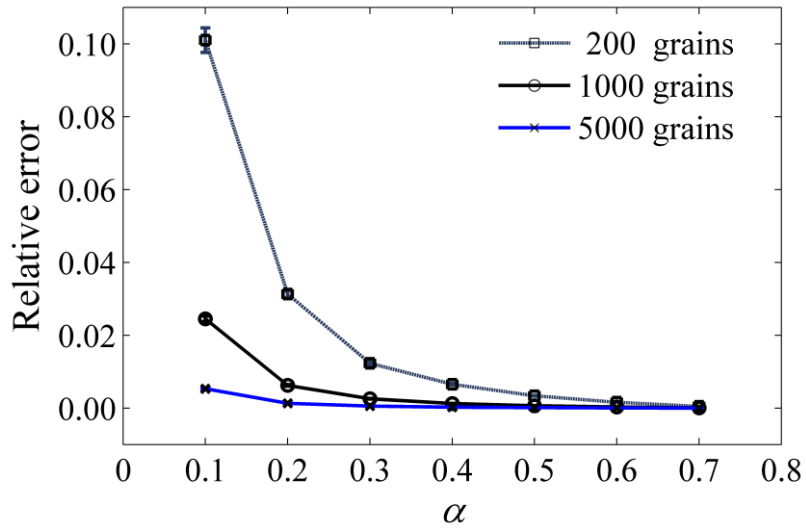


Figure 4.4. The relative error of the ideal regularities and those of generated VTs. Error bars represent 95% confidence intervals.

The following facts about the effectiveness of the CPVT model for regularity control can be deduced from these results: 1) For a large number of grain (> 1000), the CPVT regularity control is very accurate *and* precise for all prescribed regularity values; 2) for a small number of grains, the CPVT regularity control is very precise for all prescribed regularity values, but is only very accurate for $\alpha > 0.25$; 3) for any grain size, both the accuracy *and* precision are improved for larger prescribed regularity values. Overall, the CPVT model provides excellent accuracy *and* precision in the regularity control for tessellations either having large numbers of grains or highly regular distributions.

4.3.1.2 Distribution analysis

The other key feature of the CPVT model is automatic generation of VTs with grain size distribution control. The following results are presented to identify the quality of the grain size distribution control of the 2D-CPVT model. Results are shown for a range of regularity values, from which the distribution parameters can be found according to the descriptive model given by Eq. 4.6. Figure 4.5 summarises the grain size distributions of the generated VTs; the bar graphs are the ideal grain size distributions, and the data points and error bars give the mean and standard deviation respectively of the VT output based on 1000 generations with the same control parameters. The standard deviations demonstrate the degree of variation for the grain size intervals shown, which allows comparison between ideal and generated grain structure local distribution characteristics for the grain size intervals shown.

VTs with greater than or equal to 1000 grains exhibit nearly perfect agreement, in terms of generated grain size accuracy and precision per grain size interval, with the ideal distributions for regularity values of $\alpha = 0.1$ and $\alpha = 0.7$; the remaining generated cases shown in Figure 4.5 are somewhat less accurate and/or precise compared to the ideal distributions. Notably, the distribution with 200 grains is very accurate in terms of mean grain size per interval but has a very large variability from one independent generation to the next. By contrast, the 5000 grains cases show little variability, but the accuracy is somewhat inferior for $\alpha = 0.3$ and $\alpha = 0.5$. One reason for the inferior results for $\alpha = 0.3$ and $\alpha = 0.5$ is the stiffness of the $\alpha(c)$ curve given by Eq. 4.6 for regularity

values less than 0.5. This can amplify stochastic errors and even very small inaccuracies in the fitting model given by Eq. 4.6. That is, a small noise may lead to appreciable lack of correlation between the c and α values. The statistical results of Figure 4.5 (f), (g), (j) and (k) suggest the value of the distribution parameter c is consistently too large, hence an improved descriptive model relating c to α should give a slightly smaller c value for a given value of α compared to the proposed model. In the case of 200 grains, and $\alpha = 0.7$, the relative inaccuracy is a result of the significance of the way in which grains cut by boundaries are counted, which influences the mean grain size.

4.3.2 Comparison of virtual and true grain structures

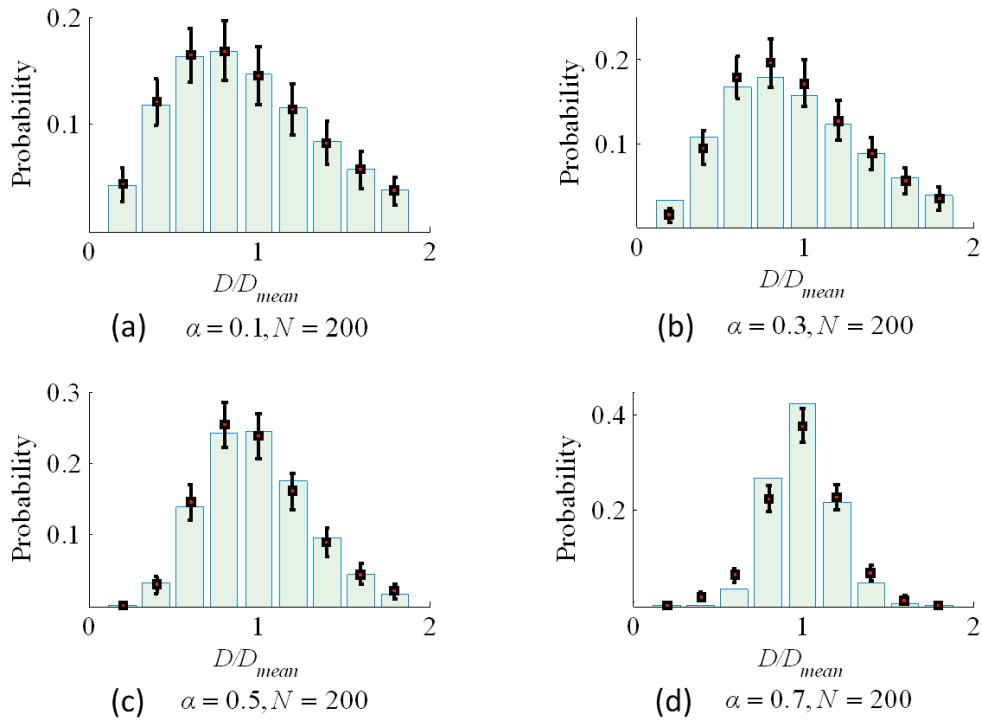
A process of applying the 2D-CPVT model to generate virtual grain structures is demonstrated in this section. SEM observations of the grain structures of commercial aluminium alloys were used to extract the necessary physical properties, shown in Figure 4.6 (a) and Figure 4.7 (a). The workpiece related input to the 2D-CPVT model was taken directly from the domains of the images, while the physical parameters were identified by image processing including digital filtering and enhancement, segmentation of the grains by a watershed algorithm, and measuring the grain areas using pixel information. The images after segmentation are presented in Figure 4.6 (b) and Figure 4.7 (b), which clearly show the grain shapes and boundaries. The input parameters for the CVPT model test are listed in Table 4.2.

Table 4.2. Physical parameters and corresponding grain structure properties for the two grain structures.

Label	Physical parameters (μm^2)				Equivalent grain size [†] (μm)			Model parameters		
	D_{mean}	D_L	D_R	P_T	d_{mean}	d_L	d_R	d_{reg}	α	δ
Irregular	4078	2039	6117	75%	68.6	48.5	84.0	68.6	0.265	18.2
Regular	2632	1579	3685	80%	55.1	42.7	65.2	55.1	0.453	25.0

[†] Assuming a hexagonal grain shape.

Virtual grain structures were generated for the regular and irregular cases, as shown in Figure 4.6 (c) and Figure 4.7 (c), with histograms of grain size shown for the ideal and virtual cases in Figure 4.6 (d) and Figure 4.7 (d). As noted previously, repeated application of the same physical parameters using the CPVT model will result in slightly different grain structures, but all the virtual grain structures are statistically consistent with the specified physical parameters. The orientations of grains can be assigned based on a fixed texture, or be defined by a random number generator based on a uniform distribution or a normal distribution. Grain orientations can also be defined according to measurements, such as EBSD. In this case, grain orientations were created by a normal distribution random number generator. Figure 4.6 (e) and Figure 4.7 (e) show the resultant FE models with virtual grain structures and grain orientations assigned by a uniform random number generator. In addition, corresponding meshing results are given in Figure 4.6 (f) and Figure 4.7 (f). Further details of CPFE simulations carried out using the VGRAIN system have been described in [140]



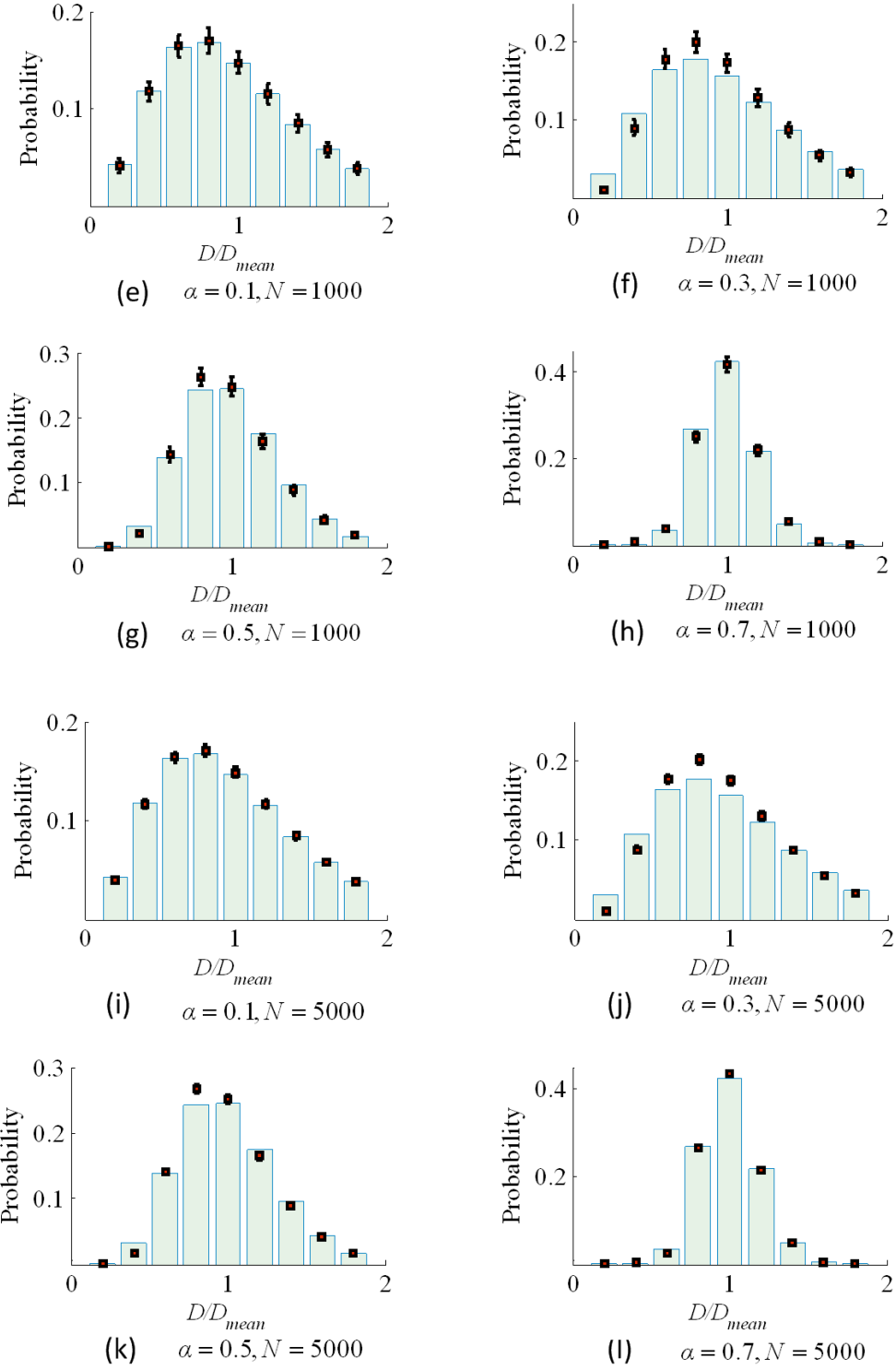


Figure 4.5. Grain size distributions of VTs with different regularities and numbers of grains. Error bars with standard deviation highlight the dispersion of frequency variation of local grain distributions from generated VTs.

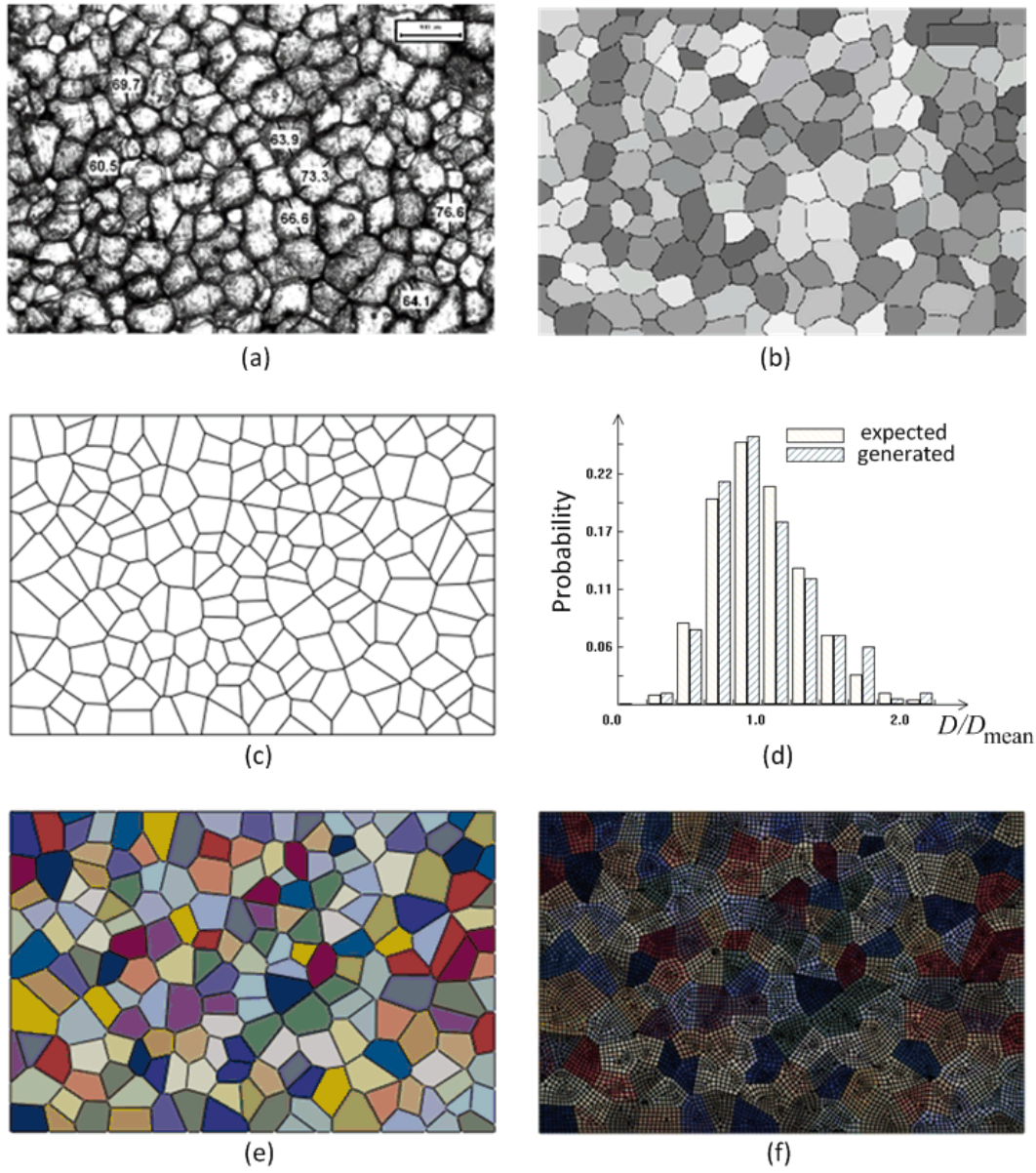


Figure 4.6. CPFE simulation for a regular grain structure. (a) Original microscopic image (domain is $846 \mu\text{m} \times 563 \mu\text{m}$, with $100 \mu\text{m}$ scale shown); (b) the result after image processing where the grain structure was segmented into 181 grains; (c) the sampled virtual grain structure; (d) the grain size distribution of the virtual structure (a histogram of the virtual structure is plotted by shaded bars, and the ideal distribution given by the remaining bars); (e) a FE model with assignment of grain orientations; (f) the FE model after meshing.

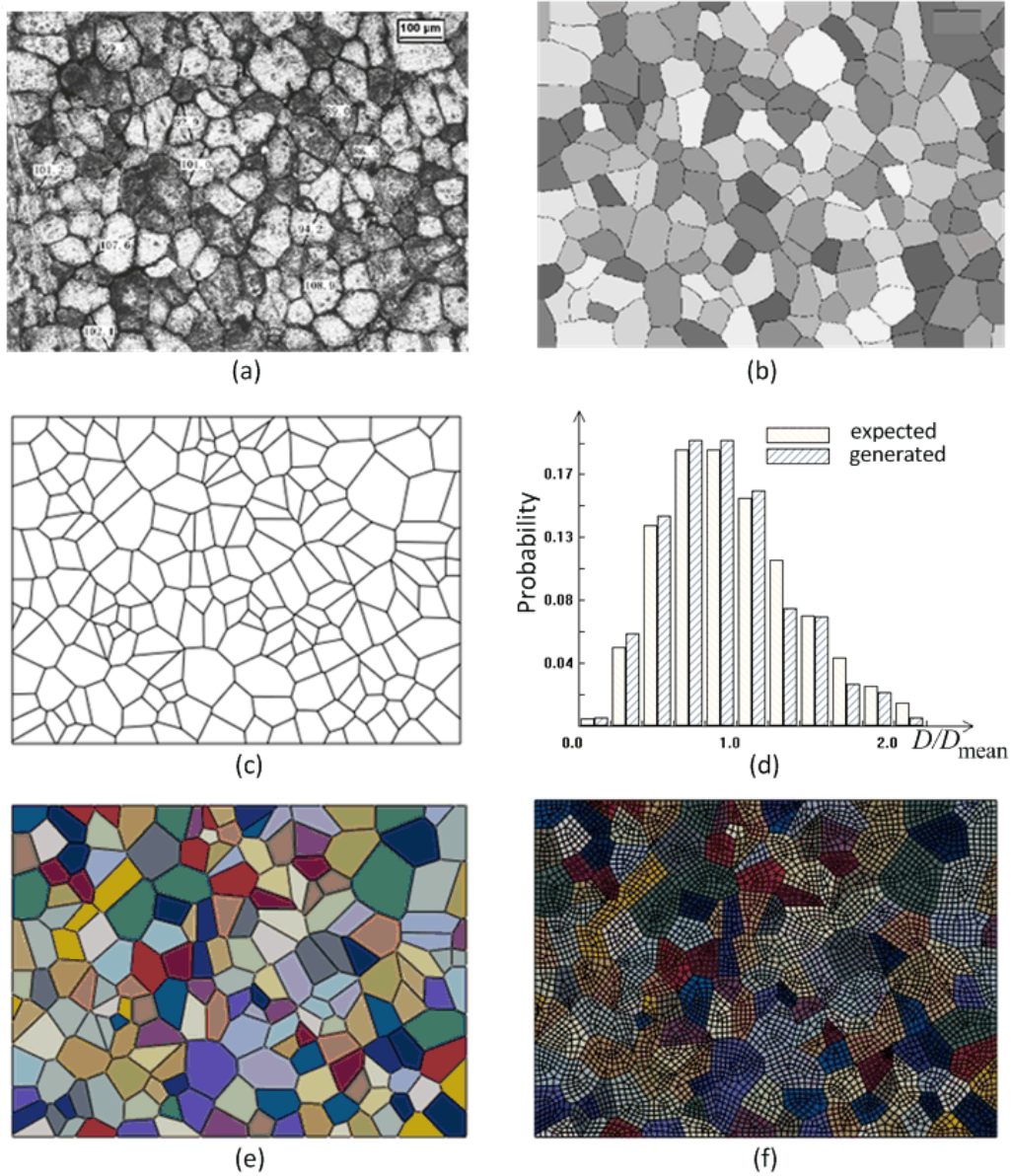


Figure 4.7. CPFE simulation for an irregular grain structure. (a) Original microscopic image (domain is $1000 \mu\text{m} \times 730 \mu\text{m}$, with $100 \mu\text{m}$ scale shown); (b) the result after image processing where the grain structure was segmented into 179 grains; (c) the sampled virtual grain structure; (d) the grain size distribution of the virtual structure (a histogram of the virtual structure is plotted by shaded bars, and the ideal distribution given by the remaining bars); (e) a FE model with assignment of grain orientations; (f) the FE model after meshing.

4.4 CPFE application

To facilitate grain structure generation and micro-mechanics modelling, an integrated system called VGRAIN has been developed and used in this work to build materials microstructures, where the proposed 2D-CPVT model is applied to generate virtual grain structures. The integrated process of defining the microstructure of the material is illustrated in Figure 4.8.

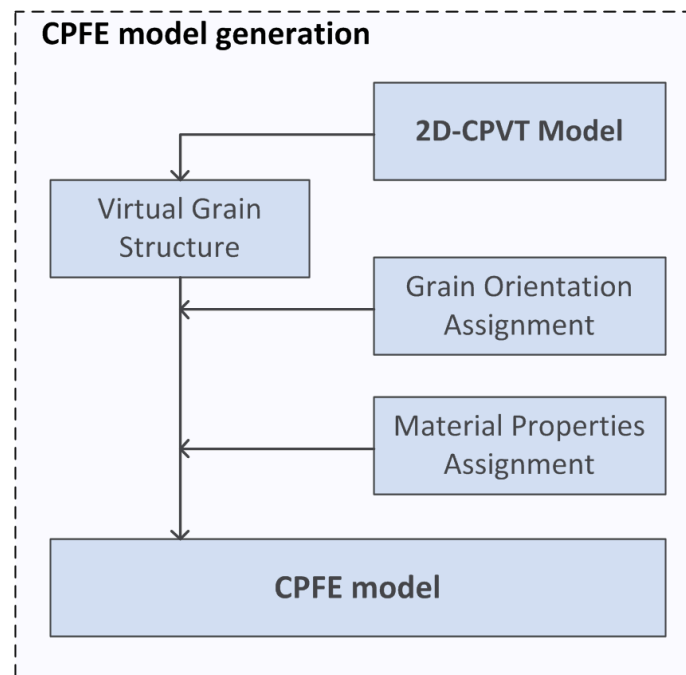


Figure 4.8. The procedure for generating a CPFE model.

Two more modules have also been developed to include grain orientation and material properties assignment. The orientation of each grain can be assigned based on a fixed texture, or be set by a random value from a random number generator based on a uniform distribution or a normal distribution. Grain orientations can also be defined according to measurements, such as EBSD. In the VGRAIN system, the generated grain structure together with the grain orientations can be directly imported into commercially available FE codes, e.g., ABAQUS/CAE, for further pre-processing operations, such as meshing, boundary and loading conditions defined based on the simulation requirements.

The details about the VGRAIN system development will be presented in Chapter 8. Crystal plasticity constitutive equations as in Wang et al. [18] were implemented in the commercial FE code ABAQUS/EXPLICIT through the user defined material subroutine VUMAT. The detailed implementation and CPFE analysis procedures are presented in Chapter 6.

To demonstrate the proposed CPVT model and material properties definition procedure in the VGRAIN system, 2D plane strain CPFE analyses of uniaxial tension have been performed. It is worth noting that singly-, doubly- or triply-periodic boundary conditions can be prescribed for representative volume elements with grain structures generated by VGRAIN, although this has not been automated in the present version to maintain generality. The crystal plasticity calibration of [6] for 316L stainless steel is used here. The crystal plasticity is not the focus of this study; details of the calculations can be found in [18]. The overall dimensions of the microfilm workpiece are $100\ \mu\text{m} \times 30\ \mu\text{m}$ and a displacement of $U = 80\ \mu\text{m}$ is applied at the right edge of the model and the lateral faces are free of constraint. The overall setup is shown in Figure 4.9.

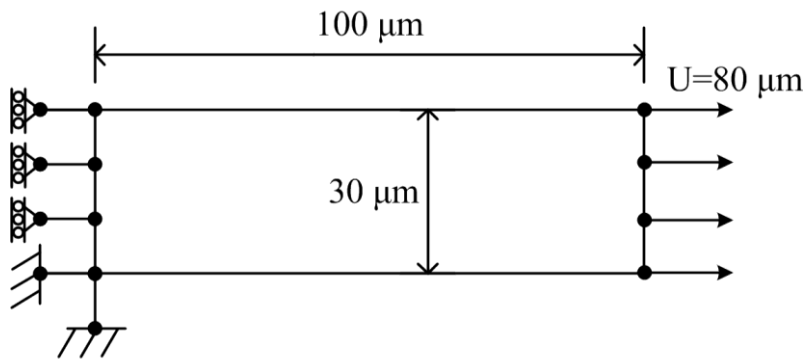


Figure 4.9. Schematic diagram for the plane strain CPFE model.

Two specimens were generated with different regularities: One was much more regular than the other. The physical parameters and the corresponding distribution parameters from the CPVT model are given in Table 4.3. Both microstructures have the same mean grain size of $50\ \mu\text{m}^2$, in terms of area, that is, the equivalent grain diameter

is approximately 8 μm assuming a hexagonal grain shape. By assigning different physical parameters, different regularities were achieved using the CPVT model. The simulated irregular and regular grain structures are presented in Figure 4.10 (a) and Figure 4.11 (a), respectively. It should be noted that repeated application of the same physical parameters using the CPVT model will result in slightly different grain structures, but all the simulated grain structures have the feature that the grain sizes are statistically conformal with the specified physical parameters.

Table 4.3. Physical parameters and corresponding grain structure properties for the two CPFE models.

Label	Physical parameters (μm^2)				Equivalent grain size [†] (μm)			CPVT model parameters			
	D_{mean}	D_L	D_R	P_r	d_{mean}	d_L	d_R	d_{reg} (μm)	α	δ (μm)	N
Irregular	50	20	80	79%	7.6	4.8	9.6	7.6	0.156	1.18	60
Regular	50	30	70	95%	7.6	5.9	9.0	7.6	0.666	5.06	60

[†] Assuming a hexagonal grain shape.

Figure 4.10 and Figure 4.11 show the CPFE simulation procedures for both irregular and regular microstructures. Every simulation involves three major steps: definition of the grain structure as shown in (a)s, assignment of material properties, mesh and boundary conditions, as illustrated in (b)s, and post-processing and analysis as in (c)s. The deformed models are presented in (c)s, where contours of accumulated plastic strain (ε_p) are shown and the necking features of both models can be observed.

Figure 4.12 correlates the necking regions with the original microstructures. It was found in [18] that necking caused by strain localisation is increasingly prohibited by a greater number of grains through the specimen width. Although strain localisation necking is delayed until greater applied strain when the number of grains through the specimen width is greater, the onset of necking in that case was demonstrated to be sudden with rapid progression towards a high level of thinning. The grain structures shown in Figure 4.10 (a) and Figure 4.11 (a) have the same average grain size. However,

the regular grain structure has approximately the same number of grains through the specimen width at any location along the length, whereas the irregular grain structure has between two and seven grains through the width. Although necking by strain localisation was shown in [18] to occur in locations with fewer grains through the width, the situation shown here is distinctly different in that only regular grain patterns were simulated in [18]. Although in the case of an irregular grain pattern there may be a location with relatively few grains through the specimen width, strain localisation necking also requires that those grains have crystallographic orientations favourable to slip, that is, there must be at least one slip system available having a relatively high Schmid factor under the applied loading. For a specimen with a regular grain pattern having, for example, an average of two grains through the specimen width, there is statistically a much higher (depending on the specimen aspect ratio) likelihood there will be a location along the length with grains oriented such that they are favourable to slip, compared to a specimen with an irregular grain pattern having a higher mean grain size but also one or more locations along the length with two grains through the specimen width.

Hence, there is a greater statistical variation in the necking response for the film shown in Figure 4.12 (c) and (d) than for that in (a) and (b). The necking zones (NZ) shown in Figure 4.12 indicate that, for the strain shown which is quite extreme, the NZ with more grains through the width, given by (a) and (b), actually necked more than the NZ give by (c) and (d). However, this can be understood by the previous discussion in that either the grains in the NZ of the irregular grain pattern were oriented in a way not favourable to crystallographic slip, compared to those in the NZ of the regular grain pattern, or, the level of applied strain was sufficient for the sudden onset of necking and rapid thinning demonstrated in [18] to occur in the regular grain pattern such that it overcame the more gradual necking in the irregular grain pattern (in the latter case, the degree of necking in the irregular grain pattern would have been greater than that in the regular grain pattern for some smaller value of applied strain).

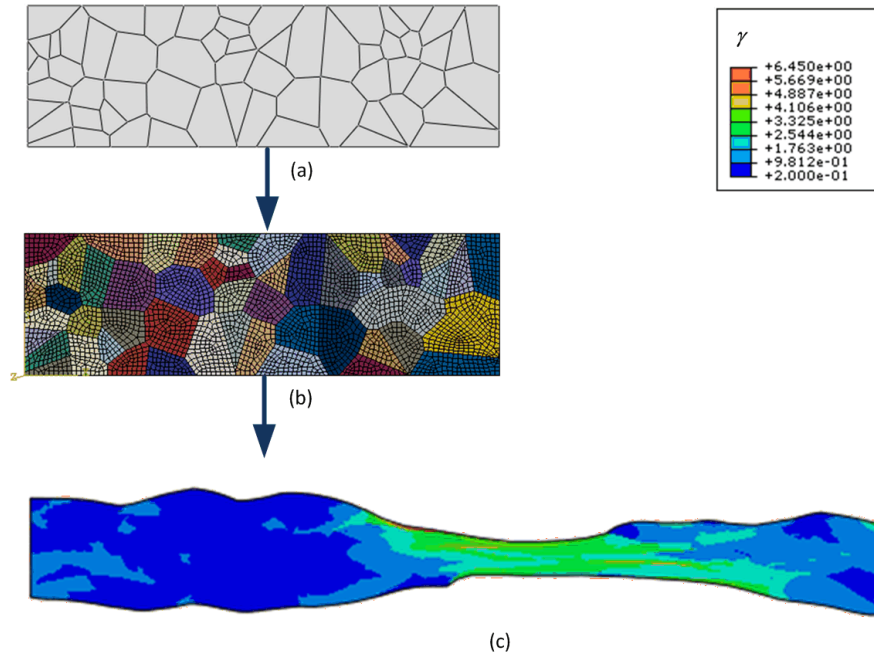


Figure 4.10. CPFE simulations for an irregular grain structure.

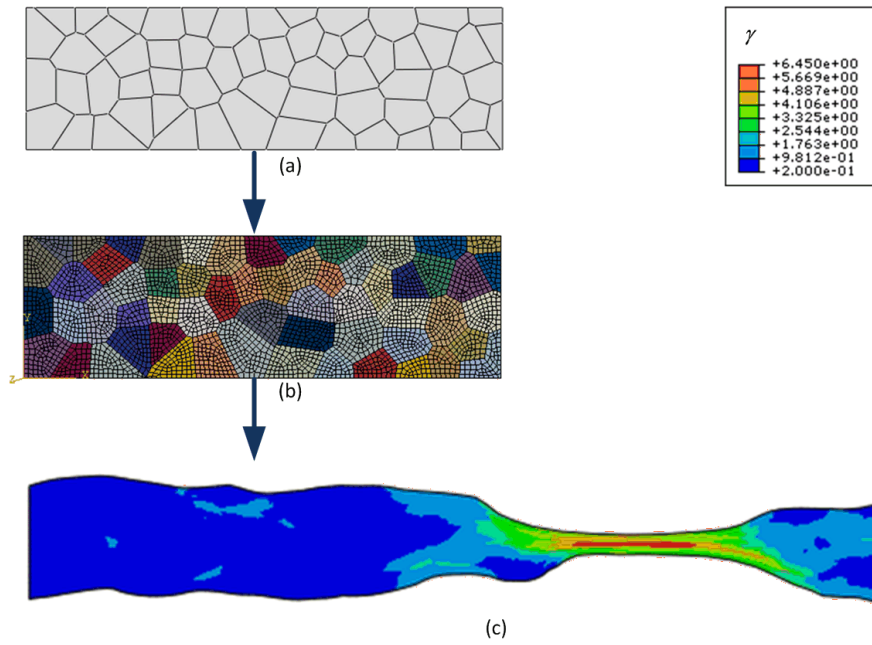


Figure 4.11. CPFE simulations for a regular grain structure.

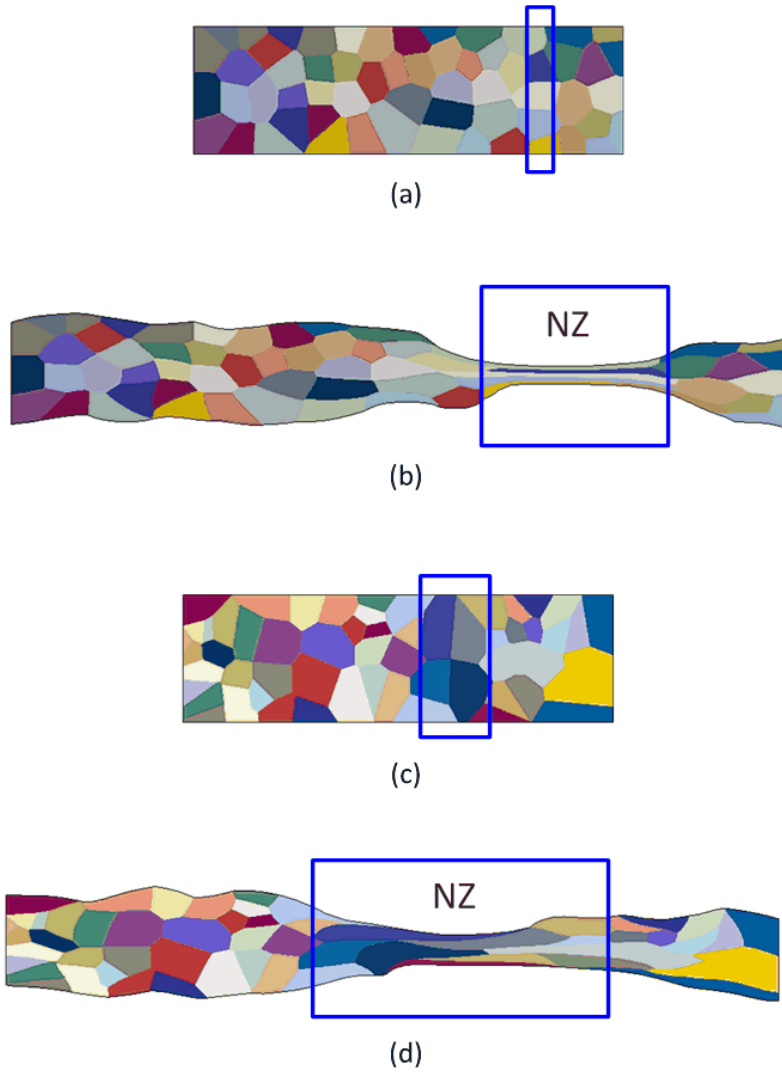


Figure 4.12. Comparison of local deformation for both grain structures.

4.5 Summary

The controlled Poisson Voronoi tessellation model, $\text{CPVT}(\delta|\Omega, N_{\text{seed}})$, has been applied to generate planar virtual grain structure. 2D grain structure uniformity is described by a non-local quantity, i.e. the regularity parameter α , which is used to derive the control parameter δ in the 2D-CPVT model. An improved descriptive model has been established for the mapping from c to α . In conjunction with the solution of c from a set of physical parameters, as given in Chapter 3, the control parameter δ can be determined for a 2D-CPVT model to generate virtual structure with grain area distribution control.

To investigate the performance of the CPVT model, two groups of statistical tests have been conducted. The first group of tests concentrates on the fitness of the regularity control. Results show that for VTs with medium and large numbers of grains, the improved descriptive model performs extremely well. In contrast, for VTs with a small number of grains, random noise arises in irregular grain structure cases, whilst for structures with a regularity value larger than 0.3, the regularity control mechanism performs well regardless of the number of grains in the tessellation. The second group of tests provides a comprehensive comparison of virtual and ideal grain size distributions given as user input. Comparisons have been carried out with consideration of scale effects, i.e., random noise occurred in structures with few grains. The statistical results show the robustness of the CPVT model for generating expected grain structures in terms of grain size distribution. It should be mentioned that the VTs with $\alpha = 0.3$ and $\alpha = 0.5$ were slightly more irregular than the user input. This might be due to the characteristics of the mapping from the distribution parameter c to the regularity α , in which a tiny fitting error in c may result in a large disturbance in α .

Another case study was focused on engineering applications of the 2D-CPVT model and the VGRAIN system. Two images of a commercial aluminium alloy, which present different regularity and grain organisations, have been employed to derive the physical

parameters for configuration of the 2D-CPVT. The overall process has been presented, including image processing for the physical parameters, generation of virtual grain structures and formation of corresponding FE models, which can be used for CPFE analysis. It has been shown that the 2D-CPVT model can now be fully operated based upon real microstructure data *via* the ability to input physical parameters. The major benefit of this model is the flexibility to repeatedly reconstruct virtual grain structures at any scale with proper regularity and grain size distribution characteristics. Therefore, the procedure to reconstruct virtual grain structures from metallographic measurements has been seamlessly integrated.

Two plane strain CPFE analyses have been performed to demonstrate applications of the 2D-CPVT model for planar grain structure generations. By assigning different physical parameters two virtual grain structures with different regularities (grain area distribution properties) were obtained. Simulation results showed the influence of grain regularity on the mechanical behaviour, and strain localisation on the necking behaviour.

Chapter 5

2D Cohesive zone model

A novel scheme is presented for incorporating cohesive interfaces in virtual grain structures for crystal plasticity finite element (CPFE) analyses of inter-granular crack initiation and propagation. The 2D-CPVT model has been used to define the virtual grain structure. The main objective was to provide a novel cohesive zone representation and meshing scheme for CPFE simulations. This has been achieved by the development of an efficient geometrical algorithm designed to generate non-zero thickness cohesive zones (CZs) representing the grain boundaries and multiple junctions. In order to automatically partition multiple junctions, two feasible junction-partitioning schemes are presented, the second of which has the advantage of partitioning junctions using uniform quadrilateral elements and naturally defining their normal and tangential directions. For the second scheme, a rule-based method is presented that carries out the preliminary meshing of CZ junctions, including data representation, edge event processing, cut and trim operations. Furthermore, a grain structure modelling system is presented that was used to generate grain structures, implement the proposed cohesive zone formation and junction partition methods and generate models for CPFE studies. To demonstrate the proposed junction-partitioning and CZ representation schemes, two finite strain CPFE simulations are presented for plane strain uniaxial tension and three-point bending, demonstrating large scale crack initiation and propagation under shear and opening modes.

5.1 Motivation

Micro-forming processes are required to manufacture micro-components for miniature devices and products. In micro-forming, deformation dominated by localised strain in a relatively small number of grains may occur when the number of grains in the smallest part dimension is sufficiently small, and a sufficient number of grains in a given

cross section are capable of large crystallographic slip (i.e. crystal orientations with a large Schmid factor), which is statistically more likely for parts with a high aspect ratio, for example, micro-films, micro-pins and micro-tubes [162]. In these cases, significant thinning and/or localised damage may lead to rapid, random failure in the part depending upon the underlying microstructure. Strain localisation can be captured naturally by crystal plasticity finite element analyses of the forming process [18]. However, capturing large-scale inter- and intra-granular fracture in crystal plasticity finite element (CPFE) analyses requires the addition to the standard CPFE formulation of a method capable of simulating both crack nucleation and propagation with no prior prescription of the crack path. A multitude of methods exist for predicting nucleation, for example, by coalescence of micro-cracks [163], continuum models of ductile damage based on void growth and coalescence [164-166] and other continuum damage criteria [167]. The subject of predicting crack propagation has been extensively studied for over half a century, including classic analytical field solutions for brittle and ductile materials (e.g. power-law hardening materials [168, 169], perfectly-plastic FCC and BCC single crystals [170]), energy based methods [171], critical equivalent plastic strain [172] and mixed critical stress and energy methods, such as the cohesive zone [173, 174]. Traditional linear elastic fracture mechanics (LEFM) methods can be used for effectively predicting crack propagation [175] provided the initial crack and crack path are prescribed, that is, they cannot directly handle the problem of nucleation. Furthermore, there are numerical difficulties in simulating more than one crack propagating simultaneously [176].

Another technique for simulating crack evolution with the potential for simulating decohesion and sliding between grains deforming according to crystal plasticity theory is the extended finite element method (XFEM). This method has been utilised to simulate crack propagation without the need for remeshing [177], including crack face separation by a traction-separation law [178], although requires further development in order to simulate the complicated cracking scenario considered here. The cohesive zone itself stands out as uniquely well suited to the present objective as it is capable of capturing both nucleation and propagation of multiple cracks without prior specification of the

crack path. However, this is only the case provided cracks are restricted to occur at material interfaces, and that its implementation is independent of the grain material model, which in this case is crystal plasticity. Cohesive zone models (CZMs) provide an effective and versatile means for simulating the fracture process [179-183] and have been increasingly employed in finite element simulations of delamination, debonding and general crack initiation and propagation [184-186], hence are appropriate for implementation within CPFE simulations.

In micro-forming, the grain size is often comparable to the smallest dimension of the part, hence the assumption of material homogeneity is inappropriate. In heterogeneous polycrystalline materials, crack initiation and propagation are not dominated by the mean values of stress and strain, but by local maximal values around grain boundaries or interfaces [19], hence properties of the microstructure including size, shape, arrangement and orientation of grains should be considered in understanding the evolution of deformation and local damage. In crystal plasticity finite element (CPFE) analysis, microstructural properties are included in the FE model by explicit representation of grains and their orientations, with plasticity occurring by a continuum representation of crystallographic slip. Traditionally, the concept of Voronoi tessellation (VT) has been extensively used in modelling polycrystalline grain structures for metallurgical applications [22, 23, 111, 153]. The formulation of a VT is naturally analogous to a grain growth process and correspondingly the final structure accounts for the natural variation of grains in terms of both morphology and organisation.

In order to perform micromechanics analyses of inter-granular crack nucleation and propagation for micro-forming applications, there are essentially three geometrical features that need to be represented in a CPFE model: a realistic grain structure, a representation of the grain boundary network (in this case by a group of embedded cohesive zone elements) and a feasible scheme for partitioning grain boundary junctions. Therefore, in employing a VT for the grain structure representation, the original single-line grain boundaries must be replaced by cohesive layers having a specified thickness. Figure 5.1 (a) shows an example of a CPFE model with a grain structure described by a VT with cohesive layer boundaries (henceforth referred to as a VTclb) generated by the

scheme proposed in this chapter. Note that, whilst VTs provide a natural representation of grain structures, they also introduce significant complexity in grain boundary connectivity. Therefore, a primary difficulty in developing a generic algorithm for automatically generating cohesive layers and partitioning the cohesive layer junctions is coping with the event of degeneration of small edges, which is one of the major aims of the present study.

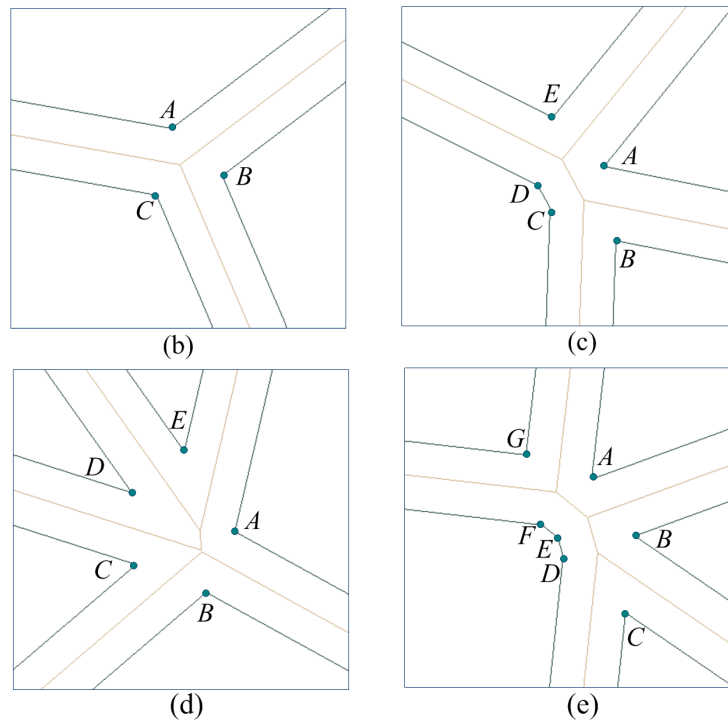
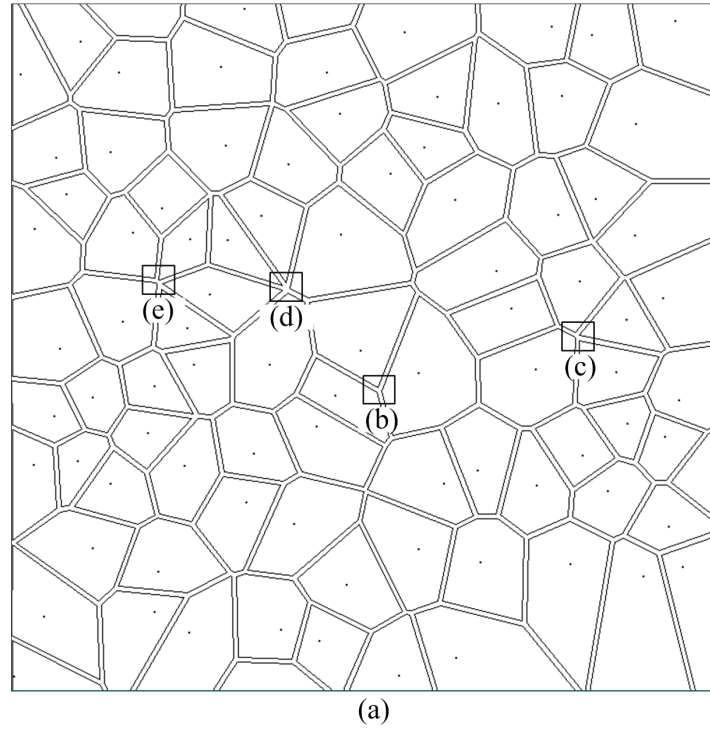


Figure 5.1. An example of the VTclb scheme and specific cases of cohesive junctions.

Furthermore, to build a FE model in commercial FE/CAE software, e.g. ABAQUS, which is typically required for industrial implementations, the following restrictions apply:

- 1) Each interface zone between two grains should be discretised with a single layer.
- 2) Cohesive zones including the layers themselves and their multiple junctions (e.g. triple points) can only be meshed by means of quadrilateral elements, since this shape naturally identifies the normal and tangential directions.
- 3) At least one of either the top or bottom edges of the cohesive element must coincide with a grain element.

Obviously, automatically meshing a cohesive layer boundary is straightforward. However, at a junction of multiple cohesive layers, as shown in Figure 5.1 (b)-(e), further procedures must be introduced to partition the CZ junction into quadrilateral elements. In addition to triple junctions, as shown in Figure 5.1 (b), there are a number of special junction types. For example, Figure 5.1 (d) shows an occurrence of a fivefold junction *ABCDE*. Figure 5.1 (c) depicts a linked triple junction, where one edge of a cohesive layer degenerates to a point *A* upon offsetting the boundaries, hence two neighbouring triple junctions, i.e., *EDA* and *CBA*, are linked to the point *A*. Similarly, the junction in Figure 5.1 (e) presents a double linked triple junction. In FE simulations, it is a common fact that a more uniform mesh generally gives more accurate results, but the complicated junctions that may occur in a VTclb make it difficult to automatically generate uniform junction partitions.

5.2 Voronoi tessellation with cohesive layer boundaries

In this section, an offset method is presented that transforms ordinary Voronoi tessellations into non-zero thickness boundaries. The transformed VTs can be used to represent virtual grain structures and model cohesive zones for studying grain boundary sliding, inter-granular cracking nucleation and propagation, etc.

5.2.1 VT and VTclb

In order to incorporate a CZ model to formulate inter-granular traction-separation relations, cohesive interface elements need to be embedded along grain boundaries. As stated above, the grain boundaries in a VT consist of a network of lines. Naturally, in the presence of a VT, cohesive elements can be directly produced by replacing the original lines with a network of cohesive layers. Figure 5.2 (b) illustrates a VTclb, where grains are reconstructed by inward offsetting of the original grain boundaries, shown in Figure 5.2 (a), by a specified distance. Although the thickness of cohesive layers is generally thin, whatever thickness is specified, there is a chance that small grain edges can disappear as a result of the offsetting procedure, e.g., after offsetting the boundary AB of the two adjacent grains in Figure 5.2 (a), the two offset upper and lower boundaries degenerate to vertices C and D for the two grains, respectively, as shown in Figure 5.2 (b).

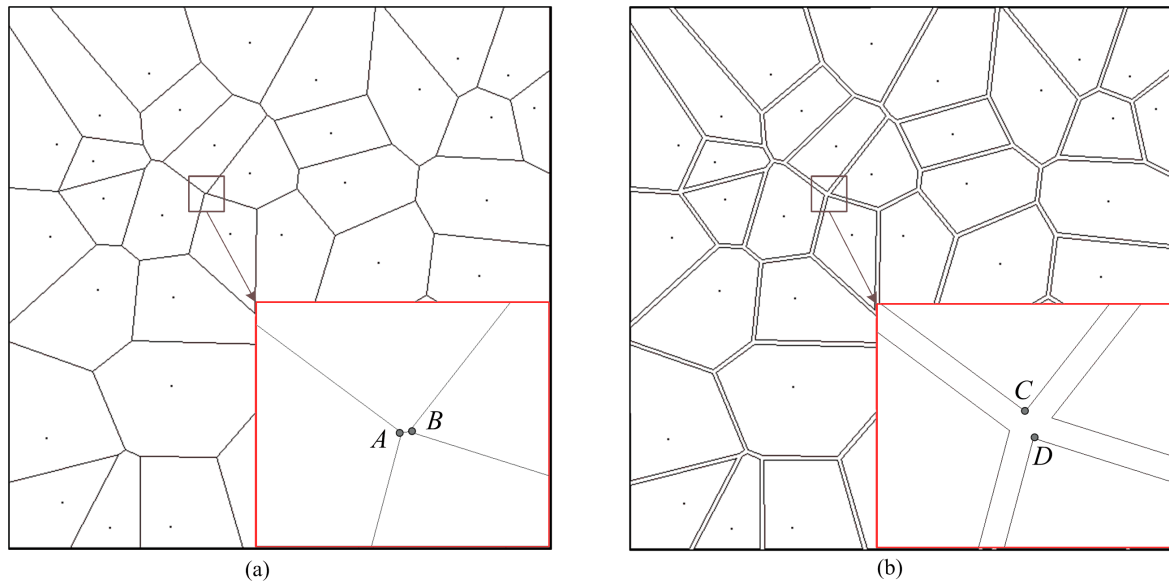


Figure 5.2. Illustration of offsetting grain boundaries to represent cohesive layers. (a) A grain structure based on the VT representation; (b) a VTclb, with a degenerate grain shown with one of the original boundaries missing after the offsetting procedure.

5.2.2 Grain boundary offsetting

Generation of cohesive layers for a VT can be achieved through the processing of individual grains. An intuitive method of inward offsetting of a convex polygon can be described by: 1) Creating offsets for all decomposed edges followed by reconnecting the resulting pieces according to the original topological relations, i.e., the head-to-tail sequence of consecutive edges; 2) tracking and removing boundary inversions [187]. However, tracking boundary inversions in this context requires complex heuristics to patch various special cases; this sacrifices efficiency for the sake of robustness.

Instead, in this work, a structured offset method is employed to reliably produce inward-offset polygons, hence a complete VTclb. In this context, the term “structured” means that for each single edge, the inward offset is restricted to a decomposed sub-polygon, which is divided by the polygon’s medial axis. The medial axis of a simple polygon is a tree-like planar structure consisting of a set of line segments, as illustrated by the dot-lines in Figure 5.3 (a). Any point on one of the line segments has more than one closest point lying on the boundary of the polygon, e.g., the point L on the medial axis in Figure 5.3 (a), to which the points M and N on the polygon are equidistant, and the point G , to which the points P , Q , R and S are equidistant. Therefore, it follows that each line segment of the medial axis bisects the angle formed by the two sides that it intersects or, in the case of line segments that do not intersect the boundary of the polygon, the angle between the sides nearest to it. For example, in Figure 5.3 (a), the medial axis CG bisects the angle $\angle BCD$ and the medial axis GH bisects the angle $\angle AOD$. The algorithm to build the medial axis has been detailed in [188], which takes $O(n)$ time complexity where n is the edge number of the polygon.

Furthermore, the medial axis of a simple polygon with n edges divides the polygon into n sub-regions, as shown in Figure 5.3 (b), such that any point in the i^{th} region is closest to the i^{th} edge of the polygon among the edges of the given polygon. Thus, for each edge e_i of the polygon, there corresponds a decomposed sub-region r_i , which contains the offset edge e_i' of the edge e_i . If the offsetting distance is large enough, the offset line may be external to the given sub-region, hence be degenerated into a point, as

in sub-region 4 in Figure 5.3 (c). Determination of whether the offset edge e_i' is degenerated can be directly obtained by comparing the offset distance with the distance between the far endpoint and the edge e_i . Once all the offset edges are obtained, the inward-offset grain can be generated by linking these edges, as shown in Figure 5.3 (d). The algorithm to generate the cohesive zone layers for a given VT is presented in Figure 5.4.

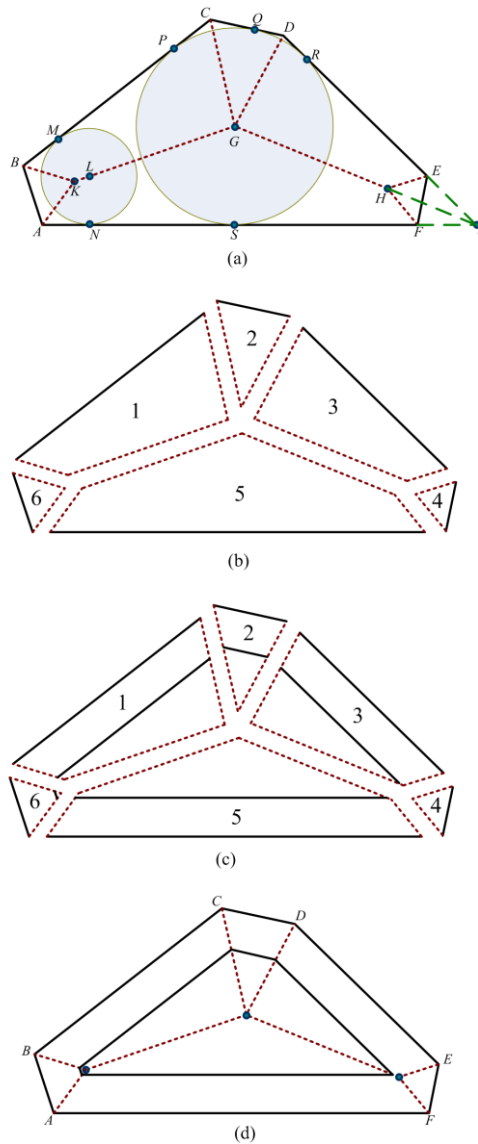


Figure 5.3. Illustration of the structured offset method. (a) Generating the medial axis; (b) decomposing the grain into a set of sub-polygons; (c) structured offset boundaries; (d) formation of an inward-offset grain.

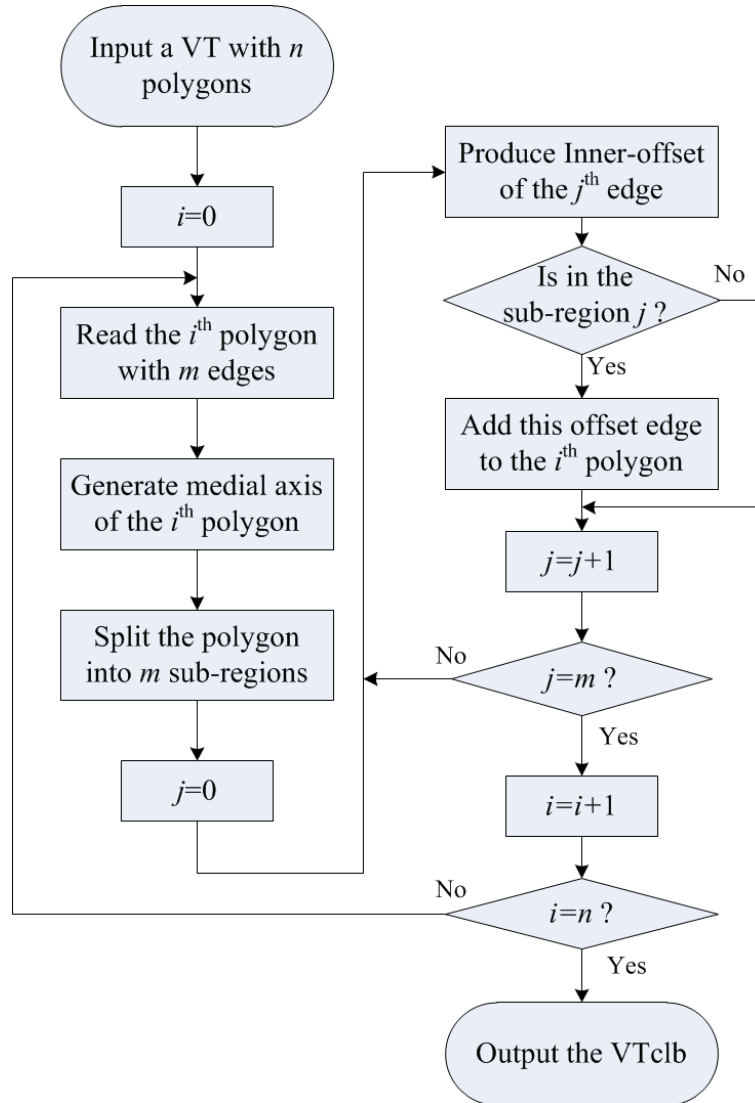


Figure 5.4. The algorithm to generate the VTclb structure for a given VT.

5.3 Cohesive junction partition

Utilisation of finite element methods to solve boundary value problems requires explicitly discretising the cohesive zones including grain boundaries and cohesive junctions into mesh elements. As mentioned before, there must be a single layer of cohesive elements between two adjacent grains, and all the mesh elements must be quadrilateral. Conversely, due to the geometrical complexity of grains in a VT, there are a

range of complex junctions, e.g., triple points, quadruple junctions, fivefold junctions, and other types of linked junctions. In addition, to ensure simulation accuracy, there is a general requirement for the mesh to be as uniform as possible. Therefore, formulation of a consistently uniform junction partition scheme and developing efficient algorithms for the partitioning are critical for crystal plasticity finite element simulations.

5.3.1 Schemes

In a real grain structure, grains are generally organised irregularly, hence the junction shapes are highly variable. The most common is the triple junction, where three boundary layers meet. Despite the variety of junction shapes, a general pattern scheme is required to mesh junctions automatically using quadrilateral elements. Here, a triple point junction is considered to present possible junction partition patterns.

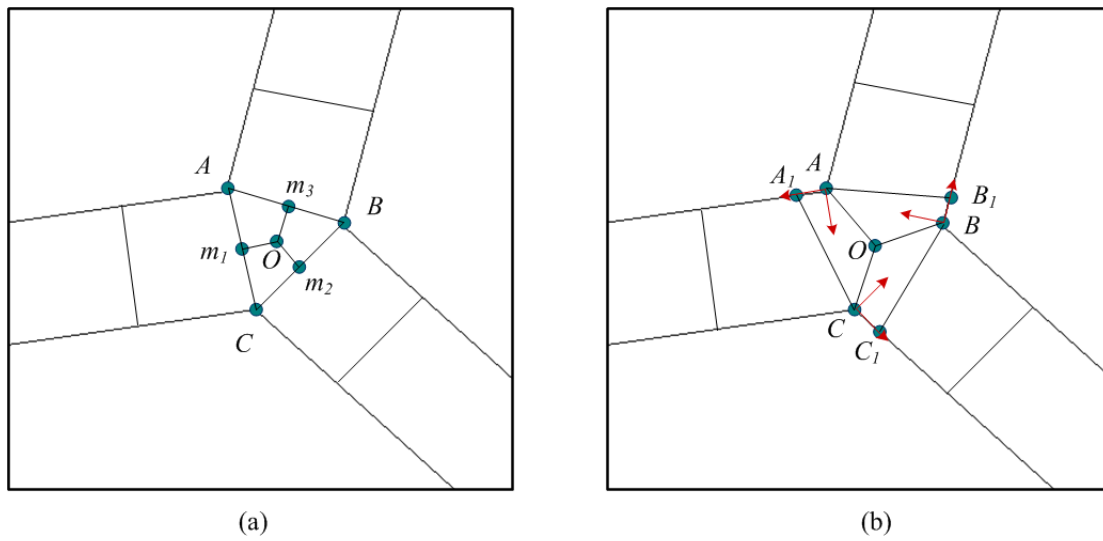


Figure 5.5. Illustration of two junction partition patterns for meshing cohesive zones.

Figure 5.5 (a) shows a simple way to subdivide the junction. In this pattern, corner vertices including A , B , and C , are directly linked to define the junction domain. In this way, the junction and boundary layers of the grains are separated into different domains. Since this junction is a triangle area, it should be further subdivided into quadrilateral

elements. This can be achieved by constructing three perpendicular line segments to the triangle's sides from the centre of the triangle. However this partition pattern involves free nodes, i.e., the intersection nodes m_1 , m_2 and m_3 . Although this can be remedied by introducing a linear multi-point constraint (MPC) to tie free nodes to the midpoints of the corresponding cohesive elements sides, this scheme does not provide any rationale for the choices of normal and tangential directions for the junction elements and adds computational complexity.

Figure 5.5 (b) provides an alternative scheme not requiring the introduction of unnecessary free nodes. In this pattern, the junction region is initially cut by the line segments linking the centre and corner vertices, i.e., AO , BO and CO . In this way, the cohesive zone junction is divided into three independent cohesive layers without requiring an extra central junction as in the first pattern. To minimise the effect of the irregular layer end shape, particular procedures are required to mesh the ends of the cohesive layers nearest the junction. A duplicate node is introduced at A , then moved to A_1 , etc., hence forming the quadrilaterals $AOBB_1$, $BOCC_1$, $COAA_1$. It can be seen that this scheme results in more uniform junction elements, and also the assignment of normal and tangential directions becomes natural and convenient, as illustrated by the arrows in Figure 5.5 (b). Although the second pattern is a clear improvement, junctions are not always as simple as the triple junction. The rest of this section describes a geometrical method for implementing this junction partition scheme for any junction type.

5.3.2 Algorithm

Whilst the second partition scheme has the merits of avoiding the introduction of free nodes and capability of providing more regular mesh elements, especially around the junction region, this scheme introduces significant difficulties in automatically dividing cohesive junctions in a real VT and meshing the cohesive layer ends. There are two procedures required: 1) Divide the junctions and segment the entire cohesive zone into a set of n individual cohesive layers for an n -fold junction, and 2) mesh the cohesive layer ends.

5.3.2.1 Initial partition

Here, a cohesive layer unit comprises an elementary rectangular cohesive zone, with upper and lower boundaries of the two adjacent grains, and two ends, each defined by its respective junction partition, the grain boundaries and the boundary of the elementary cohesive zone. As illustrated in Figure 5.6 the region of $OAA'O'B'B$ forms a cohesive layer unit. In this unit, the edges of AA' and BB' are the interfaces between the cohesive layer unit and the adjacent grains. The two ends AOB and $A'O'B'$ are partitioned from the two corresponding cohesive junctions based on the second scheme. In order to prepare for further meshing operations, the cohesive layer unit is represented by a data structure consisting of the six points of the three line segments, including the two cohesive layer interface boundaries AA' and BB' , and a central line segment OO' defined as the edge of the original VT.

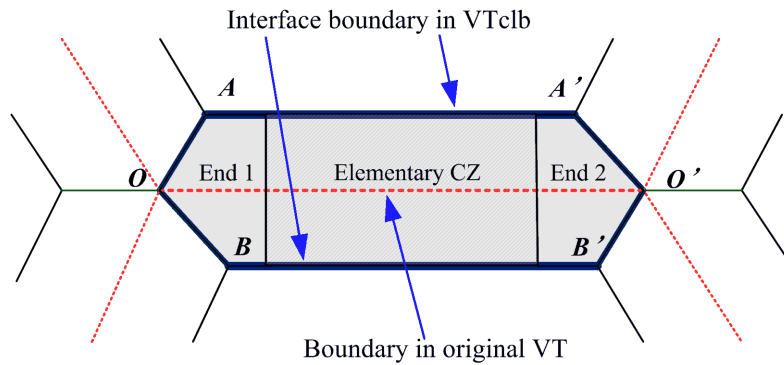
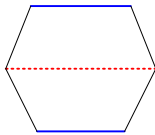
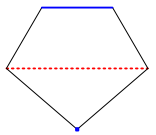
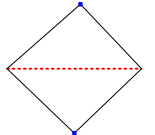
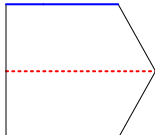
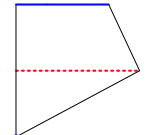
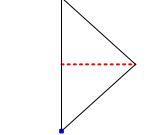


Figure 5.6. Illustration of a cohesive layer unit in a VTclb.

There are six categories of cohesive layer units, as shown in Table 5.1. Type-I and type-II are classified based on the shape of a layer's ends. In type-I, the two ends are defined by the junction partitions, whereas type-II intersects the VT's domain border, hence loses one of its two ends. Each type of layer unit has three possible configurations: the ideal case, the case of one edge degenerated to a point and the case of two degenerated edges. Henceforth, the six categories of layer units are labelled as "type-case", for example, II-2 indicates a type-II cohesive layer unit with one degenerate edge.

Table 5.1. Summary of the six types of elementary cohesive layer unit. (Thick lines represent the boundaries of a VTclb, dot-lines are the original grain boundaries in the corresponding VT, and normal lines are junction partitions.)

	1. Ideal	2. One edge degenerated	3. Two edges degenerated
Type-I (Inner layer)			
Type-II (Border layer)			

5.3.2.2 Adjustment

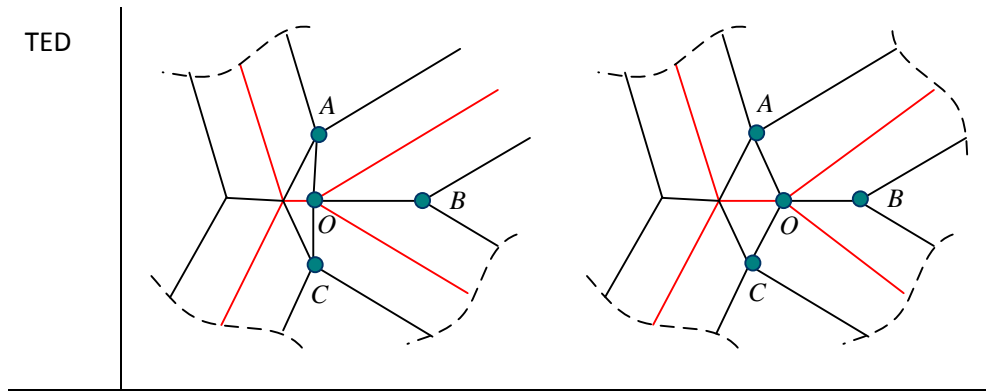
In a FE model, cohesive layer units are prepared for meshing by quadrilateral elements. Since meshing starts from the ends of a cohesive layer unit, the two edges forming a layer's end are taken as part of the first quadrilateral element. It is important to note that, in this situation, the centre of the junction is assumed to be one node of an element, e.g., the points O and O' shown in Figure 5.6. However, in a real VTclb structure, there are a few complicated special cases that require further consideration.

There are four possible elementary special cases listed in Table 5.2 (others may occur as combinations of the elementary cases) resulting from partitioning junctions into cohesive layer units. The first is referred to as the *corner case*, in which the cohesive layer is split by a domain corner and the central line does not intersect the corner. In this case, the end is still formed by two edges, but, in contrast to the normal layer unit, in which the end point is the centre of a junction, the end point is the VT's domain corner.

Therefore, for the purpose of uniformly deriving the data structure, the end point of the central line in this case is moved to the corner. The *border case* is that of a cohesive junction cut by a border, where the related central lines cannot meet in the junction centre. Note that, although there is no centre point to be used to form the mesh element nodes, there are still two edges existing for each layer unit. Therefore, the central lines can be moved to the junction intersection, and the intersection point can then be used as the centre of cohesive junction.

Table 5.2. Special case ends of a cohesive layer unit.

	Initial geometrical data	Transformed geometrical data
Corner		
Border		
TEDC		



The third case usually occurs at the VT's domain border, when the border cuts a cohesive layer unit such that only some part of the unit end remains. This case is referred to as the *two-edges-degenerated cut case* (TEDC), that is, this type of layer unit is regarded as a cohesive layer unit that has its junction centre moved to the domain boundary, as shown in Table 5.2. This case is the same as the II-3 type of cohesive layer unit shown in Table 5.1. The original triangle is relatively tiny, hence can be removed and the centre of the junction then moved to the other end of the central line segment. The final case, TED or *two-edges-degenerated*, is a case of cohesive layer unit of type I-3, as shown in Table 5.1. The I-3 type can be represented by a simple quadrilateral element. However, the shape of the TED case might be non-convex, as shown in Table 5.2. Therefore, an additional procedure is required for generality that adjusts its shape by means of repositioning the two junction centres. If an end of a cohesive layer unit is found to be non-convex, or some inner angle is larger than a user-specified tolerance, the related junction centres, for example, O is moved to O' , which is the centroid of triangle ABC . After processing these special cases, all the layer units can be passed to the next meshing procedure.

5.3.2.3 Preliminary mesh

The cohesive layer unit types, listed in Table 5.1 dictate the algorithms used to mesh the cohesive zone. Note that layer unit II-3 is processed as the TEDC case, hence is removed from the cohesive zone geometry. In addition, layer units I-3 and II-2 are quadrilateral elements, which can be represented directly by mesh elements. Layer unit

I-2 has one interface edge degenerated to a point. In this case, the interface layer consists of a point and one edge, so this unit can be efficiently split into two quadrilateral elements by linking the degenerated point to a selected point in the other interface edge, which may be chosen to be the middle point of that line segment, as the point P shown in Figure 5.7 (a). As a result, there are only two types of cohesive layer unit that need to be considered in meshing.

The mechanism to mesh a cohesive layer unit can be described as a two-step process, including “cut” and “trim” operations. The cut operation aims to turn a two-edge end into a one-edge end by cutting the end corner, while the trim operation is applied to a one-edge end, by cutting a quadrilateral element from the layer unit. The result from the trim operation is a regular layer with rectangular ends and a trimmed mesh element. The rules of preliminary meshing are demonstrated in Figure 5.7 (b)-(d) and implementation details are explained by the following:

- 1) Initially, a minimum element width L is specified by the user, which determines the minimum edge length of mesh elements in a cohesive layer.
- 2) Given a two-edge end of a cohesive layer unit, e.g., AOB in (b) and (c) of Figure 5.7, if $|AO| \geq |BO|$, the end is cut, starting from A along the line perpendicular to the opposite edge. The intersection point is denoted as P . If $|BP| \geq L$, the preliminary mesh of this end is finished, as shown in Figure 5.7 (b). Otherwise, the point P is moved to Q , where $|BQ| = L$, as shown in Figure 5.7 (c). For the latter case, first the cut operation is finished, although, a further trim operation is required. The trimming operation starts from the point N , where $|QN| = L$, then the cohesive layer is trimmed along the line perpendicular to the opposite edge, i.e., NM .
- 3) Given a one-edge end of a cohesive layer unit, the starting point for the trim operation is determined by the angle of the end, as illustrated in Figure 5.7 (d). The criterion is that if $\angle OBB' \geq \angle OAA'$, the starting point P is chosen along the boundary edge BB' .

Two examples demonstrating the implementation of the aforementioned meshing operations are shown in Figure 5.8 for the general cases of a two-edge end and a one-edge end. For the right end of the layer unit in Figure 5.8 (a), since $|B'O'| \geq |A'O'|$, the opposite edge from B' is cut by a perpendicular line. Since $|A'P'| \geq L$, the point P' is kept as the 4th node for the first mesh element. Note that no further trim is required in this case. For the left end in Figure 5.8 (a), $|AO| \geq |BO|$, hence the opposite edge from A is cut by a perpendicular line. Since $|BP| \leq L$, P is moved to Q with $|BQ| = L$. Thus the mesh element $AQBQ$ is obtained from the cut operation. Furthermore, the point N is selected such that $|QN| = L$. From N , this end is trimmed by a perpendicular line, by which a mesh element $MAQN$ is formed. The left end in Figure 5.8 (b) is a type II-1 layer unit. The line AB is part of a border of the VT's domain. Since this end is one edge, only a trim operation is needed. The point P is selected from the edge BB' , since $\angle OBB' \geq \angle OAA'$ and $|BP| = L$. The opposite edge is trimmed from point P by a perpendicular line, by which a mesh element $ABPM$ is obtained.

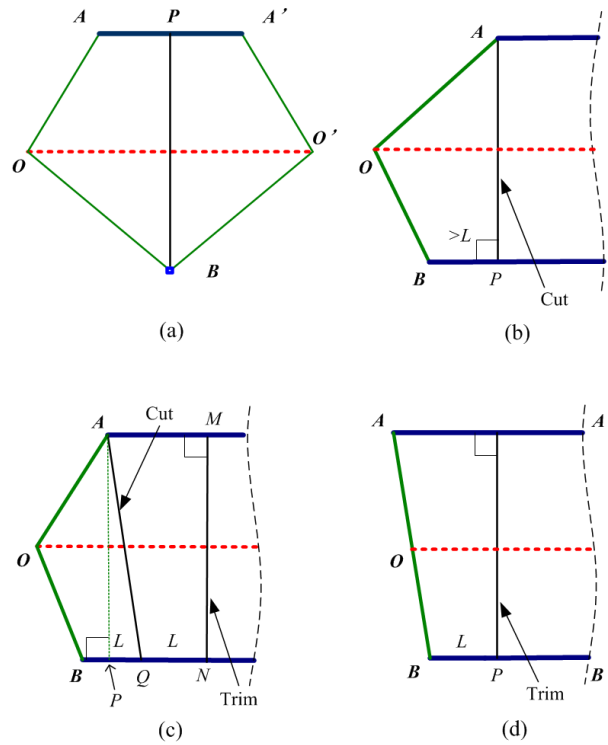


Figure 5.7. Illustration of meshing rules.

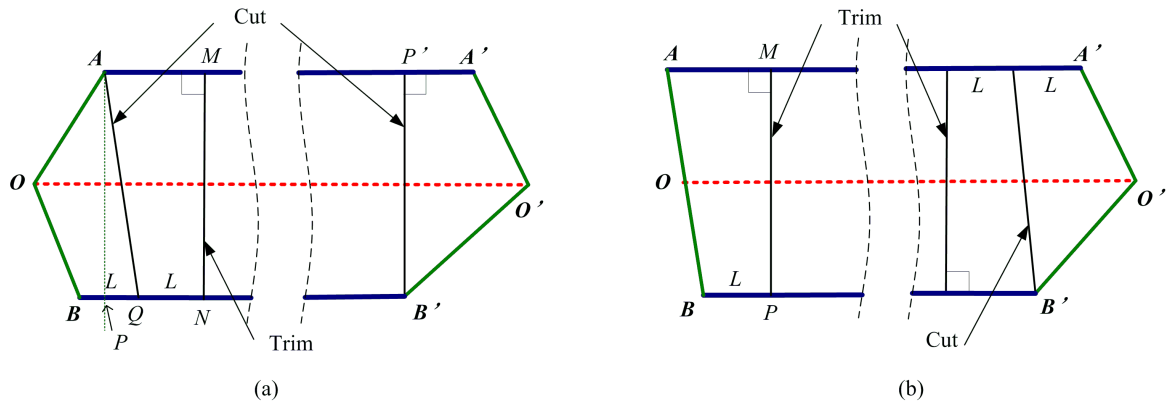


Figure 5.8. Example cases of meshing operations.

In summary, the meshing process is performed in the order of cutting, trimming and further meshing the rest of the elementary rectangular CZ. It is worth noting that if a boundary edge is less than a certain length, e.g., $6L$, only cutting operations are required, as shown in Figure 5.9 (a) and (b). Moreover, the selection of the minimum element width L should be comparable to the thickness of the cohesive layers. The relation that L is half the thickness is used in this study.

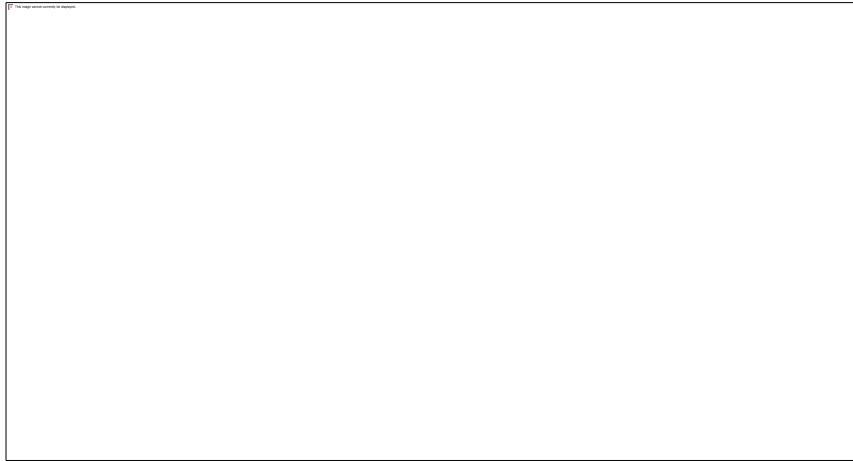


Figure 5.9. Example cases of meshing CZ layer units with short interface boundaries.

5.4 CPFE applications

In [140] and [189], the controlled Poisson Voronoi tessellation (CPVT) model has been developed to generate virtual grain structures for crystal plasticity finite element simulations. Voronoi tessellations are taken as the geometrical representation of polycrystalline materials grain structures. The CPVT model incorporates four physical parameters: the mean grain size, a small grain size, a large grain size and the percentage of grains within that range. These physical parameters are intuitive input to describe the grain morphology, which essentially determines grain distribution features including the regularity. Virtual grain structures obtained from the CPVT model are statistically equivalent to metallographic observations of polycrystalline materials in terms of the tessellation's regularity [154] and grain size distribution [140].

5.4.1 System for Grain structure generation

In this work, the proposed cohesive zone representation technique and the preliminary meshing scheme, as illustrated by the shaded modules in Figure 5.10, are incorporated into the 2D-CPVT model for simulations of grain boundary sliding and separation. The major implementation steps to generate a virtual grain structure with cohesive zones at grain boundaries are explained as follows.

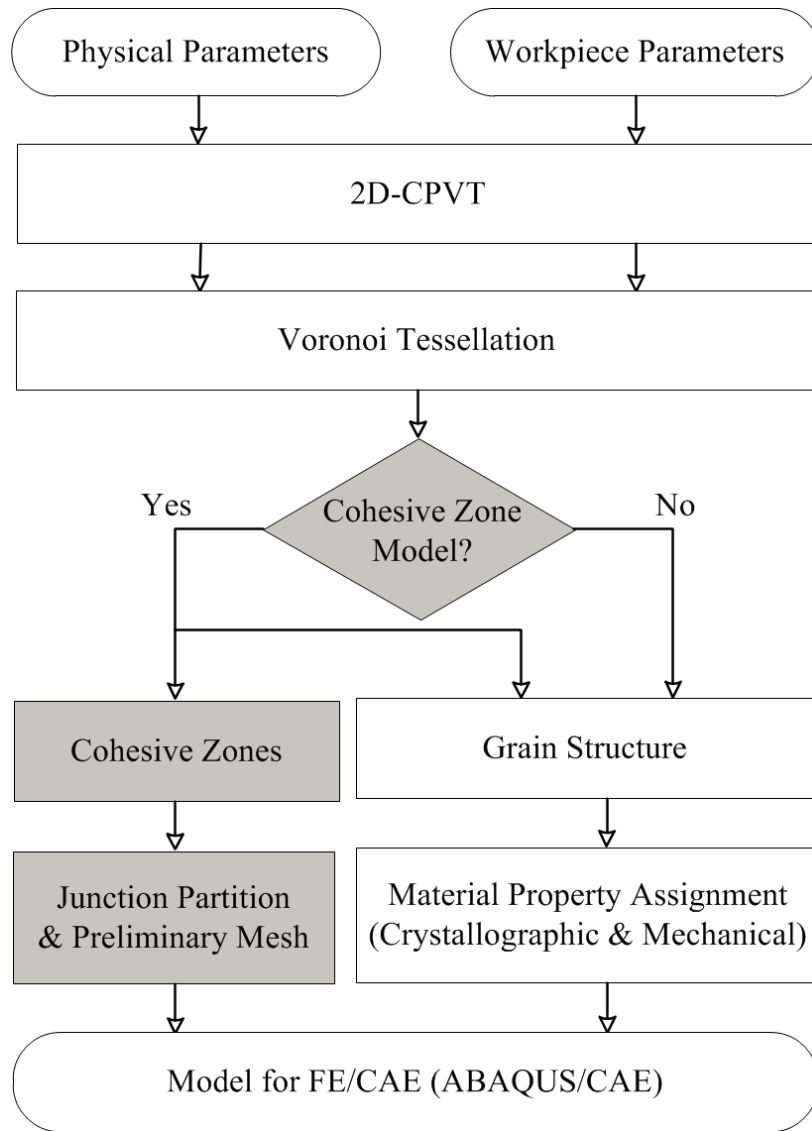


Figure 5.10. Illustration of a grain structure generation and CPFEM model definition process.

Two groups of user input, including the physical parameters, which determine the implementation parameter for the 2D-CPVT model, and workpiece parameters, which define a virtual structure's domain, are specified. It is worth mentioning that by assigning the physical parameters, including the mean grain size D_{mean} , a small grain size D_L , a large grain size D_R and the percentage P_r of grains in that range, repeatedly generated VTs are statistically equivalent in terms of regularity and grain size distribution. If the generated virtual grain structure is required to include a cohesive zone model for grain

boundaries representation, the grain structure is further processed using the proposed VTclb method. Next, material properties including mechanical properties and grain orientations are assigned to the grains in the resultant grain structure, and for the cohesive zones, junctions are preliminarily partitioned. The final virtual grain structure with material properties and cohesive junction partitions can be directly exported to commercial FE/CAE platforms, e.g., ABAQUS, by means of corresponding text scripts such as the python script for ABAQUS. Final meshing is then carried out based on the VTclb partitions by the commercial solver.

5.4.2 Case studies

To demonstrate the proposed VTclb method and the novel cohesive junction partition scheme in modelling cohesive zone grain boundaries in polycrystalline materials, two crystal plasticity finite element analyses, including a 2D plane strain analysis of uniaxial tension and a 2D three-point bending test, have been performed. The two virtual grain structures, including the grains, the cohesive zones and junction partitions, were generated using the VGRAIN system. Schematic diagrams of the simulations are presented in Figure 5.11 and Figure 5.15. In both virtual specimens, grain boundaries were represented by cohesive elements. The thickness of cohesive layers for both FE models was $0.5\ \mu\text{m}$, which is much smaller than the dimensions of the grains whose fracture was analysed [190] and also compromised the number of mesh elements to be used. The grain grayscale values correspond to the grain orientations, which were randomly generated using the VGRAIN system. Crystal plasticity constitutive equations as in Wang et al. [18] were implemented in the commercial FE code ABAQUS/EXPLICIT through the user defined material subroutine VUMAT. A calibration of the crystal plasticity equations for a free-cutting steel, as described in Karimpour et al. [191], using tensile test data generated over a range of strain rates and temperatures, was used in the simulations.

5.4.2.1 Uniaxial tension

For the uniaxial tension model shown in Figure 5.11, the overall dimensions of the specimen were $300\ \mu\text{m} \times 150\ \mu\text{m}$. The physical parameters input to the VGRAIN system were presented in Table 5.3. A displacement U , corresponding to an engineering strain level of 0.3, was applied on the right edge of the model with the lateral faces free of constraint.

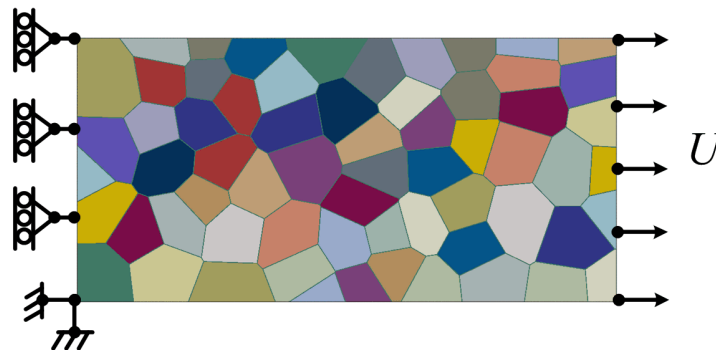


Figure 5.11. Schematic diagrams of the plane strain model for CPFE simulation.

Table 5.3. Physical parameters and corresponding grain structure properties for the two CPFE models.

Label	Physical parameters (μm^2)				Equivalent grain size [†] (μm)			CPVT model parameters			
	D_{mean}	D_L	D_R	P_r	d_{mean}	d_L	d_R	d_{reg} (μm)	α	δ (μm)	N
Tension	700	350	1050	97%	28.4	20.1	34.8	28.5	0.6	17.3	64
Bending	700	350	1050	97%	28.4	20.1	34.8	28.5	0.6	17.3	320

[†] Assuming a hexagonal grain shape.

The damage initiation criterion of the cohesive constitutive law was based on the maximum of the ratios of the cohesive stress components to the corresponding damage initiation stresses reaching unity

$$\max \left\{ \frac{t_n \langle t_n \rangle}{t_n^0}, \frac{|t_t|}{t_t^0} \right\} = 1,$$

where $\langle \cdot \rangle$ is the Heaviside step function and superscript 0 denotes the damage initiation threshold. The normal and tangential cohesive stresses t_n and t_t were calculated by:

$$t_n = \begin{cases} (1 - D)\bar{t}_n, & \bar{t}_n \geq 0 \\ \bar{t}_n, & \bar{t}_n < 0 \end{cases}$$

and

$$t_t = (1 - D)\bar{t}_t,$$

where \bar{t}_n and \bar{t}_t are the linear elastic traction components and D is the damage parameter ranging from 0 to 1. The damage evolution is based on a failure energy criterion, defined by:

$$D = \frac{\delta_m^f(\delta_m^{\max} - \delta_m^0)}{\delta_m^{\max}(\delta_m^f - \delta_m^0)},$$

where $\delta_m = \sqrt{\langle \delta_n \rangle^2 + \delta_t^2}$ is the effective displacement, δ_m^{\max} is the maximum value attained during the loading history and δ_m^0 is the value at damage initiation. (Note that necessarily $\delta_m^{\max} > \delta_m^0$ if the damage initiation criterion is satisfied.) Moreover,

$$\delta_m^f = 2G^C/T_{\text{eff}}^0,$$

where T_{eff}^0 is the effective traction at damage initiation and G^C is the fracture energy.

Two different ratios of the normal to tangential damage initiation stresses, i.e., t_n^0/t_t^0 , were simulated: $t_n^0/t_t^0 = 1$, and $t_n^0/t_t^0 \rightarrow \infty$, i.e. a very high damage initiation stress of the normal direction relative to that of the tangential direction, forcing damage to occur only under shear. For the first case, the damage initiation stresses were $t_n^0 = t_t^0 = 150$;

for the second case, the damage initiation stresses were $t_n^0 = 2000$ and $t_t^0 = 150$. For both cases, the damage evolution energy of CZs was $75 \mu\text{J}/\text{m}$. In Figure 5.13, the results of a demonstration uniaxial tension CPFE simulation with $t_n^0/t_t^0 = 1$, i.e. equal separation and shear behaviour are presented. Figure 5.13 (a) shows the initiation of a crack and contours of damage accumulation in other areas, while Figure 5.13 (b) shows the propagation of the crack. Early-stage strain localisation necking can also be observed. In Figure 5.13 (c), the geometrical evolution of a junction (indicated in Figure 5.13 (a)), where the crack passed through, is shown. The results of the junction partition operation are highlighted by the dot-lines, of which the white dot-lines and black dot-lines are from the *cut* and *trim* operations, respectively. It can be seen that both of the triple junctions were partitioned uniformly by three quadrilateral elements and these elements exhibited smooth geometrical deformation during cracking. The crack path is normal to the direction of applied tension, as expected given the equal weighting of the normal and shear cohesive stresses, and that the maximum shear stress is lower than the maximum normal stress. Figure 5.14 shows the results of a uniaxial tension CPFE simulation with $t_n^0/t_t^0 \rightarrow \infty$, i.e., with only shear failure possible. Crack nucleation and full propagation are presented in Figure 5.14 (a) and (b), respectively. The oblique angle of the crack path (roughly 45° to the axis of applied loading, along grain boundaries where the shear stress is greatest), as shown in Figure 5.14 (b), confirms that failure in the simulation is governed by maximum shear stress, and depicts the limit of the interplay between shear and normal damage modes. Figure 5.14 (c) illustrates the damage evolution of the junctions specified in Figure 5.14 (a). In both cases (Figure 5.13 and Figure 5.14) at each junction, elements which share the centroid node have the same normal and tangential directions as their corresponding cohesive layer units, hence their behaviour is consistent with that of the other elements in their units. In addition, Figure 5.12 presents the strain-stress properties of the two simulations, results show that the grain structure with the CZ property only considering pure shear damage was more ductile than the grain structure with the CZ property of equal normal and tangential damage initiation stresses. This feature can also be observed from the deformed grain structures shown in Figure 5.13 and Figure 5.14.

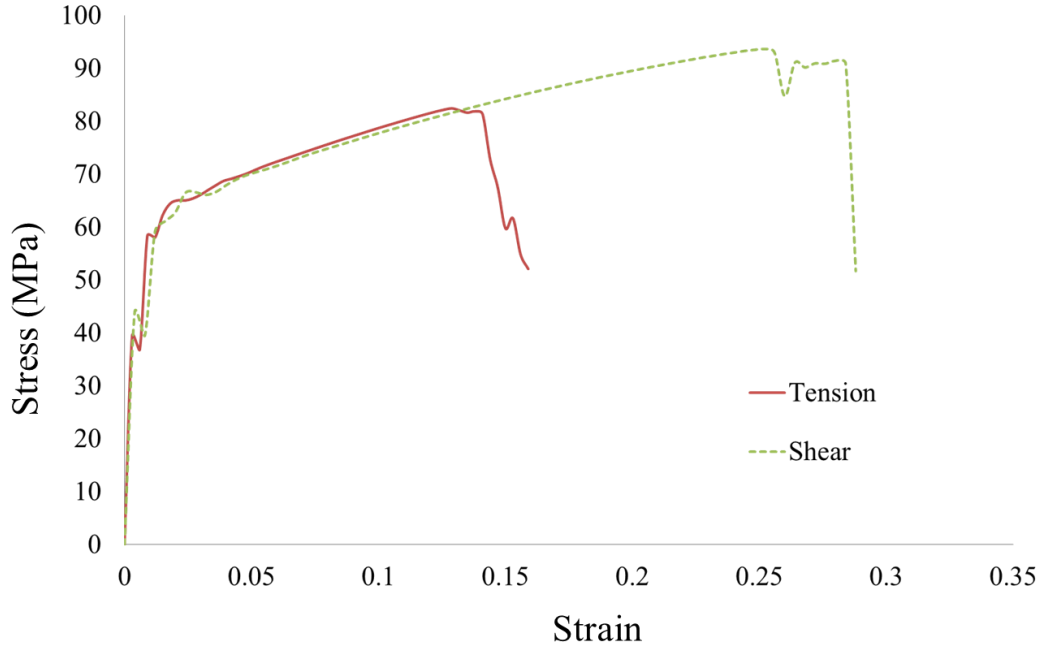


Figure 5.12. The stress-strain curves corresponding to the two different ratios of the normal to tangential damage initiation stresses.

5.4.2.2 Three-point bending

The three-point bending model demonstration shown in Fig. 14 has a specimen domain of $1500\text{ }\mu\text{m} \times 150\text{ }\mu\text{m}$ and physical parameters are given in Table 5.3; the crystal plasticity calibration is also the same. The model configuration was set up in accordance with the metallic materials bend test standard specified in ISO 7438:2005. It can be seen that the proposed VTclb grain structure model and junction-partitioning scheme are capable of simulating the crack initiation at the tensile surface, shown in Fig. 14 (a), and subsequent inter-granular fracture and crack propagation through the entirety of the specimen, shown in Fig. 14 (b). Besides the major crack path shown in Fig. 14 (b), there were other boundaries with considerable accumulated damage during the bending process shown by the contours of damage. Fig. 14 (c) illustrates the geometrical separation behaviour of two of the junctions under this mode of loading in the region specified in Fig. 14 (a).

5.5 Summary

Voronoi tessellations account for the natural variation of grain shapes and sizes. However, this advantage leads to complexity in the grain boundary connectivity. There are usually triple junctions, quadruple junctions, higher order multiple junctions and complex special cases. Therefore, in order to simulate grain boundary sliding and inter-granular crack nucleation and propagation using non-zero thickness CZ models in a general (any number of grains, any regularity) virtual polycrystal, automatically generating cohesive zones and effectively meshing the junctions must be accomplished. The main objectives of this work were the development of an effective CZ representation method for the Voronoi tessellation model, and resolving the challenging problem of automatically partitioning multiple junctions for meshing with quadrilateral elements for any possible scenario.

The method to automatically generate cohesive layer boundaries (VTclb) has been proposed to define the grain structures for polycrystal CZ simulations. Furthermore, a robust geometrical algorithm has been developed to produce the VTclb structure for any given Voronoi tessellation. The algorithm is based on the medial axis structure and proceeds in three major steps: Generating the medial axis for a grain, dividing the grain into sub-regions in accordance to its edges and finding the offset edges, and linking all the offset edges into the inward-offset grain.

Two feasible junction-partitioning schemes were presented. The first scheme requires extra edges and hence additional nodes in order to mesh the junction by quadrilateral elements. In addition to the resultant extra geometrical complexity, the extra nodes are free of constraints and thus MPCs must be provided. In contrast, the second scheme is able to partition a junction by more uniform quadrilateral elements and has the advantage of assigning the partitioned elements with natural normal and tangential directions. Therefore, a rule-based method has been developed to implement the second scheme for general junction partitioning and preliminary meshing of CZ junctions. It consists of dividing the cohesive layer units and partitioning the layer units by cut and trim operations.

The proposed VTclb junction-partitioning and preliminary meshing method is a powerful companion for the 2D-CPVT model. Using this combination, VTclb grain structures can be automatically generated with grain size distribution control. After a grain structure has been generated, the junction partitioning is then performed and the preliminary mesh automatically obtained. Grain orientations can be assigned for CPFE analysis with preferred texture or randomly generated based on a random number generator. The quality of the resultant grain structure and mesh has been verified by CPFE simulations: Two virtual uniaxial tensile tests and a three-point bending test. Crack initiation and propagation have been studied, and the details of junction elements deformation were examined. The results show that the novel junction-partitioning scheme is ideal to automatically generate nonzero thickness CZ boundaries for simulating grain boundary sliding and decohesion in polycrystals.

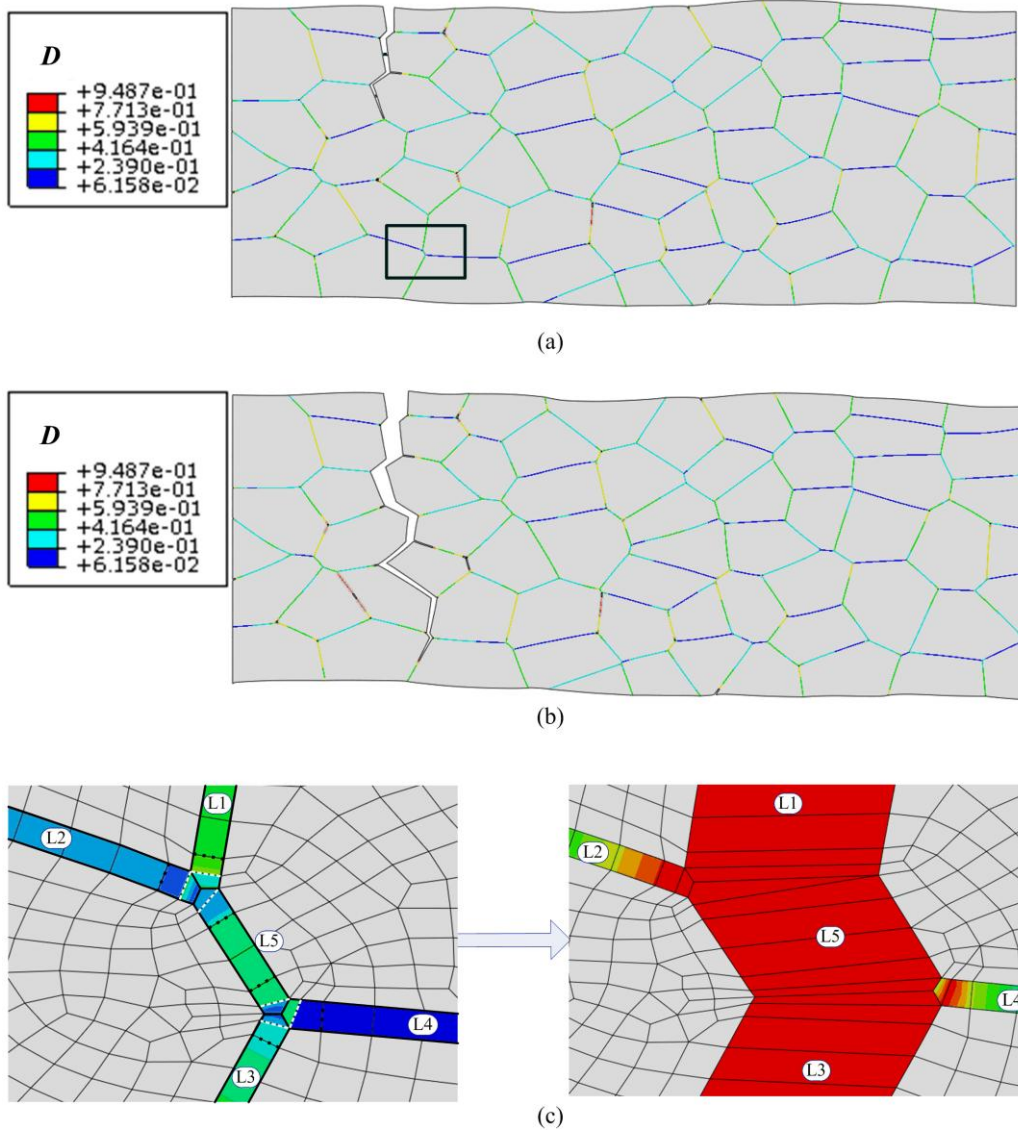


Figure 5.13. Plane strain uniaxial tension CPFE simulation with $t_n^0/t_t^0 \rightarrow 1$. (a) Crack initiation; (b) complete failure; (c) grain boundary triple point before and after the crack passing through it, where L1 - L5 are cohesive layer units.

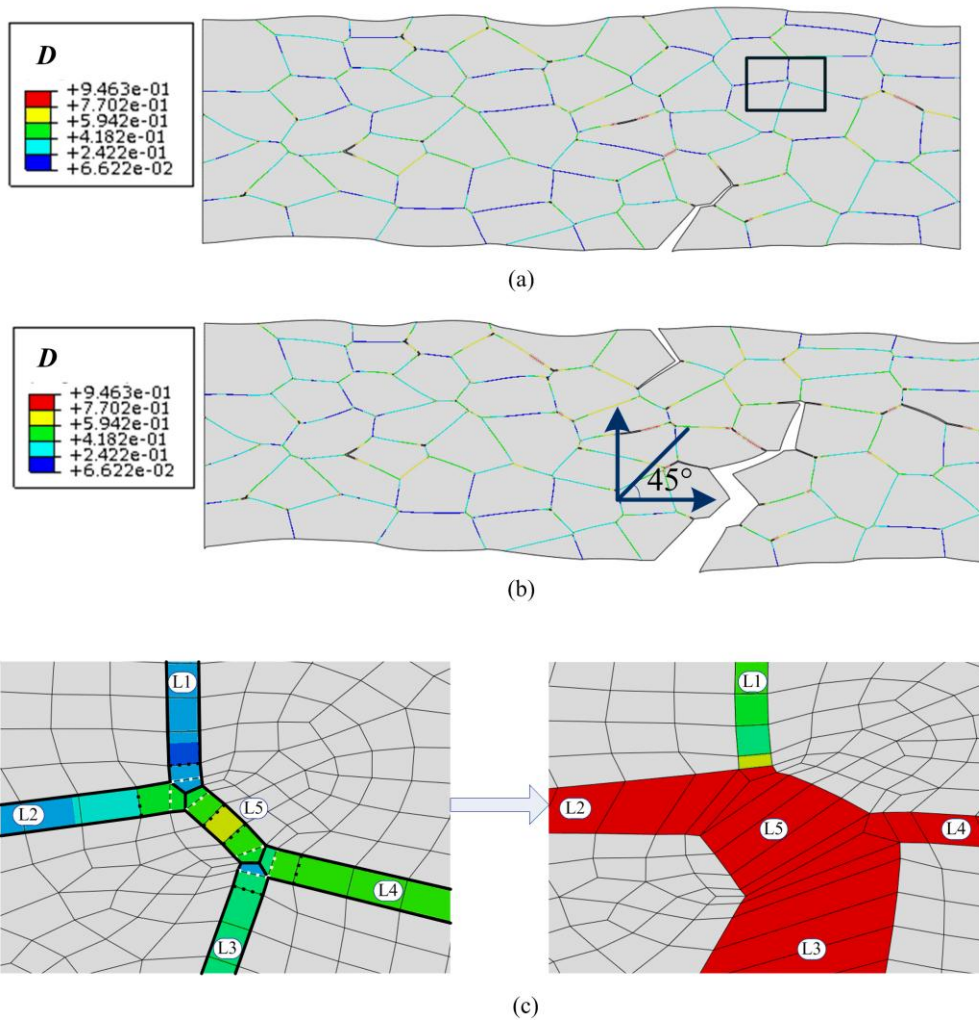


Figure 5.14. Plane strain uniaxial tension CPFE simulation with $t_n^0/t_t^0 \rightarrow \infty$. (a) Crack initiation; (b) complete failure; (c) grain boundary triple point before and after the crack passing through it, where L1 - L5 are cohesive layer units.

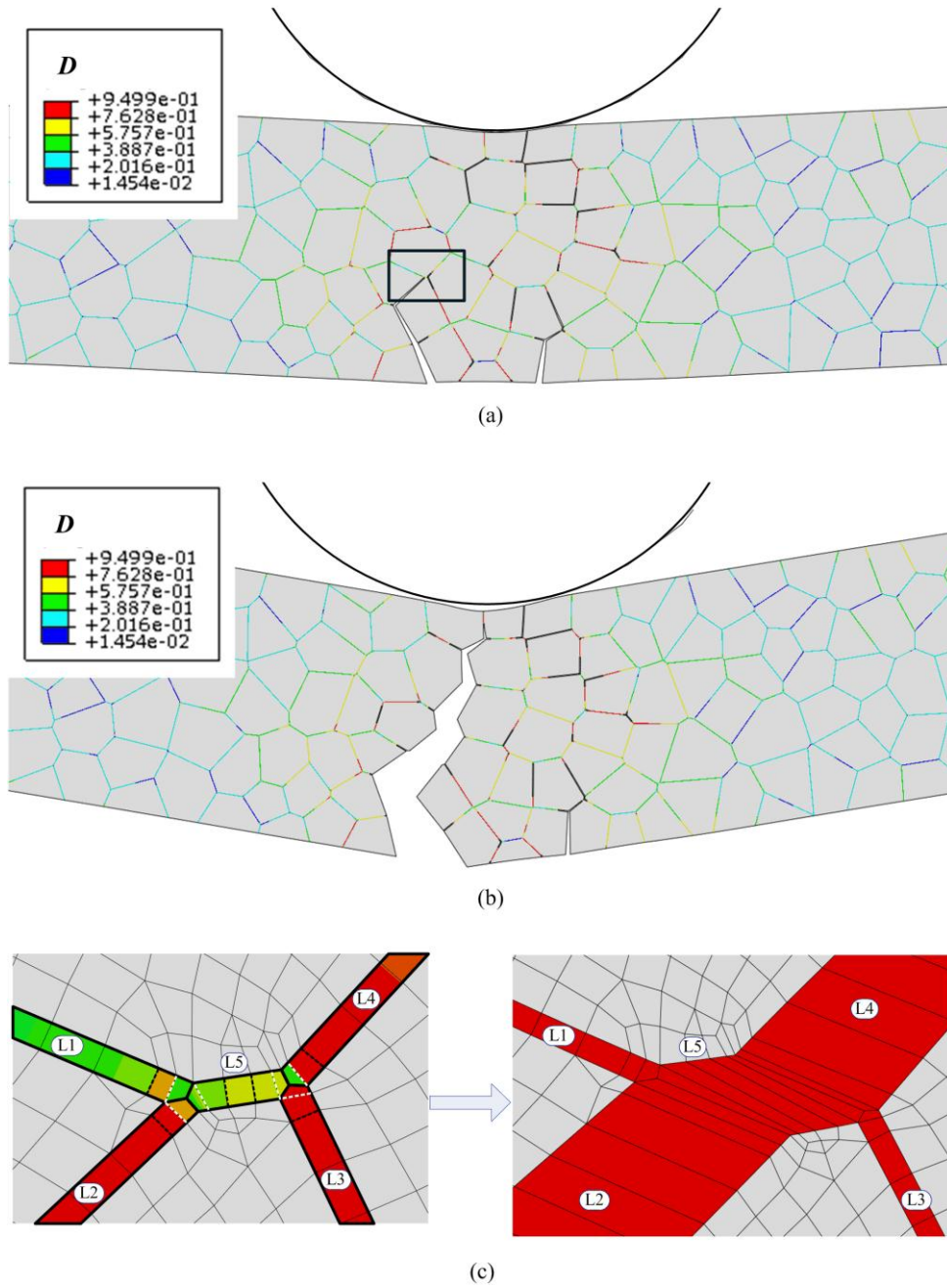


Figure 5.15. Schematic diagrams of the three-point bending model for CPFE simulation.

Chapter 6

3D CPVT models

6.1 Introduction

In this chapter, the CPVT model has been further developed to generate three-dimensional (3D) virtual grain structures. A virtual grain structure generated using the 3D-CPVT model has the property that its grain volume distribution is statistically equivalent to the actual grain structure in term of the specified physical parameters: the mean grain volume, a small grain volume value, a large grain volume value, and the percentage of grains within that range. Development of the 3D-CPVT model requires three steps: 1) Defining the regularity, α , that specifies the uniformity of a tessellation, and deriving the control parameter, δ , based on the regularity, 2) establishing the mapping from the regularity to the distribution parameter of a one-parameter gamma distribution, i.e. $c(\alpha)$; and 3) defining the mapping from the set of physical parameters to the distribution parameter.

A regularity parameter is proposed to evaluate the uniformity of a 3D tessellation. Its dual relations with control parameters and distribution parameter are addressed. Wherein, the relations between the regularity and the distribution parameter, for a range of regularity values, are determined by a comprehensive set of statistical experiments. Data fitting for the grain volume distribution model is in each case obtained by an evolutionary optimisation algorithm, due to the difficulty in identifying a proper distribution parameter for highly regular grain structures. In conjunction with the results of the preceding chapter, a 3D-CPVT model is constructed. Implementation of the entire system is explained and the corresponding computational procedures are provided. To demonstrate the proposed scheme and the application for CPFE simulations, CPFE analyses of compression of micro-pillars are carried out. The effects of both regularity and grain size on the deformation are studied.

6.2 Model development

6.2.1 General considerations

6.2.1.1 Regularity

As discussed before, grain shape and size in a VT are determined by two factors: the grain seed lattice and grain growth velocity. Under the assumption of a homogeneous crystallisation process, for a VT, the grain morphology and size distributions are completely determined by the initial seed lattice. Due to the complexity of the initial seed lattice, grains in different Voronoi tessellations vary in both topological and morphological characteristics. The topological and morphological characteristics of a three-dimensional Poisson Voronoi tessellation have been intensively studied (e.g. [152, 153, 192-194]). The empirical results and rules relating to the mean volume and surface area of a 3D PVT, the number of vertices and edges of a typical grain and the length of a typical edge have been derived. In addition, a few higher level properties, such as the density of grain vertices or grain centroids, the expected edge length density and face area density were also obtained.

Although these empirical laws are useful in charactering PVTs in terms of topological and morphological relations, they cannot be applied to evaluate the degree of randomness of Voronoi tessellations in general. For example, the tessellation in Figure 6.1 (b) has a much more regular grain structure than that in Figure 6.1 (a). However, intuitive judgements must be quantified based on a general criterion. In Chapter 4, a regularity parameter, α , was proposed for planar Voronoi tessellations. This parameter is a relative quantity defined by the ratio of the minimum seed spacing in the given VT, and the regular distance in the equivalent regular hexagonal tessellation, which assumes the most regular planar tessellation. The parameter α takes values between 0 to 1, such that $\alpha = 1$ corresponds to the regular hexagonal tessellation, and $\alpha = 0$ to a Poisson Voronoi tessellation.

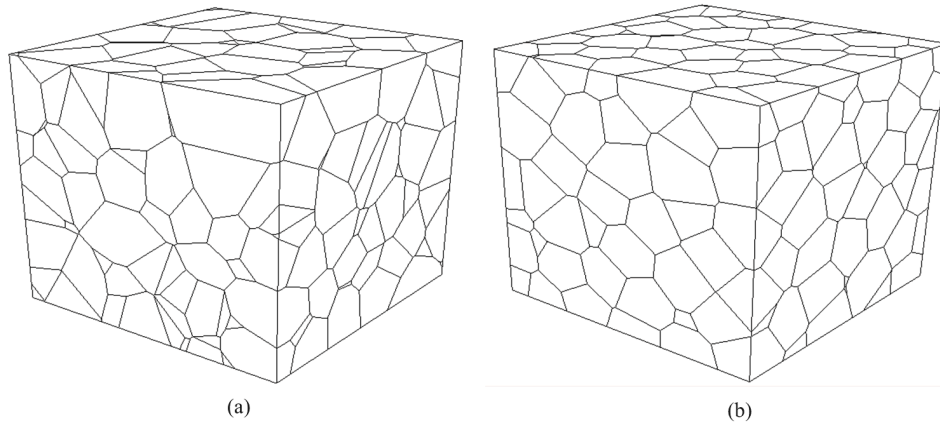


Figure 6.1. Voronoi tessellations with different degrees of uniformity. (a) An irregular structure of a PVT type; (b) a regular structure of a non-PVT type.

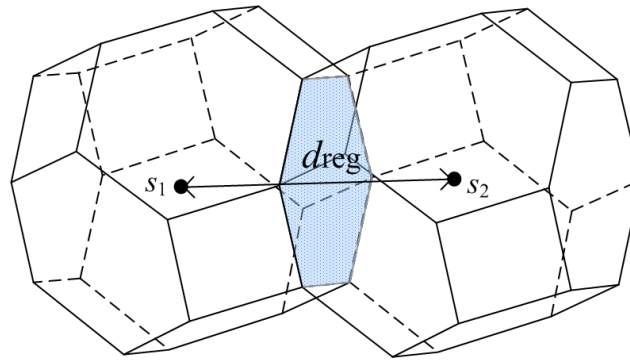


Figure 6.2. Definition of the regular distance d_{reg} , where s_1 and s_2 are seeds taken from a BCC lattice.

For a three-dimensional Voronoi tessellation, a similar parameter was proposed by Zhu et al. in [195, 196]. The 3D regularity, α , is defined by the ratio of the minimum seed spacing of the tessellation to that of the equivalent regular truncated octahedral tessellation. As aforementioned, there are basically three types of uniform tessellations consisting of cubes, rhombic dodecahedron, and truncated octahedron. With comparison of the surface area per unit volume, presented in Table 2.1, a uniform tessellation filling with truncated octahedrons, i.e. tetrakaidecahedral tessellation (TT), is regarded as the

most regular tessellation, since it has the minimum value of surface area per unit volume. Note that as a fully ordered 3D Voronoi tessellation, the seeds in a TT are arranged in a BCC lattice. A tetrakaidecahedron has 14 planar sides, 8 are regular hexagons and 6 are squares. As illustrated in Figure 6.2, two adjacent grains in a TT are coincident at regular hexagonal faces and the seed distance must be equal to

$$d_{\text{reg}} = \frac{\sqrt{6}}{2} \left(\frac{D_{\text{mean}}}{\sqrt{2}} \right)^{1/3}, \quad 6.1$$

where D_{mean} is the mean grain size (in terms of volume) of the given VT. Therefore, the regularity parameter α can be defined as

$$\alpha = \frac{\delta}{d_{\text{reg}}}, \quad 6.2$$

where δ is the minimum seed spacing in the given VT. Note that, as the regularity α decreases from 1 to 0, the corresponding tessellation becomes more disordered, changing from a uniform TT to a fully-random tessellation, i.e., a PVT.

6.2.1.2 Control parameter

The minimum seed spacing, δ , used to define a VT's regularity by Eq. 6.2, also provides an alternative way to generate a VT with regularity control. That is, a 3D Voronoi tessellation with a specified regularity can be achieved by controlling the generation of the seed lattice for the VT, with a minimum allowable seed spacing distance, δ , defined by

$$\delta = \alpha d_{\text{reg}}. \quad 6.3$$

Explicitly, given a Cartesian coordinate system and a cubic domain with volume V , seeds are placed in the cubic domain by generating x , y and z coordinates sequentially from a uniform distribution random number generator. A constraint is introduced requiring that a subsequent seed is accepted only if the distances from it to the other existing seeds are

all larger than or equal to δ . This process is continued until N_{seed} points have been generated. This seed generation process is referred to as a 3D-CPVT seed generation process and the minimum seed spacing, δ , is used as the control parameter of a 3D-CPVT model. Note that, since seeds in the final seed lattice are controlled by a minimum distance, the final tessellation tends to have a slightly larger regularity than the specified value. Moreover, as the number of seeds N_{seed} increases, the regularity of a resultant VT asymptotically approaches the specified value.

It worth noting that, this seed generation process is conducted within a given domain, that is, when forming a VT tessellation, no outside seeds are to be considered. Thus, the local space surrounding to the domain boundaries is totally taken by the grains, whose seeds are near to these boundaries. At this point, it is different from an observed real microstructure cut from a polycrystalline material. In quantitative metallography, grains cut by domain boundaries are counted as half grains, since parts of outside grains remain in the observed area and also parts of the internal grains bordered with boundaries are removed. In contrast, for a VT determined by the 3D-CPVT seed generation process, grains cut by boundaries are assumed to be whole grains rather than half grains. That is, the number of seeds N_{seed} to be generated is equivalent to the number of grains N ,

$$N_{\text{seed}} \approx N, \quad 6.4$$

where the number of grains N is determined by means of the domain volume V_0 and the mean grain size (in terms of grain volume) D_{mean} ,

$$N = V_0/D_{\text{mean}}. \quad 6.5$$

In addition, the 3D-CPVT seed generation process does not require periodicity prescribed around the boundary areas. Therefore, when using the estimation of the number of grains in Eq. 6.4, random noise may be induced. On the other hand, an increasing number of grains within the specified domain diminish the effect of this random noise.

6.2.2 Regularity and grain size distributions

As aforementioned, the regularity parameter α takes a dual relation: It is used to derive the control parameter δ and also links to the physical parameters *via* the distribution parameter c . As addressed in Chapter 3, the physical parameters are employed for users to characterise grain size distribution properties of a desired grain structure, which are modelled by a one-parameter gamma distribution function. Therefore, if the relation between the grain size distribution of VTs and their regularities can be established, the 3D-CPVT model is fully developed.

6.2.2.1 Volume distribution modelling

In Chapter 4, the relation between a range of regularity values and the grain area distributions was modelled for 2D Voronoi tessellations. The grain size distribution was fitted by a one-parameter gamma distribution function, and the results provided a non-linear one-to-one mapping between the regularity and the distribution parameter (Eq. 4.6).

The grain size (volume) distribution of a 3D PVT was modelled by a variety of distribution functions: a Maxwell speed distribution function [146], as well as by three-, two- and one-parameter gamma distribution functions [147-150]. The one-parameter gamma distribution function (Eqs. 3.15-3.16) only requires the specification of one parameter and also provides a quality fit for the normalised grain size distribution of a 3D VT. Therefore, it is most convenient to build the mapping from grain size distribution to the regularity of a 3D tessellation, i.e. from c to α .

6.2.2.2 Statistical data of c to α

The relation of c to α is based on empirical results from a comprehensive statistical tests conducted in this section. Voronoi tessellations were produced based on the above 3D-CPVT seed generation process, i.e., without periodicity, within a cubic domain. With considerations of reducing random noise around surface regions, for each Voronoi tessellation, 10^5 grains were generated. For each regularity value, 100 Voronoi

tessellations were produced independently, which were used for a one-parameter gamma distribution function to fit a grain volume distribution for a specified regularity.

The data fitting using a parameter gamma distribution function is usually achieved using moment estimator [197]. However, in the case of the regularity value exceeding 0.7, the grain size values are all concentrated about the mean. To identify the distribution parameter for such a narrow distribution, the moment estimator does not give a sufficiently good result. Therefore, an optimisation procedure has been employed to fit the grain size distribution in this work. The one-parameter gamma distribution is given in Eqs. 3.15-3.16. The mean value of this distribution is one, hence 10 intervals given by $I_i = 0.3 \times [i, i + 1)$, where $i = 0, 1, \dots, 9$, within a truncated domain $[0, 3]$ are used to evaluate the difference between the model and the statistical data; it can be observed from the statistical data in Figure 6.3 that for the most random case $\alpha = 0$, there are very few grains with $D/D_{\text{mean}} > 3$.

The least squares error function is defined by

$$E(c) = \sum_{i=1}^n (f_i(c) - \mu_i)^2 \quad 6.6$$

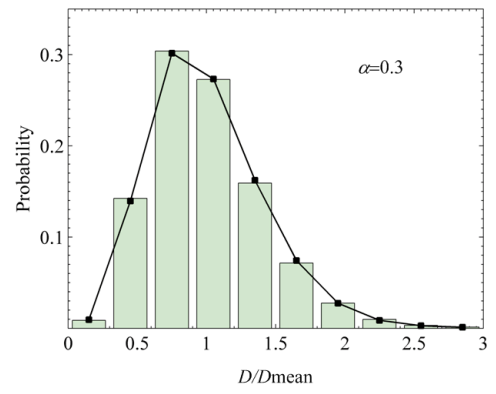
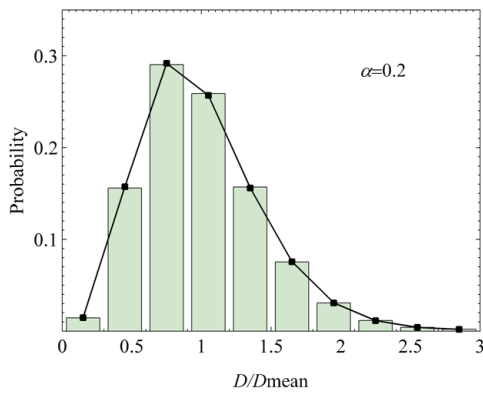
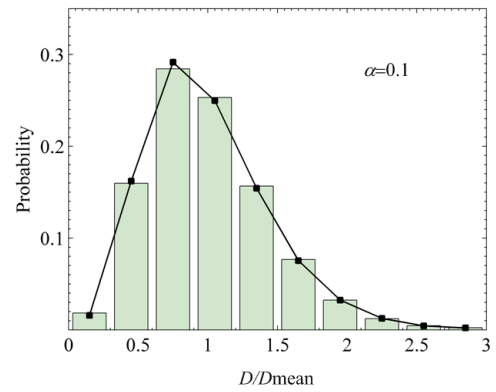
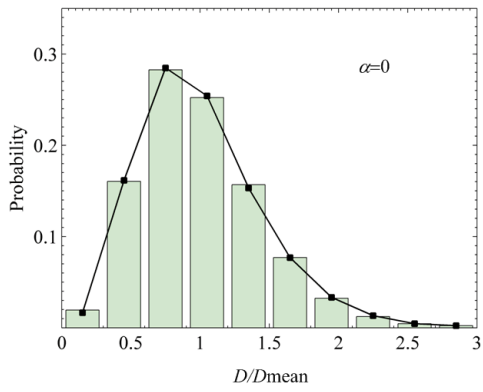
where $f_i(c)$ is the ideal probability in the interval I_i , i.e.,

$$f_i(c) = \int_{0.3i}^{0.3i+0.3} P_{x,x+dx} dx, \quad 6.7$$

and μ_i is the frequency of the grains whose sizes are within the interval I_i . The CMA-ES optimisation algorithm [161] was used to find the appropriate c values for corresponding regularity values. The optimisation results are presented in Table 6.1, where the second row lists the distribution values for the model fits and the third row list the related objective function values, defined by Eq. 6.6. illustrates the difference between the statistical data and the model fits. It can be observed that the one parameter gamma distribution provides highly accurate fit to grain size distributions for VTs with regularities $\alpha \leq 0.6$, while some small inaccuracies are apparent for higher regularities.

Table 6.1. The fitting results including the relations between the regularity α and the distribution parameter c , and the corresponding fitting errors defined in Eq. 6.6.

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.75	0.8
c	5.156	5.209	5.446	6.210	7.702	10.609	16.031	28.971	38.310	48.187
$E(c)$	1.9E-5	1.9E-5	1.2E-5	1.5E-5	6.7E-5	1.1E-4	4.6E-5	2.1E-3	5.7E-3	1.1E-2



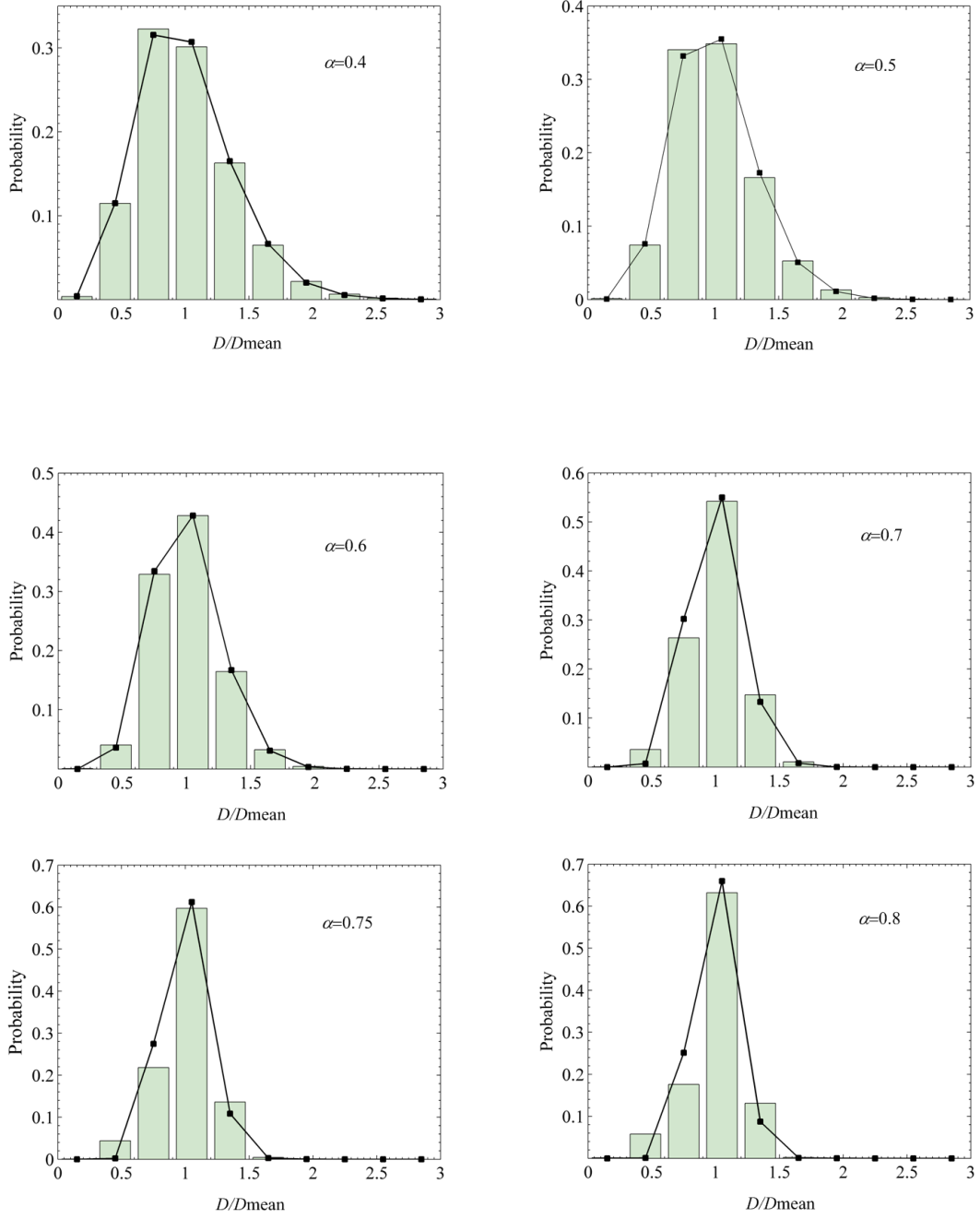


Figure 6.3. Comparison of the statistical data (bars) and the model fits (points) for different regularity values.

6.2.2.3 Descriptive model of c to α

The data pairing between the regularity measurement α and the corresponding model parameter c , presented in Table 6.1, reveals a non-linear mapping, whereas α increases, c increases accordingly. In order to fully establish this mapping over a continuous interval $\alpha \in [0,1)$, a descriptive model proposed in Eq. 4.6 was also utilised for the three dimensional model. The experimental data pairs in Table 6.1 were used to calibrate the constants in Eq. 4.6. Determination of the model parameters in Eq. 4.6 was performed using the highly efficient optimisation algorithm, CMA-ES, and the results are $c_0 = 5.156$, $c_m = 370$, $A = 2.4$, $k = 0.33$ and $n = 1.5$. The final calibrated model and the $\alpha - c$ data pairs used in the calibration are plotted in Figure 6.4.

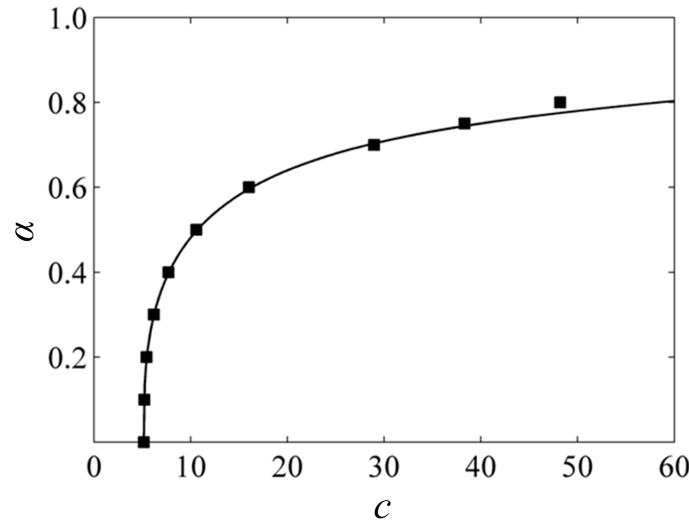


Figure 6.4. The descriptive model, relating the regularity parameter c to the distribution parameter α of a one-parameter gamma distribution function.

6.2.3 Implementation

The 3D-CPVT seed generating scheme, described in Section 6.2.1, combined with Voronoi tessellation modelling, presented in Section 3.2.1, is referred to as the three-dimensional controlled Poisson Voronoi tessellation model, denoted as a 3D-CPVT model. A flow diagram of the CPVT model was given in Figure 3.2, where the domain Ω is in 3D space

and the grain sizes are in terms of volume. The 3D-CPVT model is a three-dimensional adaptation of the general CPVT model; hence it is capable of generation of a virtual 3D grain structure with desired grain volume distribution specified *via* a user input procedure. A summary of the procedure is as follows:

Step 1. The parameters characterising the domain grains of the structure are used to determine:

- a. The domain Ω of a virtual grain structure.
- b. The number of seeds, N_{seed} (using Eq. 6.4).
- c. The regular distance, d_{reg} (using Eq. 6.1).
- d. The control parameter, δ , (using Eq. 6.3).

Step 2. The physical parameters are input by the user, followed by the operations:

- a. Compute the distribution parameter c by the Newton-Raphson method (using Eqs. 3.28-3.32).
- b. Derive the regularity parameter α from the empirical model $c(\alpha)$ (using Eq. 4.6 with the calibrated constants in Section 6.2.2).
- c. Combine c and α with the obtained regular distance d_{reg} to compute the control parameter δ (Eq. 6.3)).

Step 3. Once the required parameters are achieved, the 3D-CPVT seed generation process is applied to generate seeds, hence the corresponding virtual grain structure.

The following points are worth noting when applying the CPVT model to generate VTs with regularity control: 1) If the control parameter $\delta = 0$, there is no constraint on the seed generating process, and the resultant Voronoi tessellations are Poisson Voronoi tessellations; 2) generating a VT with regularity larger than 0.8 is not always successful, because the randomly produced seeds affect on the subsequent seed locations; the larger the regularity value, the smaller the probability of successfully achieving the VT; and 3) the proposed CPVT model is based on an empirical relation between c and α ,

which was achieved using a cubic domain without periodicity. Thus, grains built by the border of the domain affect the number of grains, hence the average grain area D_{mean} , which in turn affects the regular distance d_{reg} . Therefore, increasing the number of grains in a VT will reduce the random noise and produce better regularity control performance.

6.3 CPFE application

To demonstrate the proposed 3D CPVT model, simulations of uniaxial compression of micro-pillars are studied. The overall setup is shown in Figure 6.5, where the tapered cylinder micro-pillar has a height of $180\ \mu\text{m}$, bottom radius of $30\ \mu\text{m}$ and angle of 2° . A displacement of $U = 54\ \mu\text{m}$ (equivalent to 0.3 engineering strain) was applied on the upper face of the micro-pillar in all the simulations; the bottom face was fixed and the remaining faces were free of constraint. Two sets of simulations were conducted, one of which focuses on the influence of the grain structure's regularity, i.e. the grain size distribution characteristics, on the compression deformation, and the other concentrates on the effect of grain size on the micro-pillar's inhomogeneous behaviour.

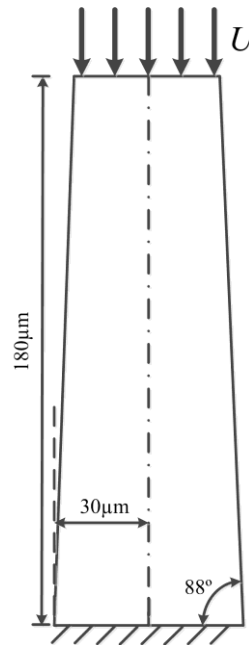


Figure 6.5. Schematic diagram for the CPFE simulation of micro-pillar compression.

The finite element analysis in this demonstration uses a rate-dependent finite deformation crystal plasticity formulation to describe the deformation of individual grains within a polycrystalline material. The elasticity is assumed to be isotropic, defined by the Young's modulus and Poisson's ratio. The shear strain rate on the a^{th} system, $\dot{\gamma}^a$, is related to the resolved shear stress τ^a , as defined in [198] by,

$$\dot{\gamma}^a = \dot{A}^a \left(\frac{\tau^a}{g^a} \right) \left(\left| \frac{\tau^a}{g^a} \right| \right)^{n-1}, \quad 6.8$$

where \dot{A}^a is a reference strain rate on slip system a , n is the stress exponent, and g^a is the current hardness of the slip system. Obviously as n increases to infinity, the material model becomes rate-independent [199]. The resolved shear stress on the a^{th} slip system, τ^a , is given by

$$\tau^a = m_i^a \sigma_{ij} s_j^a, \quad 6.9$$

where m_i^a is the slip plane normal, s_j^a is the slip direction, and the stress tensor σ_{ij} is given in terms of the elastic strain by

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl}^e. \quad 6.10$$

The number of slip systems and their orientation depends on the crystal structure, e.g., an FCC crystal contains four slip planes and each slip plane has three independent slip directions, hence $N = 12$; the simulations here are for an FCC material. The plastic strain rate is given by

$$\dot{\varepsilon}_{ij}^p = \frac{1}{2} \sum_{a=1}^N (m_i^a n_j^a + m_j^a n_i^a) \dot{\gamma}^a, \quad 6.11$$

and the stress rate takes the form

$$\dot{\sigma}_{ij} = C_{ijkl}(\dot{\varepsilon}_{kl} - \dot{\varepsilon}_{kl}^p), \quad 6.12$$

where $\dot{\varepsilon}_{kl}$ is the total strain rate tensor, $\dot{\varepsilon}_{kl}^p$ the plastic strain rate tensor, and C_{ijkl} is the fourth-order stiffness tensor. Furthermore, the strain hardening of the slips system for FCC crystals is defined by

$$\dot{g}^\alpha = \sum_{b=1}^{12} h_{ab} \dot{\gamma}^b, \quad 6.13$$

where h_{ab} is the matrix of slip hardening moduli. If $a = b$, h_{aa} , the self-hardening moduli are given by:

$$h_{aa}(\gamma) = h_0 \text{sech}^2 \left| \frac{h_0 \gamma}{g_s - g_0} \right|, \quad 6.14$$

where h_0 is the initial hardening modulus, g_0 is the initial shear strength, and g_s is the break-through stress. If $a \neq b$, h_{ab} , i.e. h_{ab} , the latent hardening moduli are defined by

$$h_{ab}(\gamma) = q h_0 \text{sech}^2 \left| \frac{h_0 \gamma}{g_s - g_0} \right|. \quad 6.15$$

In Taylor's isotropic-hardening assumption, the self- and latent-hardening rates are assumed to be identical, hence the hardening factor q is set to unity. Moreover, the accumulated shear strain γ is defined by

$$\gamma = \sum_{a=1}^{12} |\gamma^a|. \quad 6.16$$

In the initial state, $\sigma_{ij} = 0$, $\gamma^\alpha = 0$, $g^\alpha = g^0$ and $\varepsilon_{ij} = 0$. The material constants were calibrated for a free cutting tool steel [191] for a temperature of 1100°C. Young's modulus and Poisson's ratio are 6.06 GPa and 0.3, respectively. The other material

constants in Eqs. (6.8)-(6.16) are presented in Table 6.2. This crystal plasticity constitutive model has been implemented in the commercial FE package ABAQUS using the user defined material model subroutine, VUMAT, for which the fundamental procedure was introduced in [200].

Table 6.2. The values of material constants for the crystal-plasticity constitutive equations in Eqs. (6.8)-(6.16).

n	$a(s^{-1})$	$h_0(\text{MPa})$	$g_s(\text{MPa})$	$g_0(\text{MPa})$
3	10	33	150	23

Table 6.3. Physical parameters and corresponding grain structure properties for the two microstructures.

Label	Physical parameters (μm^3)				Mean grain diameter [†] (μm)	CPVT model parameters			
	D_{mean}	D_L	D_R	P_r (%)	\tilde{D}_{mean}	d_{reg} (μm)	α	δ (μm)	N
R-1	14140	7070	21210	76.6	30	26.4	0.0	0	46
R-2	14140	9900	18380	60.3	30	26.4	0.4	10.5	46
R-3	14140	11310	16970	87.6	30	26.4	0.8	21.1	46
S-1	8180	5730	10630	60.3	25	22.0	0.4	8.8	78
S-2	4190	2930	5450	60.3	20	17.6	0.4	7.0	154
S-3	1770	1240	2300	60.3	15	13.2	0.4	5.3	361

[†] Assuming a spherical grain shape.

In the first set of simulations, three specimens with different grain structure regularities were generated. As shown in Table 6.3, the specimen R-1 is a purely random grain structure, i.e., $\alpha = 0$; the specimen R-2 has regularity $\alpha = 0.4$, and specimen R-3 has the most regular grain structure, $\alpha = 0.8$. The physical parameters used to generate the three grain structures are presented in Table 6.3, where the implementation parameters derived by the CPVT model are also given. Note that, the grain size input to the CPVT model is in terms of grain volume, while the equivalent grain size assumed a

spherical shape. The undeformed micro-pillars with the specified regularities are shown in Figure 6.6. It should be mentioned that repeated use of the same physical parameters using the CPVT model will result in slightly different grain structures, but all the virtual grain structures are statistically consistent with the specified physical parameters. The grain structures shown in Figure 6.6 also have crystallographic orientations, assigned by the VGRAIN system using a random number generator based on a uniform distribution. In the VGRAIN system, grain orientations can be assigned based on a fixed texture, or randomly taken from a uniform or normal distribution. In addition, grain orientations can also be defined according to measurements, such as EBSD.

The deformed specimens shown in Figure 6.6 are demonstrations of regularity and grain size control in three dimensions, and compare the different deformation characteristics resulting from different grain size distribution properties. The three specimens have an identical mean grain size of 30 μm . The contours of accumulated shear strain (c.f. [191]) show that the degree of plastic strain localisation is lower for sufficiently regular grain structures; the maximum value of γ is lower for the $\alpha = 0.8$ case, relative to the irregular grain structures, $\alpha = 0.4$ and $\alpha = 0$. The axial deviation is also lower for $\alpha = 0.8$ than it is for either $\alpha = 0.4$ or $\alpha = 0$, although there is inherent variability depending on the details of the grain orientation distribution. Irregular grain structures have an inherently larger variability in deformation features, such as axial deviation in this case. For a given average grain size, a higher irregularity leads to a larger maximum grain size which may or may not dominate the deformation in a given cross section depending upon its orientation, which is randomly assigned. In contrast, a regular grain structure has a more homogeneous grain size distribution, which precludes the occurrence of larger grains that may accommodate severe slip localisation if appropriately oriented, i.e. having a high Schmid factor for the boundary conditions. The force-displacement curves shown in Figure 6.8 also confirm the behaviour, where it can be seen that the response is identical for all pillars until the onset of softening due to strain localisation, which occurs for a lower displacement value the more irregular the grain size distribution, whereas the regular model exhibits persistent hardening throughout the deformation.

Figure 6.7 shows micro-pillar specimens with different mean grain sizes and the same regularity ($\alpha = 0.4$, moderately irregular); the normalised grain size distributions are statistically equivalent. Axial deviation of the deformed sample increases with increasing mean grain size for a given regularity, and a material with a smaller mean grain size exhibits a more homogenous deformation. This is clearly shown by the simulation in Figure 6.7 (c) where the maximum value of the accumulated shear strain, γ , is considerably lower and the cylinder has maintained its shape and not deviated axially after deformation. The force versus displacement curves given in Figure 6.9 illustrate that strain localisation reduces the applied force required to achieve a given pillar displacement, and that the onset of softening is delayed by a smaller grain size to the point that in the case of $d = 15 \mu\text{m}$ there is no evident softening due to strain localisation and the material continues to harden steadily. Note that, according to the simulation results obtained from different grain structures with different regularities and different mean grain sizes, it can be found that the property of mean grain size have more significant influence on the mechanical responses than the characteristics of regularity, i.e., grain size distribution.

6.4 Conclusion

Representing a grain structure within a FE/CAE computational environment is essential for micromechanics simulations. Two approaches may be used, depending on the requirements of the application: Computer reconstruction of an actual grain structure using, e.g. EBSD, or high-fidelity virtual grain structures generation for accurate prediction of material deformation. In order to correlate the latter with actual material in an average sense, the virtual grain structure must be statistically equivalent to the actual material, have the desired grain size distribution features and exhibit realistic grain morphology. In this chapter, a three-dimensional virtual grain structure generation model, the 3D CPVT model, has been developed to fulfil these requirements. The three-dimensional CPVT model involves four steps: (1) Determination of the grain size distribution (i.e., the distribution parameter c of a one-parameter gamma distribution) based on specified physical parameters, (2) derivation of the tessellation's regularity, α ,

from the mapping $\alpha(c)$, (3) calculation of the control parameter, δ , based on the regularity and the desired mean grain size, and (4) generation of the virtual grain structure using the 3D CPVT model with respect to the specified physical parameters. The most important advantage of using the 3D CPVT model to generate a virtual grain structure is that the grain size distribution of a resultant grain structure is statistically equivalent to the user-specified physical parameters: Mean grain volume, a small grain volume, a large grain volume and the percentage of grains within that range. The algorithms are also very computationally efficient, allowing very large grain structures to be generated in real time. Therefore, this method is ideal for FE simulations requiring virtual grain structures, providing realistic grain structure representation in terms of both grain morphology and grain size distribution.

Derivation of the 3D-CPVT model required: 1) Defining a regularity parameter, α , that specifies the uniformity of a tessellation, 2) deriving the control parameter, δ , based on the regularity, 3) establishing the mapping from the regularity to the distribution parameter of a one-parameter gamma distribution, i.e. $c(\alpha)$, and 4) defining the mapping from the set of physical parameters to the distribution parameter c . The relations between the regularity and control parameters, and the regularity and distribution parameters, were given. The relation between the regularity and the distribution parameter was determined by a comprehensive set of statistical experiments. Data fitting for the grain volume distribution model was in each case obtained by an evolutionary optimisation algorithm, due to the difficulty in identifying a proper distribution parameter for highly regular grain structures. Implementation of the entire system was explained and the corresponding computational procedures were provided. To demonstrate the 3D-CPVT model for micromechanics studies, CPFE analyses of compression of micro-pillars were carried out exploring the effects of grain size and regularity of the grain size distribution on the deformation characteristics.

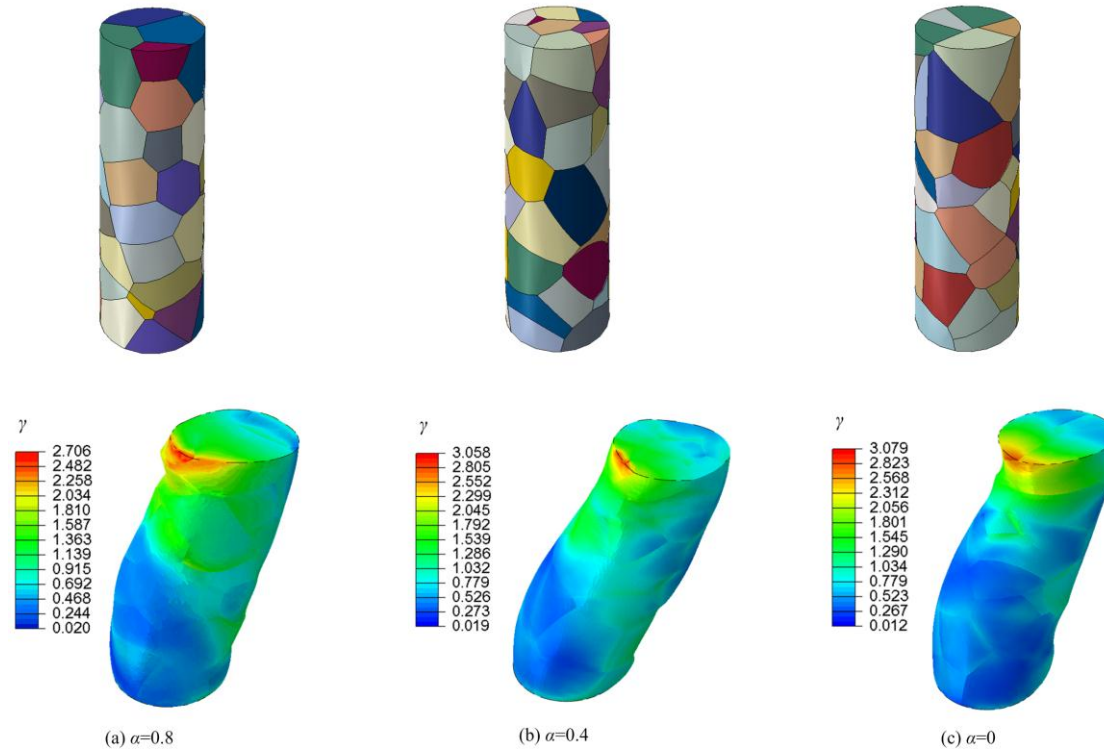


Figure 6.6. CPFE simulations of Micro-pillar compression for regular and irregular grain structures.

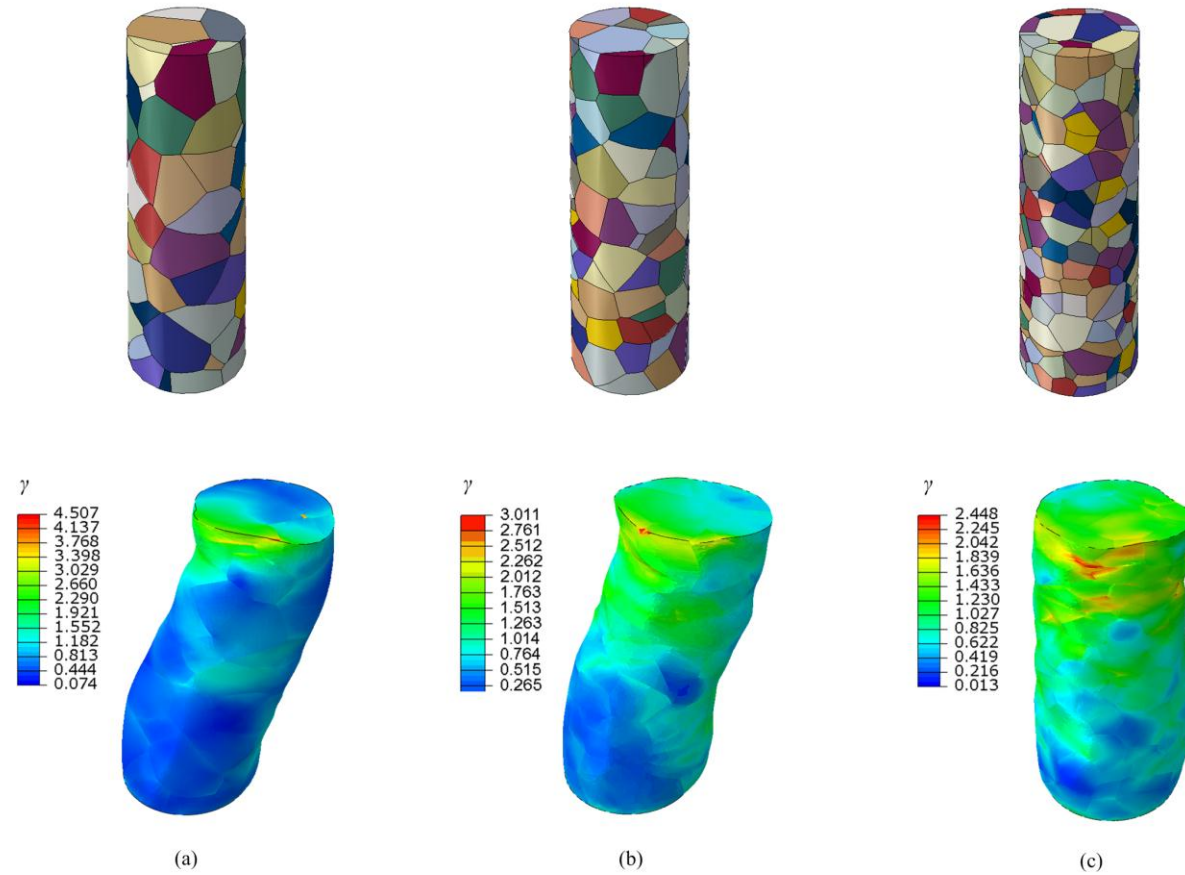


Figure 6.7. CPFE simulations of Micro-pillar compression for grain structures with different grain sizes.

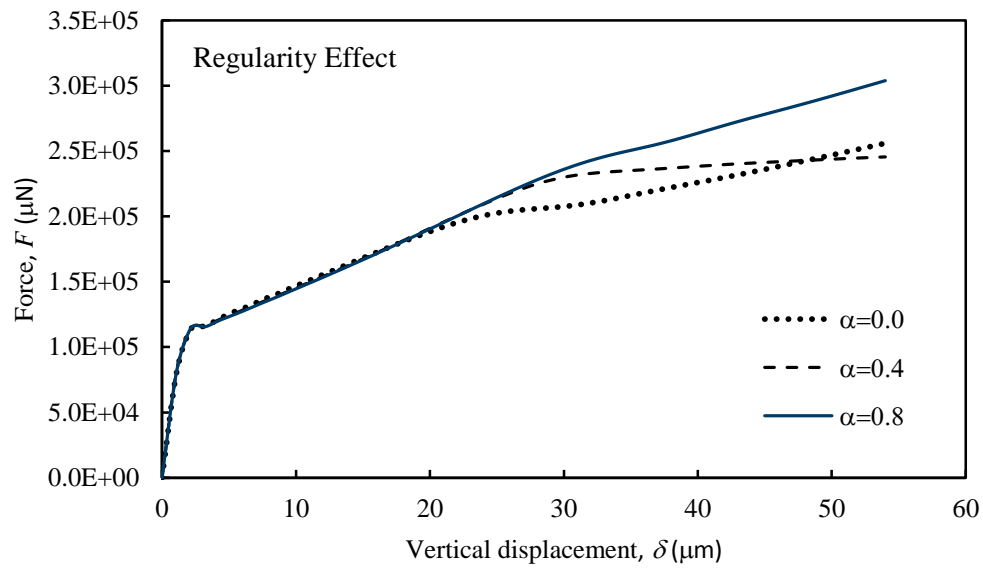


Figure 6.8. The force – displacement properties during compressing the micro-pillars with different regularities.

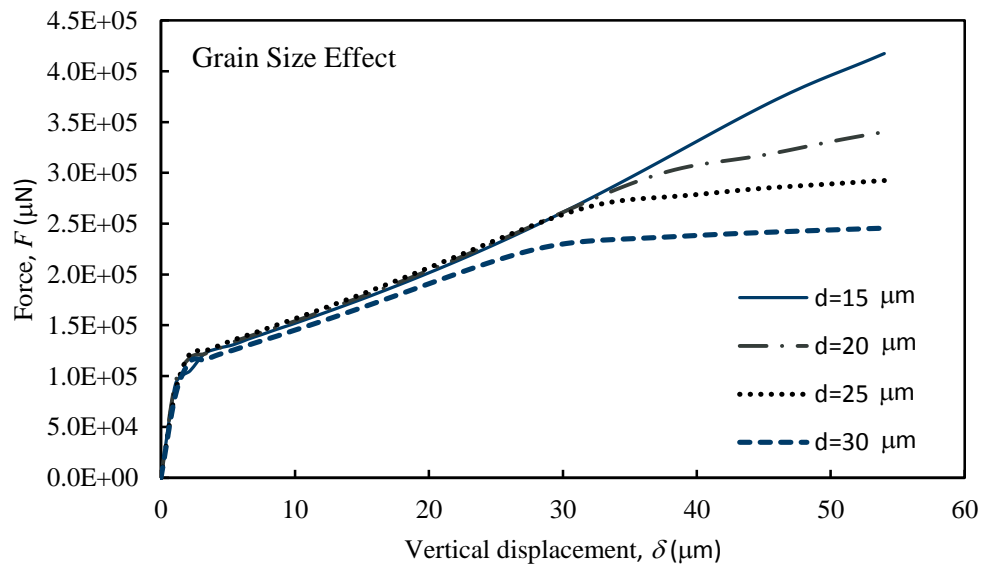


Figure 6.9. The force – displacement properties during compressing the micro-pillars with different grain sizes.

Chapter 7

Advanced Models

7.1 Introduction

In the preceding studies, the 2D- and 3D-CPVT models have been developed for the generation of virtual grain structures with grain size distribution control. In the grain structures generated by the CPVT models, grains are equiaxed and cross the grain structure domain, all grains share identical grain size distribution properties, i.e., their grain size distribution is unimodal. A grain structure with these features is called homogeneous. However, for generating true grain structures, there are some typical variations, where grain structures may not be always homogeneous, which must be considered for advanced micromechanics simulations. Two of the most important cases are gradient grain structures and grain structures with elongated grains.

This chapter aims at presenting a multi-zone CPVT scheme for generating inhomogeneous grain structures, especially for the gradient grain structures. The scheme employs the CPVT models to generate grains simultaneously in multiple zones and fills a grain structure domain with the desired grain size properties for individual zones. In the next section, desired functionalities of the multi-zone CPVT model are presented according to true features from real grain structures. Here, the multi-CPVT model has been proposed for gradient structure generation and an additional discussion is presented for deformed grain structures with non-equiaxed grains. In addition, a series of cases have been demonstrated to show the capabilities and applications of the multi-zone CPVT model, and comparisons have been carried out with real grain structures and virtual grain structures from Monte Carlo simulations.

7.2 Motivations

Although, in industry, metallurgical processing technology is usually targeted to ensure the homogeneity of materials, there are many situations where inhomogeneous materials can be

involved. The inhomogeneous grain structures may result from the mixture of different material substances, multiple phases, recrystallisation, or joints of dissimilar materials (e.g. welding, coating), due to a variety of metallurgical processes. Based on the grain size variation properties across one dimension, there are basically two types of gradient grain structures with continuous variation of grain size and discontinuous variation of grain size.

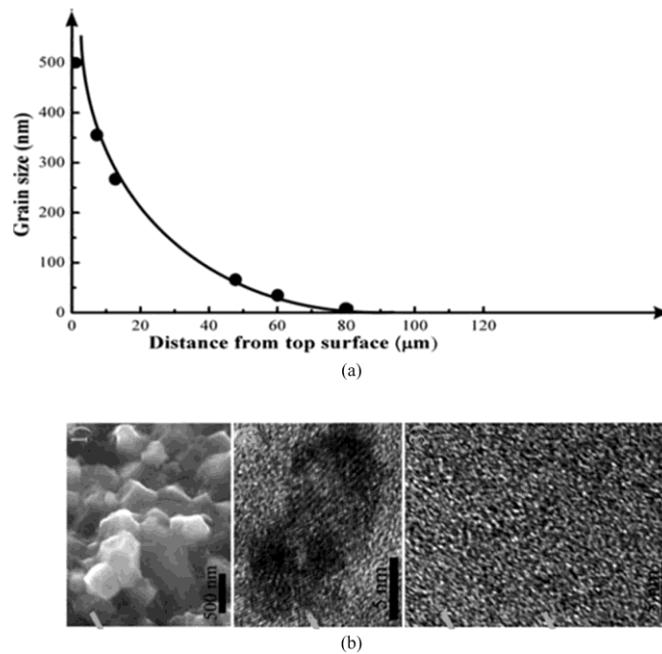


Figure 7.1. A gradient grain structure formed by processing Ti-based BMGs *via* surface mechanical attrition treatment (SMAT) (c.f. [201] for details). (a) The spatial distribution of gradient mean grain sizes; (b) the three micrographs taken from the surface to the deep matrix.

For the first type, the size of the grains across the domain varies gradually and can be depicted with a smooth distribution function with respect to a specified dimension. This type of grain structure is commonly presented in a material that has been work hardened *via* mechanical processes, for example, cold-rolling. This harnesses the effect of plastic strain to refine grains in metals and alloys in order to substantially enhance their properties. The resultant processed material has a finer grain size in the surface region compared to the base material. Figure 7.1 shows a gradient grain structure of Ti-based bulk metallic glasses (BMGs) processed *via* surface

mechanical attrition treatment (SMAT). The spatial distribution curve implies a continuously varying characteristic in terms of grain size.

A gradient grain structure with a discontinuous variation of grain size generally comprises a set of individual grain aggregates with each grain aggregate having a common properties and grain size distribution. Figure 7.2 shows a case of this type of gradient grain structure. This grain structure was observed from a weld zone of AA5754 after friction stir welding. In the central zone, traces of oxide inclusions were mixed into the weld, where the grains are distinctively smaller than those in the neighbouring grain aggregates. This type of grain structure mostly occurs under situations such as joining dissimilar materials or grain recrystallisation.

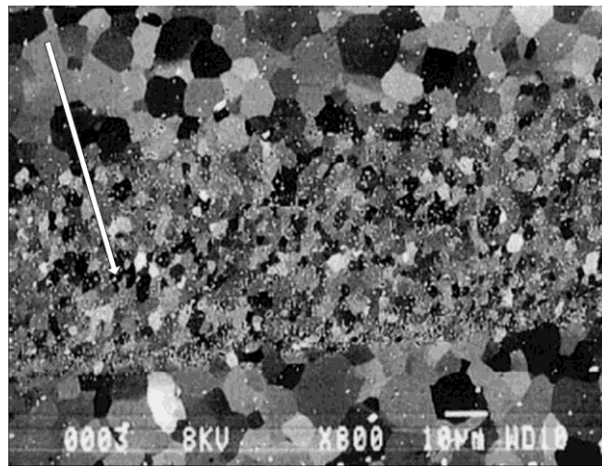


Figure 7.2. A grain structure of weld joint of an AA 5754 FSW weld zone, where fine grains in the middle area are a trace of oxide inclusions from surfaces of a butted interface [202].

As previously mentioned, grain size distribution properties significantly affect various material mechanical properties from both the macro-scale, for example, the plastic flow stress, to the micro-scale, in the form of subgrain stress or strain distributions. Understanding of the corresponding mechanical properties of a gradient grain structure is crucial for developing or tailoring the properties of advanced materials. For materials with gradient mean grain size, the macro-scale mechanical properties, for example, hardness, shear and tensile behaviours, have attracted a lot of attention (e.g. [203-205]). However, the micro-scale mechanical features have

not been studied in sufficient detail. One particular issue is the lack of effective grain structure representation methods for efficient generation of a gradient grain structure with desired mean grain size gradient and grain size distributions along the mean grain size gradient zones.

7.3 The multi-zone CPVT model

7.3.1 Multi-zone settings

In the preceding work, the 2D- and 3D-CPVT models have been developed for corresponding grain structure generation. The CPVT models aim to quickly generate a virtual grain structure with grain size distribution control within a given grain structure domain. However, all these features can only be achieved under the assumption that the grains in a grain structure exhibit a homogeneous grain size distribution. That is, grain sizes can be depicted by a unimodal distribution function. Since in a gradient grain structure, different regions have different grain size distribution properties, the CPVT models cannot directly apply to generation of this type of virtual grain structure, and hence a scheme of employment of the CPVT models must be provided to produce virtual grain structures with desired mean grain size gradient and multi-modal grain size distribution control.

Note that for a gradient grain structure with continuous or discontinuous size variation, a domain can be divided into a set of sub-regions according to their mean grain size and size dispersion characteristics. In each individual region, the grain size distribution can be approximated as unimodal. For a continuous gradient grain structure, distribution features slowly vary along a dimension, therefore, there can be a reasonable division of the whole domain into a set of piecewise zones with the property that grains belonging to each individual zone share an identical distribution. For example, the gradient grain structure shown in Figure 7.1 has the grain size spatial distribution feature that its grain size D follows the tendency

$$D = 500 \times e^{-\frac{x}{20}}, \quad 7.1$$

with respect to the distance x to the material surface. A feasible division of the domain into three zones along the depth from the surface is illustrated in Figure 7.3. The three zones have the

mean grain sizes: 316 nm, 95 nm and 15 nm respectively, and in each zone, the grain structure along the thickness direction has similar uniformity and grain size distribution characteristics. In contrast, for discontinuous gradient grain structures, a structural domain usually consists of a set of grain aggregates with distinct grain size features. Since the aggregates can be recognised according to their sizes, a convenient division of the grain structure domain is based on the regions of the aggregates.

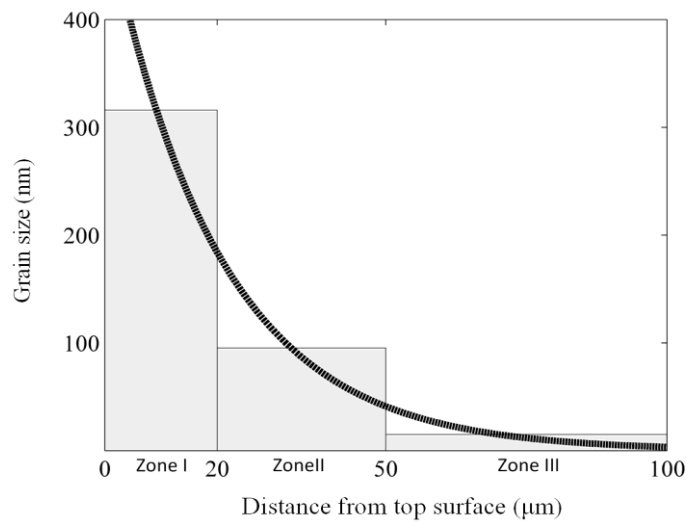


Figure 7.3. Illustration of the domain of a gradient structure in Figure 7.1 into three zones.

For the example of the region specified along the arrow in Figure 7.2, the arrow spanned two zones from one with larger grain size to the other with smaller grain size. A two-zone structure can be used to decouple the gradient grain structure into two separate grain structures with their own grain size features, as shown in Figure 7.4. The mean grain sizes of zone I and zone II are 12 μm and 4 μm respectively. In addition, for each zone, grains can be modelled by a unimodal distribution function.

In general, the scheme of dividing the domain of a gradient structure into multiple zones is referred as to multi-zone scheme. The fundamental idea is that grain size in each individual zone can be modelled by a unimodal function. Therefore, the proposed 2D- and 3D-CPVT models can

be correspondingly applied to generate a virtual grain structure within an individual zone. The remaining issue is the mechanism of processing seed generation and the formation of grain structure at the region near to the borders of adjacent zones.

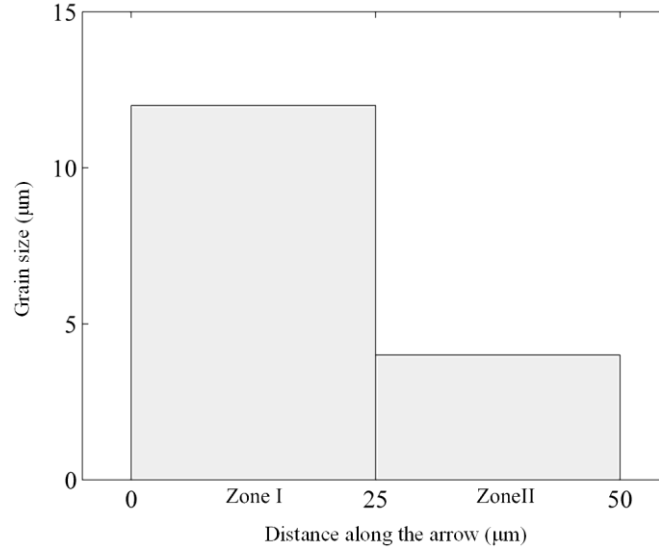


Figure 7.4. Illustration of the domain of a gradient structure in Figure 7.2 into two zones.

7.3.2 Coupling multiple zones

The multi-zone scheme operates a set of adjacent zones, having their own grain size distributions, to fill the grain structure domain. A straightforward method is to directly stack them side by side, but this fails to consider the interactions of grains near to the zone boundaries. The direct combination of individual grain structures forms lines from zone boundaries, and more significantly, fails to account for the natural morphological variation of the grains from one zone to the other zone. To avoid these issues, the following multi-zone CPVT model is proposed to generate gradient grain structures with grain size distribution control for individual zones.

To generate a multi-zone grain structure, an initial multiple zone layout must be provided, which includes a set of M sub-domains, i.e., $\{\Omega^t\}_{t=1}^M$, where t denotes the zone index and $t \in [1, M]$. For each of the sub-domains, the grain size distribution properties are assigned by an individual group of physical parameters $\{D_{\text{mean}}^t, D_L^t, D_R^t, P_r^t\}$. Each group of physical parameters is taken as

input to a 2D- or 3D-CPVT model and correspondingly the control parameters $\{\delta^t\}$ can be achieved respectively. A multi-zone CPVT model is defined by such a seed generation process that seeds are consecutively generated based on the following rules:

1. Within an individual zone, the minimum seed spacing is decided by the control parameter of this zone.
2. The minimum distance of a pair of seeds spanning two zones is determined by the smaller control parameter of the two zones

More specifically, given the physical parameters and workpiece-related parameters for M sub-domains, there are M sets of implementation parameters $\{\Omega^t, N_{\text{seed}}^t\}_{t=1}^M$ and control parameters $\{\delta^t\}_{t=1}^M$ prepared for the multi-zone CPVT. Then for the zone $t \in [1, M]$, let d_t denote the distance of a pair of seeds within the zone t ; let d_{ts} represent the distance of a seed pair spanning the zone t and its neighbour s . If the seeds within the zone t are generated according to

$$\begin{aligned} d_t &\geq \delta_t \\ d_{st} &\geq \min_{s \in M_z \cap s \neq t} \{\delta_t, \delta_s\} \end{aligned} \quad 7.2$$

a Voronoi tessellation can be formed and this virtual grain structure generation scheme is renamed as a multi-zone CPVT.

Figure 7.5 (a) presents a grain structure generated using the multi-zone CPVT model. In this grain structure, two zones show different mean grain sizes: the mean grain area in Zone II is 10 times of that in Zone I, while the two zones have identical uniformity with the tessellation's regularity. In addition, the control parameters of Zone I and Zone II, i.e., δ_I and δ_{II} , have the relation of $\delta_I < \delta_{II}$. The detailed physical parameters are listed in Table 7.1. In accordance with the 2D-CPVT mode, the distance of any seed pair in Zone I is no less than δ_I , and that in Zone II is no less than δ_{II} . That is, the generation of seeds for any individual zone is independent of the other. However, the two individual zones are coupled to each other and share interfacial grains. Based on the seed generation rule, the minimum distance of seed pair spanning two zones is determined by the minimum control parameter of the two zones, that is, they are defined by δ_I . In comparison with the multi-zone grain structure in Figure 7.5 (a), the grain structures in Figure

7.5 (b) and (c) depict the Voronoi tessellation formed from the seeds within the given zone. It shows that within each zone, grain structures are identical due to the same seed lattice, while the interfacial grains are different due to the influence of neighbouring seeds.

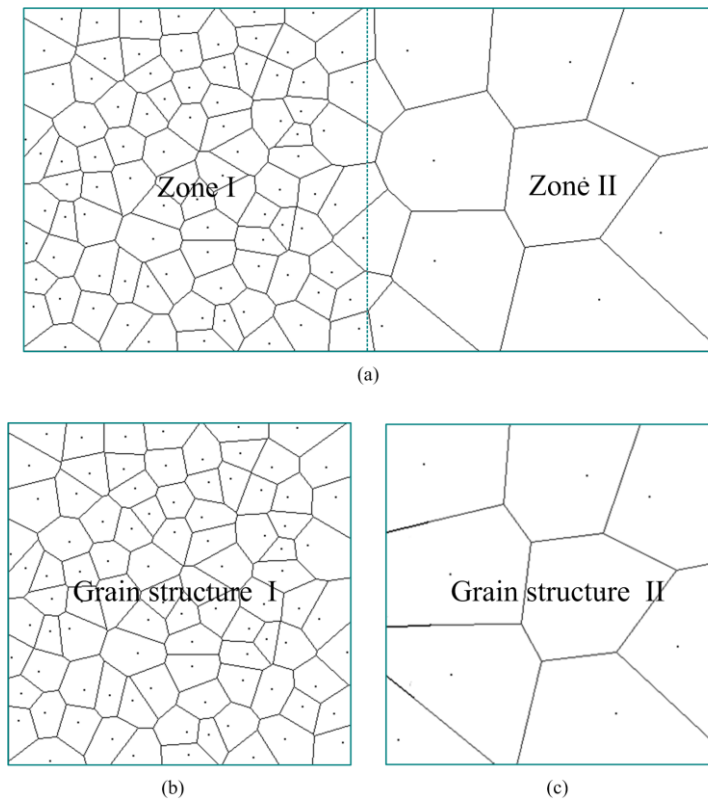


Figure 7.5. Comparison of the CPVT model with the multi-zone CPVT model. (a) A multi-zone grain structure generated based on the control parameters $\{\delta_I, \delta_{II}\}$; (b) the grain structure formed based on the seed lattice in Zone I; (c) the grain structure formed based on the seed lattice in Zone II.

Table 7.1. Physical parameters for grain structure I and II in Figure 7.5.

Label	Physical parameters (μm^2)				Equivalent grain size [†] (μm)			Model parameters		
	D_{mean}	D_L	D_R	P_T	d_{mean}	d_L	d_R	d_{reg} (μm)	α	δ (μm)
Zone I	200	100	300	90%	15.2	10.7	18.6	15.2	0.5	7.7
Zone II	2000	1000	3000	90%	48.1	33.4	58.9	48.1	0.453	24.2

[†] Assuming a hexagonal grain shape.

7.3.3 Property of interface region

As aforementioned, for each individual zone, the grains not in direct contact with the interfacial region between adjacent zones can be treated as an independent CPVT grain structure. But the shape and size of a grain spanning a zone boundary tends to be transformed due to the different seed lattice property of the two sides, i.e., the minimum seed spacing. Figure 7.6 highlights the interfacial grains of the grain structure in Figure 7.5 (a), where the dark nodes show the seeds of their neighbouring grains and the dot-line depicts the interfacial boundary of the original two zones in Figure 7.5 (b) and (c). According to the property of a Voronoi tessellation, i.e. a grain boundary is a bisector of two adjacent seeds, the seed lattice in Figure 7.6 determines that boundaries of the interfacial grains need to be translated forwards to zone II due to the difference of minimum seed spacing between two zones. Therefore, each grain in the two-zone grain structure of Figure 7.6 has a generally larger grain area than the grain with the correlated seed in Figure 7.5 (a). In contrast, the grains near to the interfacial grains in zone II reduce their grain areas comparing to those in Figure 7.5 (b), due to the translation. This feature can be deemed as a natural grain size gradient between two adjacent zones, which is generally analogous to the gradient grains in metallographic observations. In summary, the multi-zone CPVT model has two important advantages:

- a. The flexibility of grain size distribution control for individual zones in terms of both mean grain size and also grain structure's regularity.
- b. The smooth transition between two coupled zones, and natural morphological properties of the grains spanning different domains.

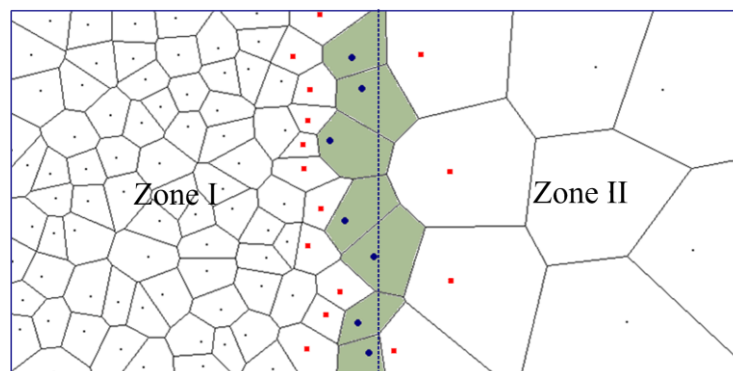


Figure 7.6. Illustration of grains spanning two adjacent zones.

7.4 Case studies

In this section, the proposed multi-zone CPVT model is applied to generate a variety of virtual grain structures to compare with real grain structures and simulated grain structures using the Monte Carlo model. In addition, the mechanism to generate deformed multi-zone grain structure, where the shape of grains in some zones is elongated, is also demonstrated. It is noted that the gradient grain structures using for comparison with the multi-zone CPVT model were all adopted directly from literature. Since there are only a few micrographs with the small number of grains presented, only the mean grain size can be well determined, while for other physical parameters including the small grain size, the large grain size and the percentage within the range cannot be accurately specified due to the lack of sufficient information. Therefore, all the following studies only concentrate on the gradient mean grain size and the procedures to implement the multi-zone CPVT model. For other physical parameters relating to the grain structure's regularity only rough estimates have been used.

7.4.1 Comparison of real grain structures

A grain size gradient nano-structured Ni coating was used to derive the physical parameter input for the multi-zone CPVT model. The grain size varies from 22 nm on surface to about 586 nm near the coating–substrate interface observed by transmission electron microscopy [205]. Four micrographs were obtained at the depths of roughly 7 μm , 17 μm , 25 μm and 40 μm . The mean grain sizes of the four regions are 22 nm, 93 nm, 393 nm and 586 nm. Due to the large mean grain size difference between the region of 22 nm and 93 nm, an additional zone with the mean grain size 40 nm, (z5 in Figure 7.7 (b)), was added to the multi-zone layout for the multi-zone CPVT model as well as an extra zone with a mean grain size 250 nm, (z3 in Figure 7.7 (b)). Note that, in the real coating structure, the length ratio between the domain and an individual grain is too large, thus, in order to clearly show small grains and the size gradient, the sizes of domain and corresponding zones were reduced for the virtual grain structure. The multi-zone settings and physical parameters for individual zones are listed in Table 7.2. Figure 7.7 (b) shows a resultant grain structure, which simulates the smooth grain size gradient from zone 1 to zone 5.

For any individual zone the normalised grain size distribution property is independent to others and can be generated based on the related physical parameter input. Moreover, repeated generation of this five-zone gradient virtual grain structures can be easily performed with the same input and equivalent results in terms of grain size distributions. It is worth mentioning that the virtual structure presented in Figure 7.7 (b) was taken from ABAQUS/CAE and ready to conduct CPFE simulations.

Table 7.2. Parameters for the multi-zone CPVT.

Label	Zone width (nm)	Grain size		Implementation parameters		
		D_{mean} (nm ²)	$d_{\text{mean}}^{\dagger}$ (nm)	d_{reg} (nm)	α	δ (nm)
Zone 1	2000	1.08E6	586	1120	0.5	560
Zone 2	1200	4.85E5	393	749	0.5	374
Zone 3	500	1.96E5	250	476	0.6	287
Zone 4	1800	2.72E4	93	177	0.5	89
Zone 5	300	5.03E3	40	76	0.6	46
Zone 6	800	1.52E3	22	42	0.5	21

[†] Assuming a circular grain shape.

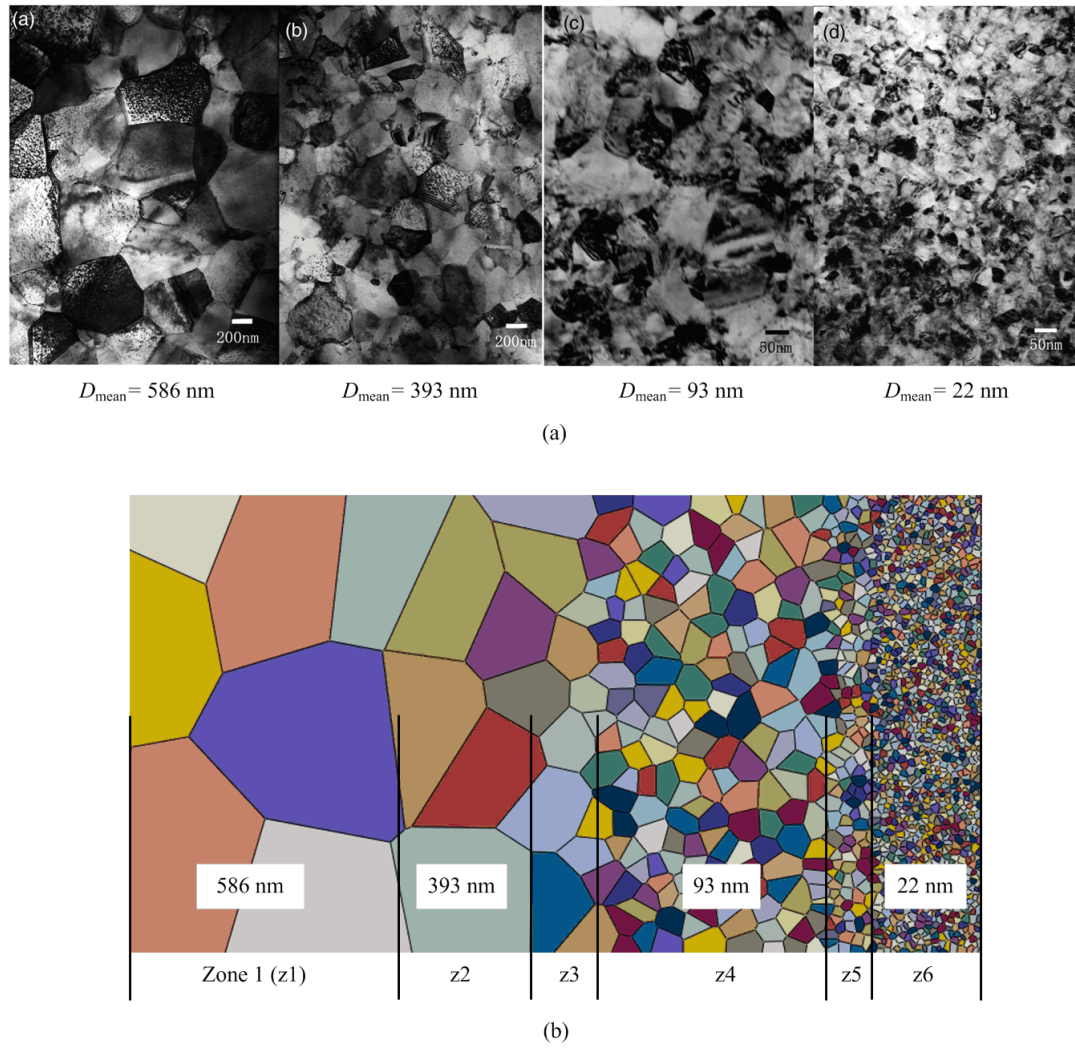


Figure 7.7. Comparison of a grain size gradient nano-structured Ni coating with a virtual grain structure generated using the multi-zone CPVT model. (a) Grain structures in different zones of the coating [205]; (b) a virtual gradient grain structure generated by the multi-zone CPVT model.

7.4.2 Comparison with a numerical model

Numerical models such as the Monte Carlo (Potts) method have been successfully applied to simulate grain structure evolution such as grain growth. Figure 7.8 (a) and Figure 7.9 (a) present two simulated grain structures of the heat affected zones (HAZ) of commercially pure titanium welds using the Monte Carlo model [206]. The chart in Figure 7.9 (a) specifies the mean grain size spatial distribution, and the region highlighted by the rectangles in the grain structure was taken

as a reference for the multi-zone CPVT model to generate equivalent two-dimensional virtual gradient grain structure. The multi-zone settings and physical parameter input are given in Table 7.3, and the generated virtual grain structure using the multi-zone CPVT model is shown in Figure 7.9. Furthermore, a three-dimensional gradient grain structure was generated based on the same mean grain size gradient and grain size distributions, which relates to α . The parameters for the 3D multi-zone CPVT model are provided in, and the generated 3D virtual grain structure is presented in Figure 7.9 which is statistically equivalent to the gradient grain structure in the selected region highlighted by rectangles.

Table 7.3. Parameters for the multi-zone CPVT to generate a 2D virtual grain structure of a HAZ.

Label	Zone width (μm)	Grain size		Implementation parameters		
		D_{mean} (μm^2)	$d_{\text{mean}}^{\dagger}$ (μm)	d_{reg} (μm)	α	δ (μm)
Zone 1	800	3.14E4	200	190	0.5	95
Zone 2	800	1.77E4	150	143	0.5	72
Zone 3	550	7.90E3	100	96	0.5	48
Zone 4	350	1.96E3	50	48	0.5	24

\dagger Assuming a circular grain shape.

Table 7.4. Parameters for the multi-zone CPVT to generate a 3D virtual grain structure of a HAZ.

Label	Zone width (μm)	Grain size		Implementation parameters		
		D_{mean} (μm^3)	$d_{\text{mean}}^{\dagger}$ (μm)	d_{reg} (μm)	α	δ (μm)
Zone 1	800	4.19E6	200	176	0.5	88
Zone 2	800	1.77E6	150	132	0.5	66
Zone 3	550	5.24E5	100	88	0.5	44
Zone 4	350	6.55E4	50	44	0.5	22

\dagger Assuming a spherical grain shape.

It can be found that both 2D and 3D virtual gradient grain structures generated using the multi-zone CPVT model have high quality agreement with the numerically simulated results from the MC model in terms of grain size and morphology. Under the input of the physical parameters relating to the grain size distribution properties and multi-zone settings, the multi-zone CVPT model can repeatedly generate statistically equivalent virtual gradient grain structures without efforts.

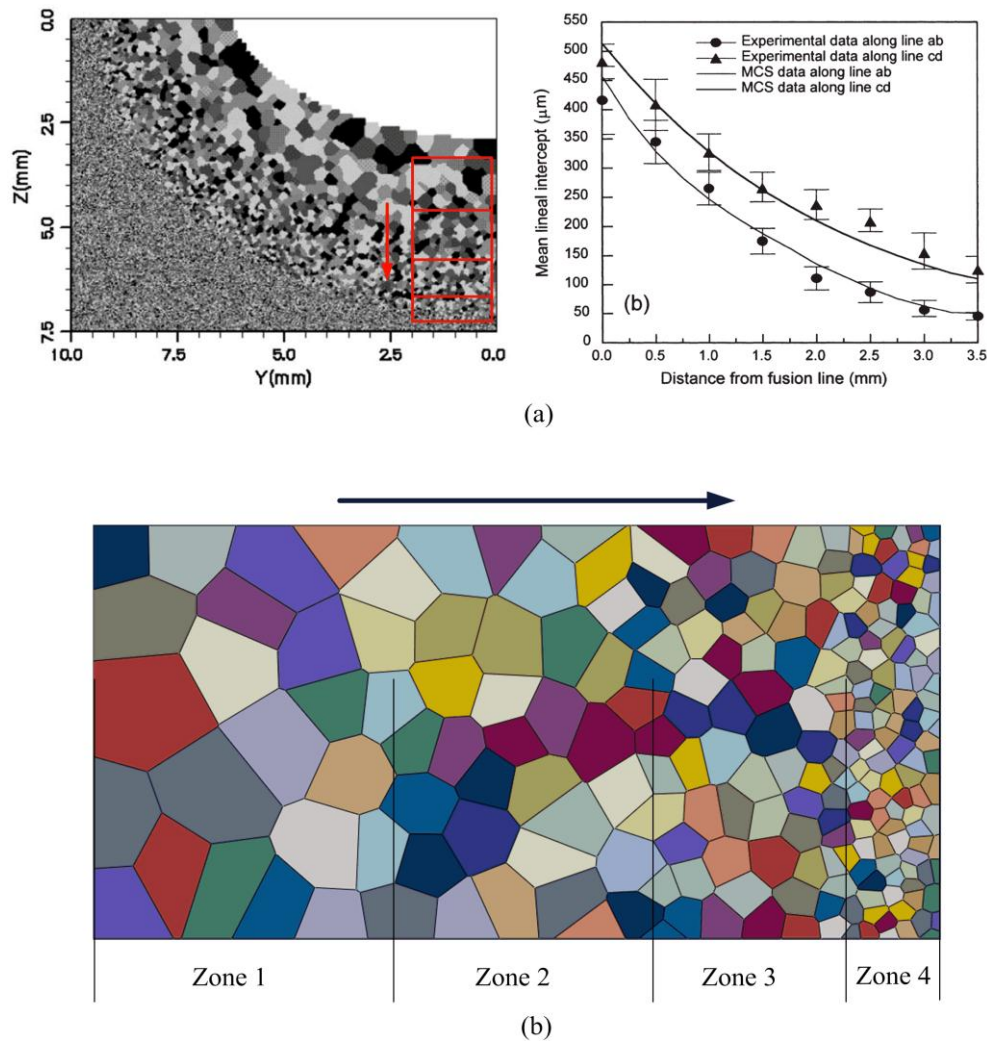


Figure 7.8. Comparison of 2D gradient grain structures obtained by the MC method and the multi-zone CPVT model respectively. (a) A grain structure using the MC method [206]; (b) a virtual gradient grain structure generated by the multi-zone CPVT model.

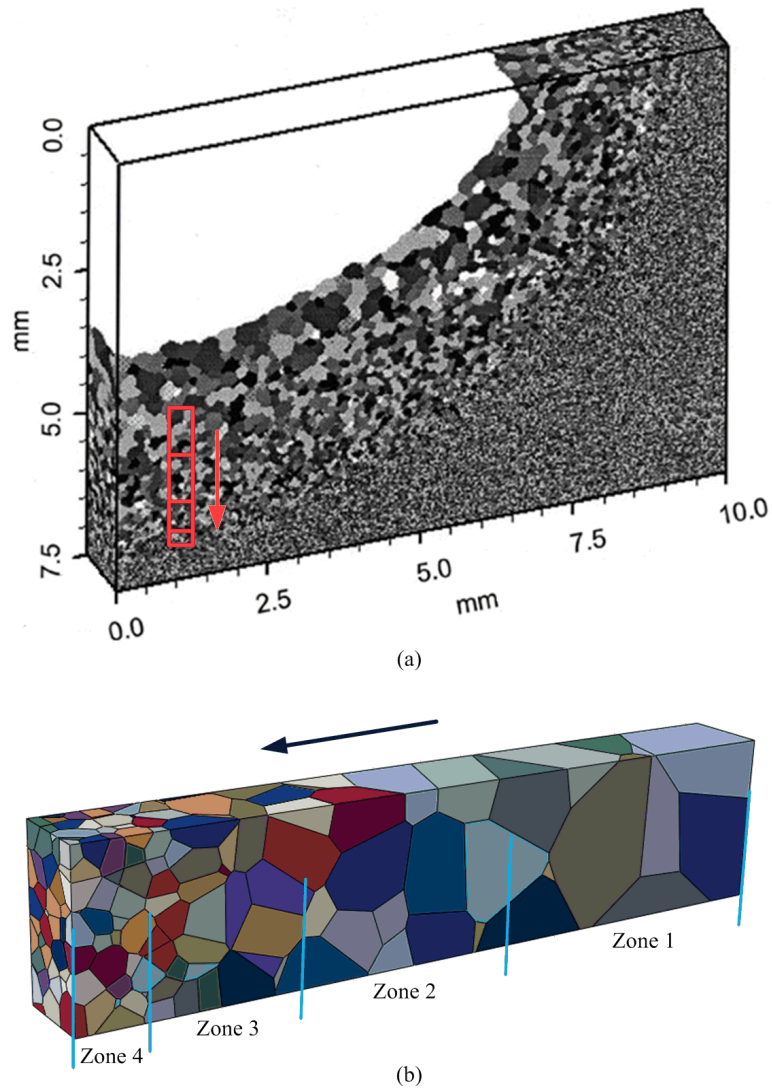


Figure 7.9. Comparison of 3D gradient grain structures obtained by the MC method and the multi-zone CPVT model respectively. (a) A grain structure using the MC method [206]; (b) a virtual gradient grain structure generated by the multi-zone CPVT model.

7.4.3 Deformed structures

As aforementioned, the other typical inhomogeneous grain structure is that consists of deformed grains with the average grain aspect of ≥ 1 . In engineering, this type of grain structures are generally formed in a cold-work, or produced with a special fabrication, for

example, coating structures. Geometry for a planar virtual grain structure with a grain aspect ratio, ρ , can be approximately achieved by transforming the coordinates of all vertices in an equiaxed Voronoi tessellation along the desired long-axis direction. In Figure 7.10, the hexagonal grain is elongated based on the specified aspect ratio value ($\rho \approx 2$), so is the grain structure with three grains. Note that, the area of the elongated grain structure in Figure 7.10 (b) is twice the original one. This is an approximate scheme of achieving a deformed structure in terms of grain size and morphology and is ready to be applied to the multi-zone CPVT model.

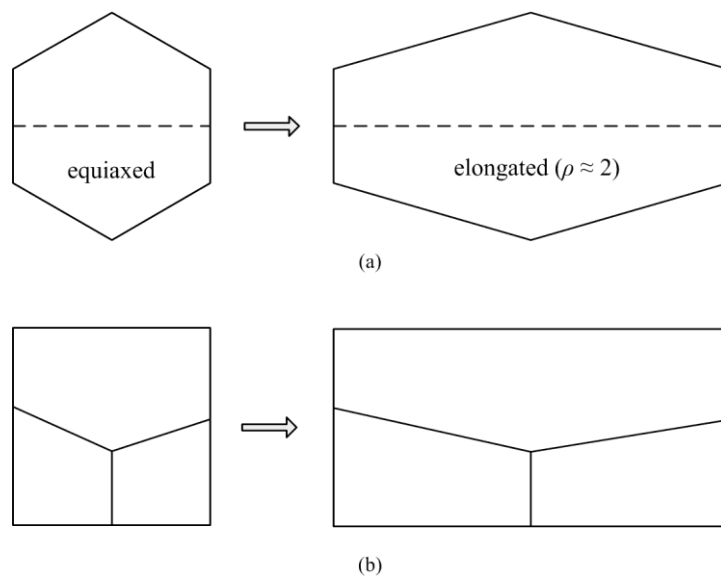


Figure 7.10. Illustration of transforming an equiaxed grain to (a) a Voronoi tessellation; (b) elongated grains.

Figure 7.11 (a) shows a grain structure of thermally grown oxide, which contains two sub-regions of grains with different grain aspect ratio. The upper region has equiaxed grains, while the lower part has columnar grains with aspect ratio approximately equal to 4. In the upper region, the mean grain size is equal to $1.77 \mu\text{m}^2$, while the mean grain size is approximately four times the value in the lower region. Since the grain aspect ratio in the lower region is $\rho = 4$, the grain structure in the lower region can be produced by means of generating equiaxed grain structure with the mean grain size $D_{\text{mean}} = 1.77 \mu\text{m}^2$ and then transforming the coordinates of equiaxed grains to elongated grains with $\rho = 4$. Table 7.5 presents the physical parameters as input to the multi-zone CPVT model. Initially, a two-zone grain structure with equiaxed grains was

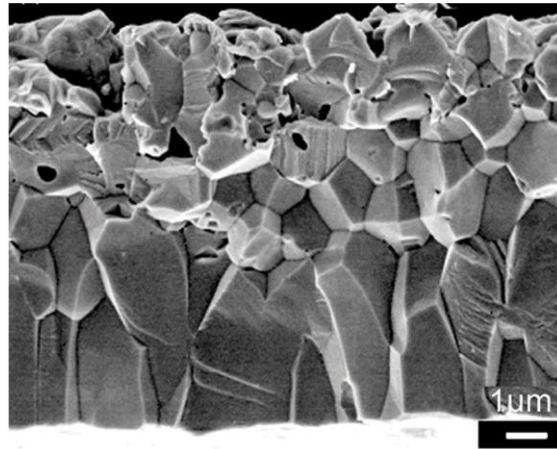
generated using the multi-zone CPVT model. Then, for Zone 2, grains were elongated by transforming the vertices' coordinates based on the given aspect ratio value of Zone 2. Figure 7.11 (b) presents a generated virtual grain structure, which is statistically equivalent to the true grain structure given in Figure 7.11 (a) in terms of the specified physical parameters in Table 7.5.

Table 7.5. Parameters for the multi-zone CPVT to generate a 2D virtual grain structure, containing an elongated zone. The parameters for the elongated zone describe the properties of the undeformed grain structure, and the final shape is determined in combination with the specified ρ .

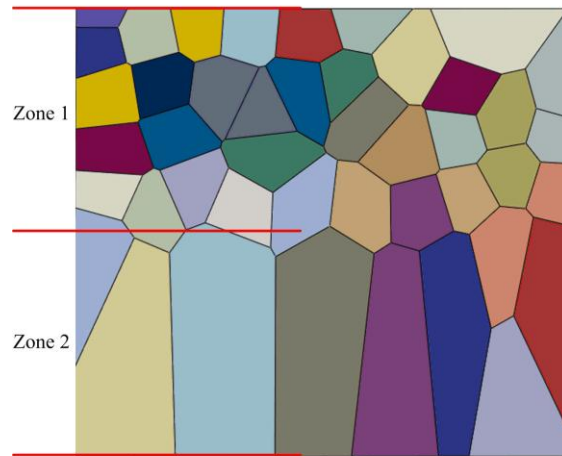
Label	Zone width (μm)	Grain size			Implementation parameters		
		D_{mean} (μm^2)	d_{mean}^\dagger (μm)	aspect ratio (ρ)	d_{reg} (μm)	α	δ (μm)
Zone 1	4	1.77	1.5	1	1.43	0.5	560
Zone 2‡	1	1.77	1.5	4	1.43	0.5	374

† Assuming a circular grain shape.

‡ Parameters were used to generate an equiaxed grain structure.



(a)



(b)

Figure 7.11. Illustration of a multi-zone grain structure, in which grains in different zones have different aspect ratio. (a) A grain structure of thermally grown oxide [207]; (b) a virtual gradient grain structure generated by the multi-zone CPVT model.

7.5 Summary

In many special situations, a grain structure may not be homogeneous along the given domain. Gradient grain structures can be found in a cold-worked metal or in a special fabricated functional material. Since representation of the grain structure is an essential step to perform micromechanics analysis for micro-forming simulations, generation of gradient virtual grain structure in agreement with micromechanics simulation requirements is critical.

In this chapter, the multi-zone CPVT model has been developed to fulfil this task, which is capable of generating grain structures statistically equivalent to a gradient grain structure. The multi-zone CPVT generates seeds in a domain containing joint multiple zones: within each individual zone, seeds are purely determined by the correlated physical parameters input to the CPVT model, and seeds near boundary regions are produced based on the proposed rule. The property of grains spanning two coupled zones was also addressed. Case studies have been presented that compare grain size gradients and grain morphological variations for both virtual grain structures, generated using the multi-zone CPVT model, and those obtained by other approaches, including true metallographic observations and numerical simulation of the Monte Carlo Potts model. These results showed high-quality agreements of grain size and morphology and verified the capabilities of the multi-zone CPVT for virtual gradient grain structure generation.

Chapter 8

Development of VGRAIN System

8.1 Introduction

A novel software system, VGRAIN, has been developed as a powerful finite element pre-processing tool, to support a variety of large-scale CPFE analyses. The system employs the proposed CPVT models to fully automate the generation of virtual grain structures for versatile applications of micromechanics studies, especially CPFE simulations. In addition to the generation of grain boundaries for a virtual grain structure, VGRAIN is capable of producing an entire materials model, including assignment of crystallographic orientations and material properties. The final structure can be directly imported into the commercial FE computational platforms, e.g. ABAQUS, *via* script-file input.

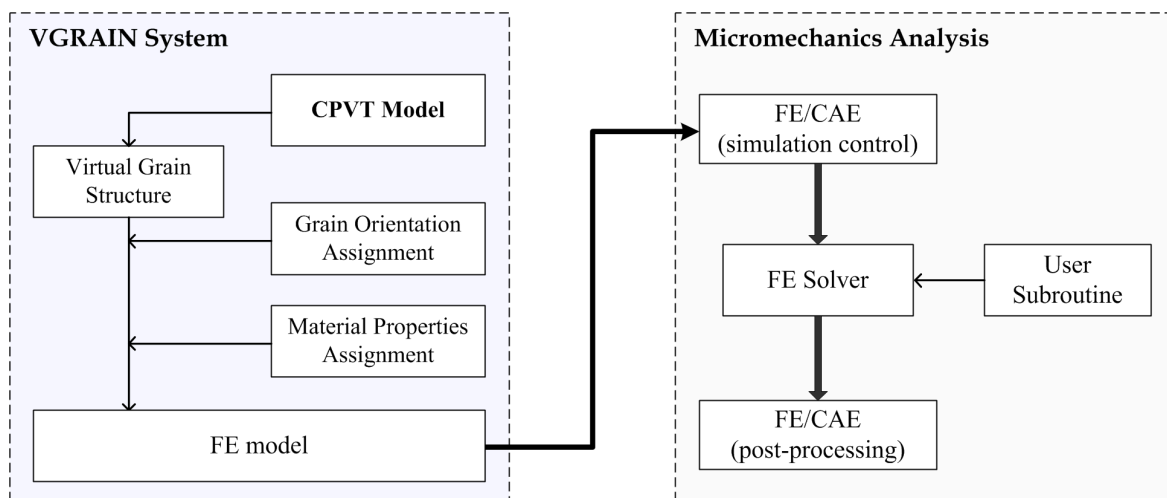


Figure 8.1. Illustration of a micromechanics simulation process employing the VGRAIN system.

Figure 8.1 presents the procedures of a micromechanics simulation using the pre-processing tool of VGRAIN. In VGRAIN software, a virtual grain structure, crystallographic orientations, and

material properties can be sequentially produced depending upon interactive user inputs *via* a collection of comprehensive and well-organised graphic user interfaces. A grain structure model presented in the GRAIN system can be exported into commercial FE/CAE platform, and subsequent FE simulations can be performed based on the imported FE model. The following section is dedicated to explanation of the organisation and functionality of the VGRAIN software system, and presents a practical example showing the use of VGRAIN to generate a virtual grain structure and study the grain structure's properties.

8.2 System development

8.2.1 System workflow

The VGRAIN system is an interactive windows application, which allows users to specify parameters or send commands *via* a collection of graphic user interfaces (GUIs) in a related context. There are four main tasks for a pre-processing system to generate a grain structure model including generating grain structure, specifying crystallographic orientations, assigning mechanical constants and exporting models to FE/CAE. All four of these tasks have been accomplished by four corresponding modules in the VGRAIN software. In addition, virtual grain structures can be produced in accordance with a set of properties. Therefore, an extra module is provided to study a variety of properties of a yielded virtual grain, e.g. grain morphology, grain size distributions, and crystallographic texture. The five modules and a general computational routine are presented in the flow chart of Figure 8.2.

The primary module (Module I in Figure 8.2) is the virtual grain structure generation module, which implements all the proposed grain structure models in this thesis including the 2D-CPVT model, the multi-zone 2D-CPVT model, the 3D-CPVT model, the multi-zone 3D-CPVT model and the cohesive zone model. Definition and generation of an initial grain structure is the first and critical step. Specification of parameters, e.g. the physical parameters and workpiece-related parameters, for any type of grain structures can be handled *via* a related GUI. With the configuration of a CPVT model, corresponding virtual grain structures can be repeatedly generated for further processing.

The crystallographic orientation definition module is provided to generate crystallographic orientations for grains in a given grain structure. The main GUI is a wizard-based dialogue for users to select and configure a generator: the uniform distribution random generator, the normal distribution random generator or the input of a constant orientation. In addition, a form dialogue is also provided to manually input or modify individual orientations. The third module is provided to allow users to assign mechanical properties of grains, which involves a collection of pre-defined parameter sets for different materials. Users are allowed to define their own material constants and state variables for FE analyses.

After generation of a grain structure and definition of its properties, the yielded structure is ready for build a FE model in a commercial FE/CAE environment. Importing a virtual grain structure from VGRAIN to a commercial FE/CAE platform is performed *via* text-based scripts. In current version, VGRAIN supports two platforms: ABAQUS and DEFORM. Note that there are many occasions where properties such as grain size distribution and crystallographic texture need to be provided. Module IV in Figure 8.2 incorporates a group of numerical procedures for users to analyse the characteristics of a grain structure, verifying the agreement of both the CPFE simulation requirements and the yielded structure. The main functionalities provided in the VGRAIN system include:

- Grain size distribution analysis, providing the histograms comparing distributions of the current generated grain structure and the user specified.
- Grain orientation analysis, providing the pole figures of the grain orientations assigned to the current grain structure.
- Grain morphological analysis, giving statistics on the number of edges and vertices, and other topological relations.
- Raw data spreadsheets, presenting the raw data on all the information of grains.

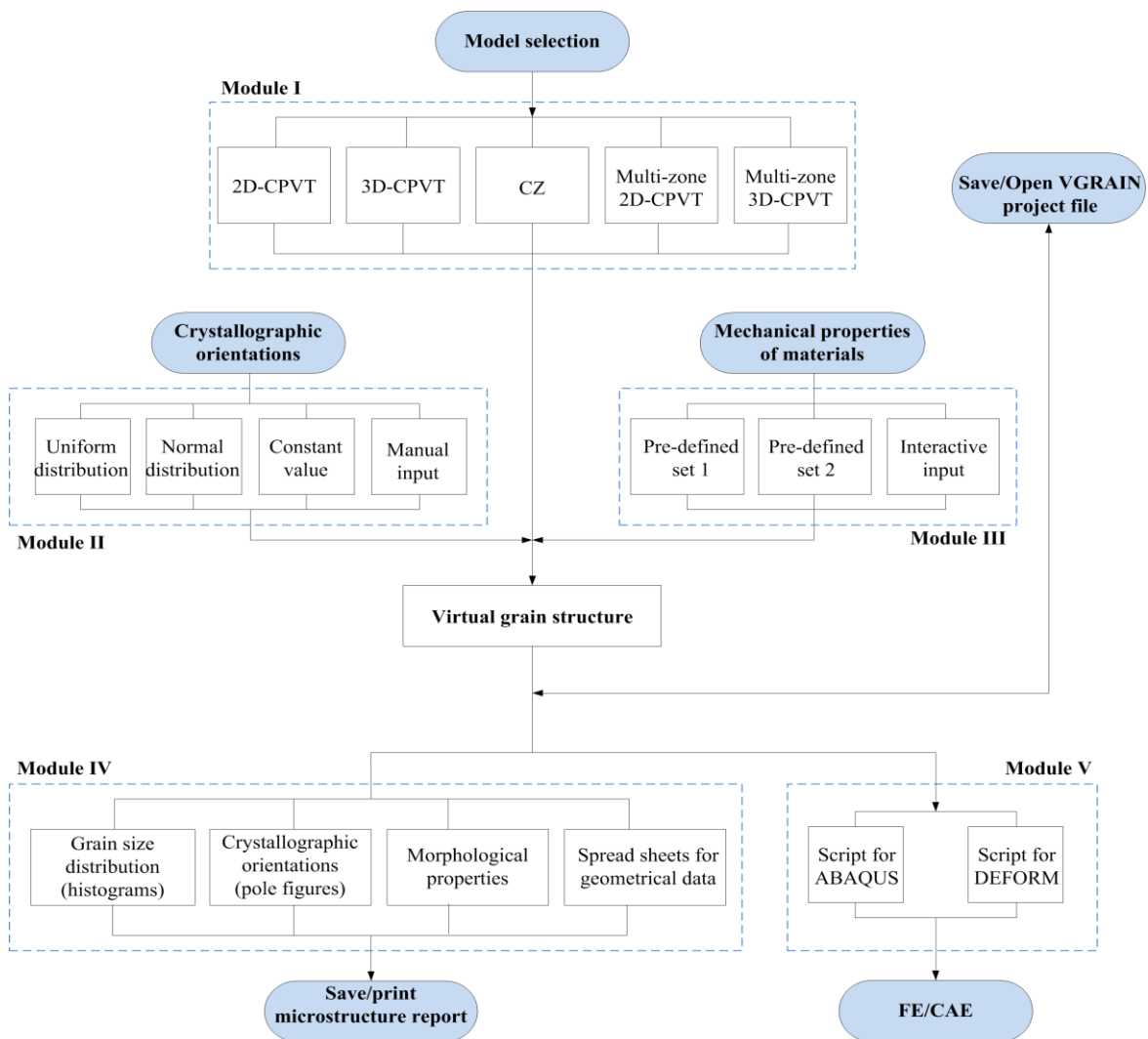


Figure 8.2. Flow chart of the VGRAIN system, showing the relations of the modules and their corresponding functionalities.

8.2.2 User interfaces

The graphical interface for the VGRAIN software is designed with considerations of providing user-friendly dialogues to define the CPVT model parameters, specify a variety of properties, export a well-built model and to also present sufficient information, including graphic visualisation and spreadsheets, to analyse a yielded structure. The layout of the main application window is presented in Figure 8.3. The right-hand view is a read-only region that visualises the information relating to the current virtual grain structure. In addition, the left-hand view also provides the entities for user to request the information of the current model, which is to then be displayed in the right-hand view with plots or spreadsheets. The major queries include plots of the current grain structure, reports of statistics of grains, and directly access of grain morphological and orientation data.

The menu bar and toolbar contains buttons relating to model and analysis operations. In the toolbar, there are mainly five groups of buttons to launch corresponding dialogues for user to specify related parameters. Group 1 contains the buttons to launch the dialogues for defining the CPVT models. A set of the buttons in group 2 perform save/open operations of a VGRAIN project file and the others are used to launch the dialogue to export the current virtual grain structure model for commercial FE platforms. Group 3 invokes the dialogues to specify mechanical constants for the current grain structure and group 4 focuses on the orientation generators to assign grain orientations for a presented grain structure. Group 5 corresponds to the definition of the cohesive zone model for a given grain structure, by which users can specify the parameters of a CZ model and perform junction partitioning. Different from the other groups, the buttons in Group 6 and 7 are shortcuts to query current grain structure information, which is displayed in the right-hand view. They are the plots of the geometrical grain structure, the coloured structure with grain orientations, the detailed grain structure, the grain size distribution histogram and the pole figures. Note that, the toolbar can be left to float over the GUI or attached to any border; in Figure 8.3 it was attached to the bottom in order to clearly present the graphic information.

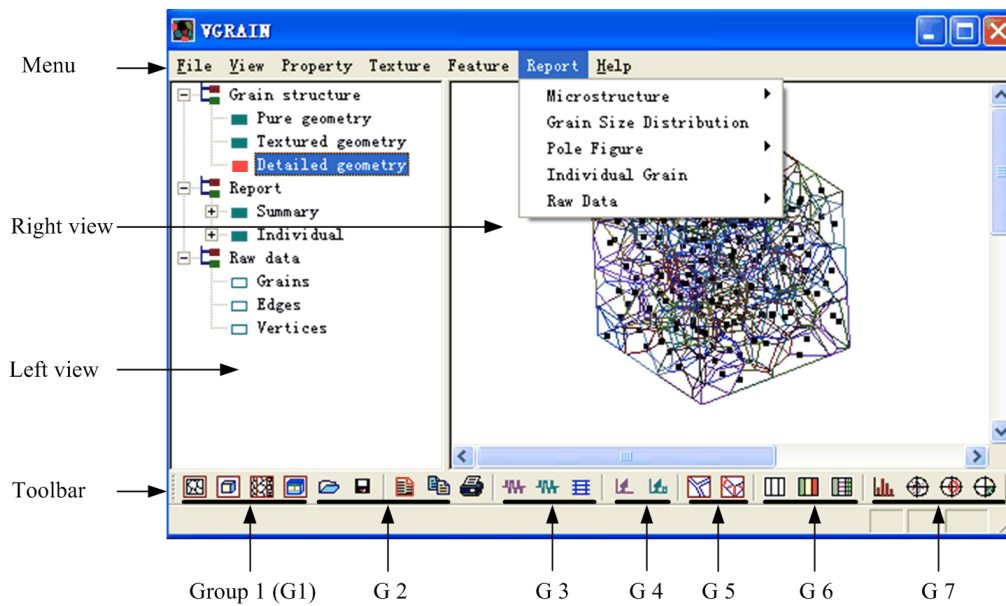


Figure 8.3. The application window of VGRAIN.

8.2.3 Code organisation

The system is written in C++ and three major classes, including the *CPVT class*, the *view class* and the *frame class*, coordinate the core workflow of the system. The CPVT class implements all the CPVT models for virtual grain structure generation. It receives the interactive user input *via* graphic dialogues and generates an entire grain structure model according to the invoked commands. Moreover, it also comprises a set of functions, interacting with external systems or devices: 1) Execution of the save and open operations for the current VGRAIN project, which contains the data of current grain structure model; 2) exportation of a grain structure model into a script file for a FE/CAE platform. Note that, the current version mainly focuses on the ABAQUS FE platform, where the format of a script file conforms to the specifications of ABAQUS journal file, and the DEFORM platform, where a script file is produced based on the specifications of DEFORM's key file. In the preceding chapters, all CPFE simulations were carried out using the integrated scheme of VGRAIN-ABAQUS. The view class maintains the left-hand view command flow and also updates the right-hand view displays, either plots or spreadsheets, by accessing the data of a present grain structure model. The commands received from the left-hand view originate from user context commands *via* a keyboard or mouse input. In contrast, the frame

class only executes the user commands invoked from the menu or the buttons of the toolbar. The schematic command and data flows of the system are illustrated in Figure 8.4.

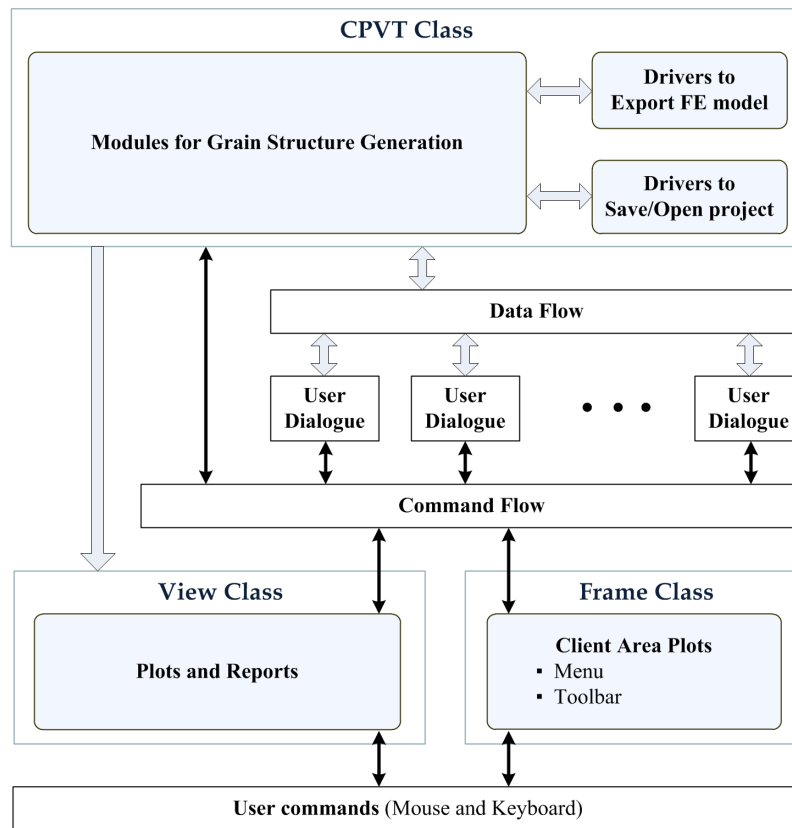


Figure 8.4. Illustration of the internal data flows and command flows.

8.3 Usage example and interface explanation

As discussed before, the VGRAIN system is capable of fulfilling the tasks of generating a virtual grain structure, analysing a generated grain structure, and export to the FE/CAE platform for FE simulations. To explore the functionalities and implementation procedures of the VGRAIN system, a case of a FE model of 2D multi-zone virtual grain structure with cohesive zones was built.

Figure 8.5 presents the procedure of definition and generation of a three-zone grain structure. Figure 8.5 (a) is the GUI to define the domain parameters and physical parameters for the 2D multi-zone CPVT model. According to the user inputs, a three-zone structural layout was defined and related physical parameters were specified. Note that, the symbol AR means aspect ratio of grains in a related zone. The right region in the GUI shows the computational results of the internal parameters such as N_{seed} , d_{reg} , and α . Figure 8.5 (b) displays the domain and grain boundaries of a three-zone virtual grain structure.

Figure 8.6 presents the definition of grain orientations and mechanical properties of the grains. The user interfaces in Figure 8.6 (a) are used to define orientation generators and specify mechanical constants of grains. There are three types of generators to produce the local coordinates of grains, including a normal distribution random number generator, a uniform distribution random number generator and a constant number assigning procedure. In this case, a uniform generator was used to generate angles for the transformation of local coordinates. This model was assumed to be a single phase material, and hence, all grains had identical mechanical properties. As a result, Figure 8.6 (c) presents the final polycrystalline material model with complete geometry, grain orientations, and material properties. Note that, the colours of grains denote the grain orientations. Furthermore, the cohesive zones were produced using the cohesive zone procedure in the grain structure generation module with a user-specified initial thickness. Figure 8.7 shows the resultant cohesive zone geometry, where the bottom-left box highlights the junction partitioning results for the selected region. After presenting the cohesive zones, properties of the cohesive zones, for example, a set of traction-separation parameters, were specified *via* the related GUI.

The VGRAIN system interacts with the commercial FE codes through various script files, which are based on the corresponding file formats. Figure 8.8 presents the way to export the generated grain structure into ABAQUS/CAE. In this example, the grain structure including cohesive zones with their junction partitioning meshes, grain orientations and material properties were all exported into ABAQUS/CAE, represented as a CPFE model. With the presentation of partitioned junctions, ABAQUS/CAE can easily perform the meshing operation. In addition, to generate virtual grain structure models VGRAIN also provide a set of useful functions to analyse the characteristics of a generated grain structure. As illustrated in Figure 8.9, users are

allowed to access the original data including grain morphological properties and orientation information, and are also able to achieve the grain size and orientation distribution features. These functions allow users to determine in advance whether the current grain structure model is satisfactory for their simulation requirements.

Set region number: 3

Workpiece Parameters

No.	Xmin_i	Xmax_i	Ymin_i	Ymax_i
1	0	80	0	160
2	80	120	0	160
3	120	200	0	160

Physical Parameters

No.	Dmean	Dsmall	Dlarge	Percentage	AR
1	300	200	400	0.85	1.0
2	50	35	65	0.80	1.0
3	300	160	440	0.65	1.0

Internal Parameters

No.	Dmean	Dreg	c	alpha	delta
1	300.00	18.76	18.01	0.60	11.28
2	50.00	7.60	17.70	0.60	4.55
3	300.00	18.76	3.71	0.23	4.40

Summary:

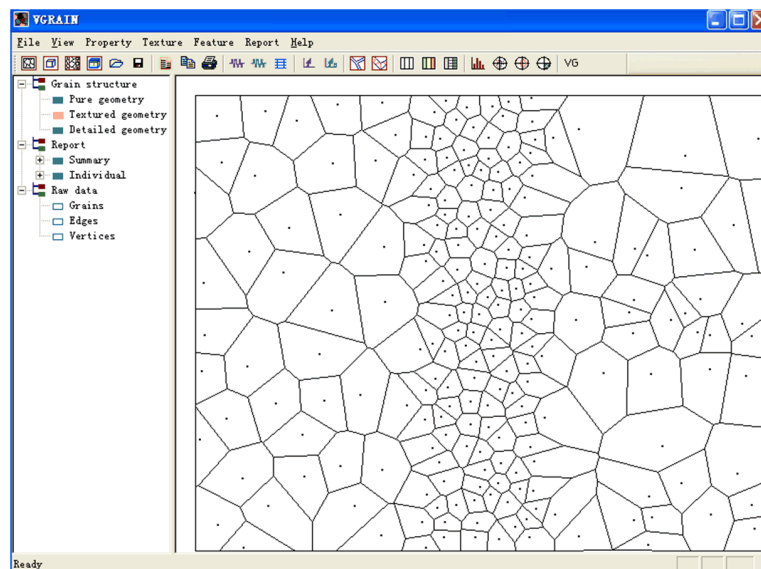
workpiece shape:

x_small 0.00 x_large 200.00
y_small 0.00 y_large 160.00
z_small N/A z_large N/A

Total region number: 3
Total grain number: 212

Calculate OK Cancel

(a)



(b)

Figure 8.5. User interface for definition of a 2D multi-zone CPVT model and the visualisation of a resultant virtual grain structure.

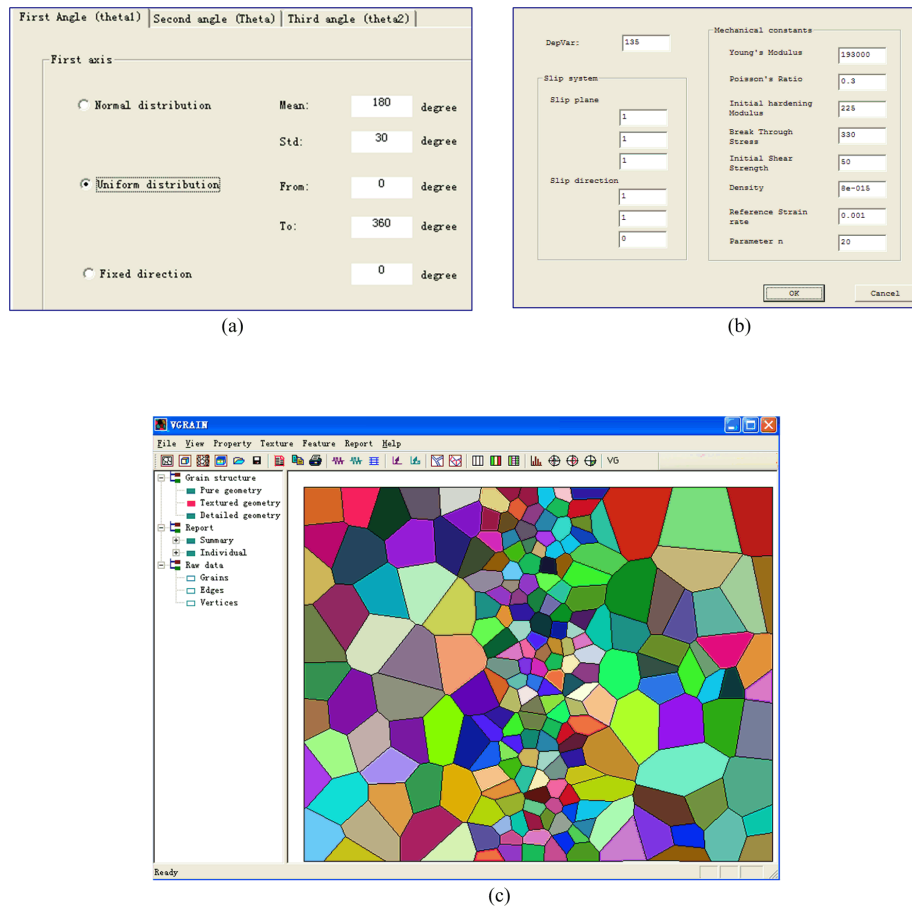


Figure 8.6. User interfaces for definition of a CPFE model based on a generated grain structure. (a) The crystallographic orientation generator; (b) assignment of material properties; (c) the resultant virtual grain structure, where colours relate to grain orientations.

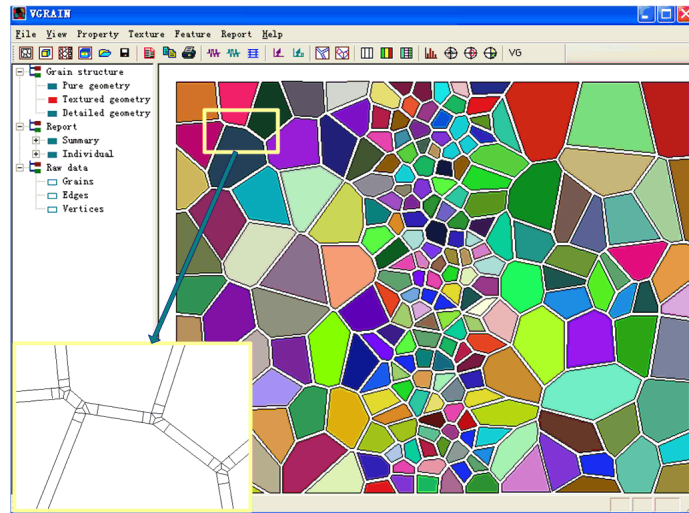


Figure 8.7. Illustration of the generation of cohesive zones and junction partitioning results.

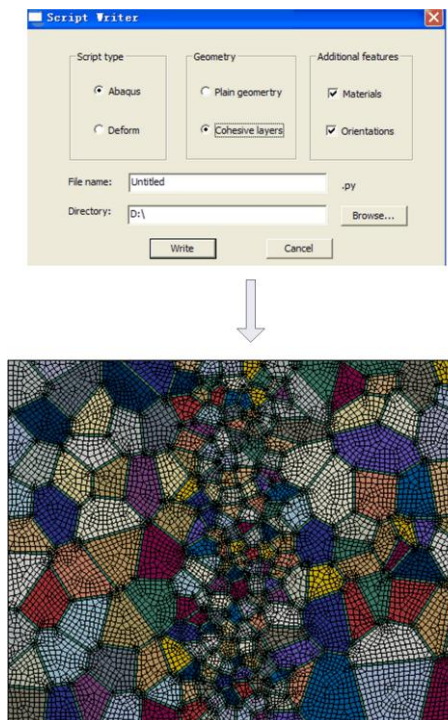
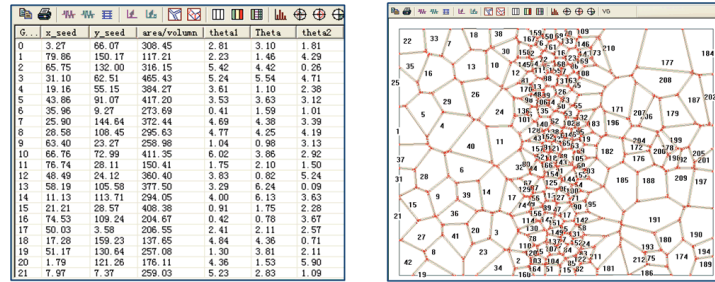
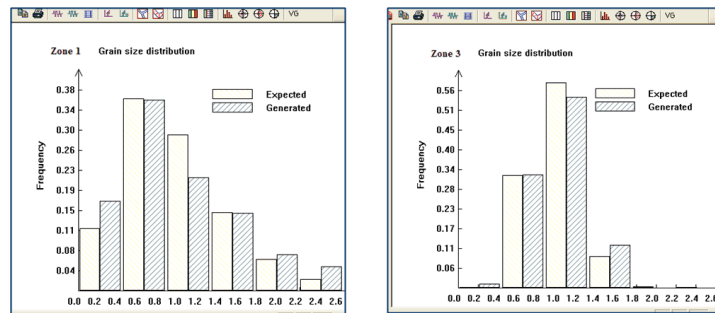


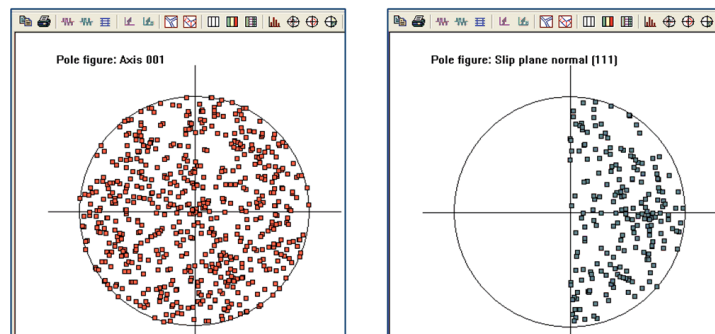
Figure 8.8. Illustration of the input of a CPFE model from VGRAIN to ABAQUS.



(a)



(b)



(c)

Figure 8.9. Illustration of the sub-modules in the VGRAIN system for analysing microstructure properties. (a) Accessing raw geometrical data; (b) comparing the grain size distributions between user-input physical parameters and the generated structures; (c) studying the crystallographic texture.

8.4 Summary

Representing a grain structure within a FE/CAE computational environment is essential for micromechanics simulations. The VGRAIN system has been developed to implement the virtual grain structure modelling methods proposed in this thesis. In summary, four major types of grain structures can be produced by the system, including two- and three-dimensional grain structures with grain size distribution control and two- and three-dimensional multi-zone (gradient) grain structures with independent grain size distribution control. A special cohesive zone modelling facility is provided to generate non-zero thickness cohesive zones and perform automatic junction partitioning for a virtual grain structure.

The design and architecture of the VGRAIN system was reviewed and the involved modules and functionalities were summarised. As a versatile FE pre-processing system, VGRAIN can provide a complete materials model with assignment of material properties and grain orientations, and it can also export a generated grain structure model into a commercial FE platform by text-based input files, for example, the journal file for ABAQUS. Moreover, as a stand-alone virtual grain structure generation system, VGRAIN has also been facilitated by a series of modules for analyses of grain morphological features and grain structure texture properties providing users with sufficient information on the generated structure. The usage example of a planar three-zone virtual grain structure has been presented to demonstrate the implementation details and the overall functions of the VGRAIN system for building a grain structure FE model.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Micromechanics simulations often require representation of a grain structure for FE simulation. There is an increasing demand for effectively generating high-fidelity virtual grain structures for accurate prediction of material deformation. Methods to construct virtual grain structures must be capable of generating realistic or statistically equivalent structures at any scale. That is, grain size distributions in the virtual structure must conform to physical observations, with the virtual grain structure generated within any specified area efficiently. In this work, a range of CPVT models have been developed to fulfil these requirements. The methodology of a CPVT model is summarised in the schematic diagram of Figure 9.1. It proceeds in four steps:

1. Determination of the grain size distribution (i.e., the distribution parameter c of a one-parameter gamma function), according to the physical parameters.
2. Derivation of the tessellation's regularity, α , from the mapping of $\alpha(c)$.
3. Calculation of the control parameter, δ , based on the regularity and the desired mean grain size.
4. Generation of virtual grain structures using the CPVT model with respect to the specified parameters.

The most important advantage of using CPVT models to generate a virtual grain structure is that the grain size distribution of a resultant grain structure is statistically equivalent to the user-specified physical parameters. Therefore, this method is an ideal FE simulation-oriented virtual grain structure method, providing realistic grain structure representation in terms of both grain morphology and grain size distribution.

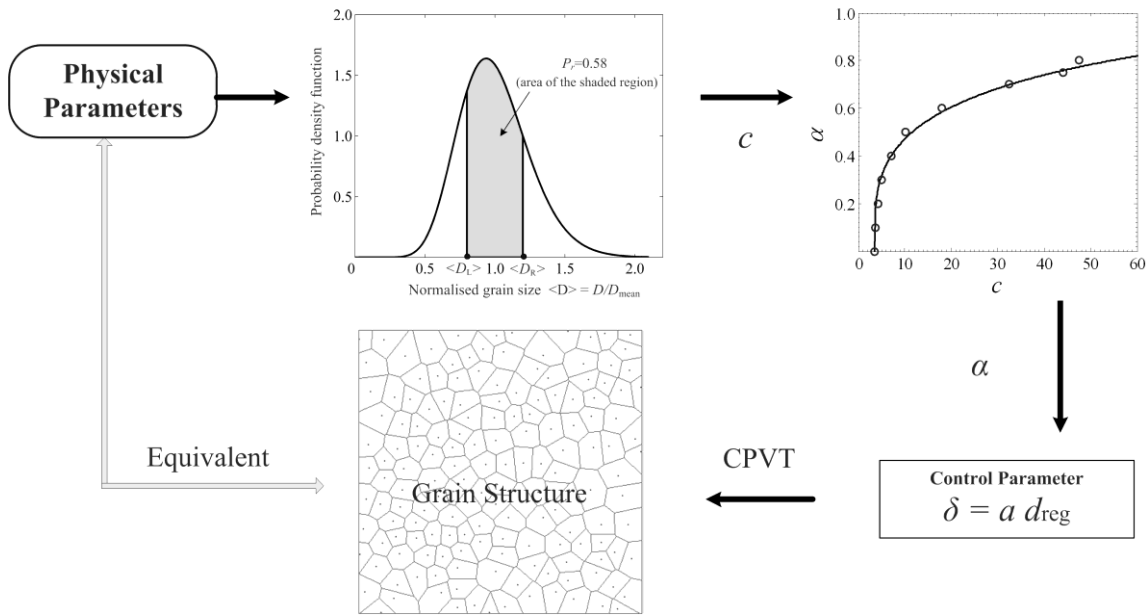


Figure 9.1. Schematic diagram showing the methodology of a CPVT model.

Development of a CPVT model requires three tasks to be accomplished: 1) Defining the mapping from the set of physical parameters to the distribution parameter, 2) establishing the mapping from the regularity to the distribution parameter of a one-parameter gamma distribution, 3) defining the regularity that specifies the uniformity of a tessellation, and deriving the control parameter based on the regularity. The first task concerns an essential component for all the CPVT models. In this work, a one-gamma distribution function was used to describe a grain size distribution characteristic and a group of four physical parameters were employed to represent the metallographic measurements of a grain size distribution property. Mathematical proofs of the uniqueness of the determination of the distribution parameter from the proposed set of physical parameters were studied, and an efficient numerical procedure was provided for computing the distribution parameter.

Tasks 2 and 3 correspond to the dimension of grain structure domain and the measurement of grain size. Thus, they were achieved in the context of generation of two- and three-dimensional grain structures, respectively. For the 2D-CPVT model, the grain size was measured

by grain area, and a descriptive model was proposed to establish the relationship between the regularity to the distribution parameter based on the statistical data studied in [154]. Furthermore, the regular hexagonal tessellation was taken as the reference of the most regular grain structure to define the regularity of a grain structure and derive the control parameter. For the 3D-CPVT model, the grain size was defined by grain volume, and a group of comprehensive statistical experiments have been performed to obtain the data pairs of the tessellation's regularity and corresponding grain size distribution parameter. The descriptive model used for the 2D-CPVT model was applied to the 3D-CPVT model with newly calibrated constants depending upon the statistical data. The BCC lattice was assumed to be the most regular three-dimensional grain structure, and hence was used to define the regular distance and the control parameter of the 3D-CPVT model. In addition to the development of the 2D- and 3D-CPVT models, CPFE simulations have also been carried out to demonstrate the typical applications and versatility for virtual grain structure modelling in micromechanics simulations.

Furthermore, a multi-zone scheme was proposed for the CPVT models to generate corresponding gradient grain structures, where grain size varies across the grain structure domain. The multi-zone CPVT model involves a novel mechanism of controlling the seed generation for grains spanning different zones in conjunction with the CPVT model that controls the seed generating process within individual zones. The multi-zone CPVT model can be applied to generate virtual grain structure with a continuous grain size gradient or consisting of a number of grain aggregates. Its merits are the flexibility of producing a large variety of gradient grain structure patterns and the natural morphological representation of interfacial grains between adjacent zones. Both of the 2D- and 3D-CPVT models are capable of generating a virtual grain structure with mean grain size gradient for the overall domain and grain size distribution control for individual zones.

In addition, to facilitate the applications of the 2D-CPVT model to incorporate cohesive interfaces for CPFE analyses of inter-granular crack initiation and evolution, a grain boundary offsetting algorithm was proposed to efficiently produce the cohesive boundaries for a virtual grain structure. In the CPFE framework a cohesive zone model requires its grain boundaries to be represented by quadrilateral elements. The most challenging issue in the current application of the cohesive zone models in micromechanics simulations is being able to automatically mesh

multiple junctions with quadrilateral elements. In this study, a novel junction partitioning scheme was proposed, which has the advantage of partitioning junctions with uniform quadrilateral elements and naturally assigning the normal and tangential directions to the junction elements. A rule-based method was presented to perform the automatically partitioning CZ junctions, including data representation, edge event processing, cut and trim operations. In order to demonstrate the novelty of the proposed cohesive zone modelling and junction partitioning schemes, the CPFE simulations of plane strain uniaxial tension and three point bending have been studied.

All the above virtual grain structure models have been implemented in the VGRAIN pre-processing software system. This system contributes to the current research on large-scale CPFE simulations, especially the field of micro-forming CPFE analyses. *Via* user-friendly interfaces and the well-organised functional modules, a virtual grain structure can be generated in a very large-scale (i.e. containing the large number of grains) with desired properties of grain morphology and grain size. Crystallographic orientations and mechanical constants of grains can be specified by user inputs based on a generated grain structure. A set of additional functions has also been developed for users to study a generated grain structure and verify the feasibility of the generated case for their simulation requirements. A well-built grain structure model in VGRAIN can be easily exported into the commercial FE/CAE platform, for example ABAQUS and DEFORM, *via* script input, whereby the VGRAIN system is seamlessly integrated into CPFE modelling and simulation.

9.2 Recommendations for future work

Representation of a virtual grain structure is an essential step to perform micromechanics simulations. Morphology and size distribution of a grain structure model must be realistic, and most importantly, it must be oriented to the requirements of a particular study. On many occasions, a grain structure model cannot exactly represent an engineering material, for example, in tailoring or optimising a new material. Therefore, the application-oriented virtual grain structure modelling techniques are of at least the same importance with a variety of models on simulating grain structure evolution such as the Monte Carlo models and the phase field models.

In this work, a comprehensive study was carried out to present a set of application-oriented virtual grain structure modelling tools. All the methods are based on the standard Voronoi tessellation representation, which assumes a simultaneous seed lattice and an isotropic grain growth. As presented in literature, there are a large number of tessellation models depicting a wide variety of solidification and grain growth processes. The methodology proposed in this work can be potentially extended to the other tessellation models, where a statistical relationship should be established between the grain size distribution parameter with a dominant quantity of the involved tessellation model such as the seed nucleation time interval and the ratio of grain growth velocity.

The novel cohesive zone representation and junction partitioning scheme has resolved the challenging problem of applying the Voronoi tessellation in simulating grain boundary sliding and separating behaviours. The demonstrated CPFE studies have shown the novelty and merits of this approach. However, this scheme is currently only implemented for a two-dimensional grain structure and current work is underway to extend this scheme to handle three-dimensional representation and meshes.

The multi-zone CPVT model is a powerful companion for versatile engineering applications. A few researches have been working to apply this model in a variety of micro-mechanics simulations in real engineering setups, such as simulating the mechanical properties of thermal barrier coatings and weld materials. Moreover, the CPVT models developed in this work concentrate on a single-phase material. An extension of this type of modelling to represent a multi-phase material will be extremely useful.

Appendix A.

Scripts for ABAQUS

A.1 2D grain structure

A.1.1 Script file

```
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
mdb.models['Model-1'].Sketch(name='__profile__', sheetSize = 1000.000 )
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.000, 0.000),point2=(10.000, 10.000))
mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='twoD', type=DEFORMABLE_BODY)
mdb.models['Model-1'].parts['twoD'].BaseShell(sketch=mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']
mdb.models['Model-1'].Sketch(gridSpacing=0.7, name='__profile__', sheetSize = 1000.000 , transform=mdb.models['Model-1'].parts['twoD'].MakeSketchTransform(sketchPlane=mdb.models['Model-1'].parts['twoD'].faces[0], sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(0.0, 0.0, 0.0)))
```

```

mdb.models['Model-1'].parts['twoD'].projectReferencesOntoSketch(filter=COPLANAR_EDGES,
sketch=mdb.models['Model-1'].sketches['__profile__'])

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.420 , 5.832 ), point2=( 10.000 ,
4.056 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.420 , 5.832 ), point2=( 6.081 ,
10.000 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 0.000 , 6.659 ), point2=( 4.365 ,
4.719 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 2.339 , 0.000 ), point2=( 4.365 ,
4.719 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.365 , 4.719 ), point2=( 6.420 ,
5.832 ) );

mdb.models['Model-1'].parts['twoD'].PartitionFaceBySketch(faces=mdb.models['Model-
1'].parts['twoD'].faces[0:1], sketch=mdb.models['Model-1'].sketches['__profile__'])

mdb.models['Model-1'].Material(name='M5-290deg')
mdb.models['Model-1'].materials['M5-290deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M5-290deg'].Density(table=((8.000000e-015,,))
mdb.models['Model-1'].materials['M5-290deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.352,
0.902, -0.251, 0.0, 0.0, 0.0, 1.0, 0.0, -0.601, 0.012, -0.799,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M5-290deg', name='Section-M5-290', thickness=1.0)

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt(( (6.219, 1.396, 0.000), ), )),
sectionName='Section-M5-290')

```

```

mdb.models['Model-1'].Material(name='M2-741deg')
mdb.models['Model-1'].materials['M2-741deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M2-741deg'].Density(table=((8.000000e-015,)),)
mdb.models['Model-1'].materials['M2-741deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.571,
0.727, -0.380, 0.0, 0.0, 0.0, 1.0, 0.0, 0.760, -0.644, -0.090,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M2-741deg', name='Section-M2-741', thickness=1.0)

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt(( (9.830, 8.676, 0.000), ), )),
sectionName='Section-M2-741')

```

```

mdb.models['Model-1'].Material(name='M1-525deg')
mdb.models['Model-1'].materials['M1-525deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M1-525deg'].Density(table=((8.000000e-015,)),)

```

```

mdb.models['Model-1'].materials['M1-525deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(

```

```

193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, -0.016,
    -0.153, -0.988, 0.0, 0.0, 0.0, 1.0, 0.0, 0.195, -0.970, 0.147,
    0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M1-525deg', name='Section-M1-
525', thickness=1.0)

```

```

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt((2.596, 8.088, 0.000), ), ),
sectionName='Section-M1-525')

```

```

mdb.models['Model-1'].Material(name='M1-881deg')

```

```

mdb.models['Model-1'].materials['M1-881deg'].Depvar(n=135)

```

```

mdb.models['Model-1'].materials['M1-881deg'].Density(table=((8.000000e-015,)),)

```

```

mdb.models['Model-1'].materials['M1-881deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(

```

```

193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, -0.370,

```



```

-0.354, 0.859, 0.0, 0.0, 0.0, 1.0, 0.0, -0.910, -0.046, -0.411,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M1-881deg', name='Section-M1-881', thickness=1.0)

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt((0.679, 3.775, 0.000), ), ),
sectionName='Section-M1-881')

```

A.1.2 Model in ABAQUS/CAE

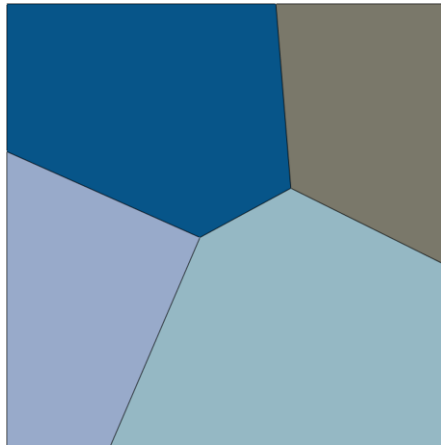


Figure A.1. A 2D grain structure consisting of four grains represented in ABAQUS.

A.2 2D cohesive zone model with junction partitioning

A.2.1 Script file

```
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *

mdb.models['Model-1'].Sketch(name='__profile__', sheetSize = 300.000 )
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0.000, 0.000),point2=(10.000,
10.000))

mdb.models['Model-1'].Part(dimensionality=TWO_D_PLANAR, name='twoD',
type=DEFORMABLE_BODY)

mdb.models['Model-1'].parts['twoD'].BaseShell(sketch=mdb.models['Model-
1'].sketches['__profile__'])

del mdb.models['Model-1'].sketches['__profile__']

mdb.models['Model-1'].Sketch(gridSpacing=0.7, name='__profile__', sheetSize = 300.000 ,
transform=mdb.models['Model-
1'].parts['twoD'].MakeSketchTransform(sketchPlane=mdb.models['Model-
1'].parts['twoD'].faces[0], sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=(0.0, 0.0,
0.0)))

mdb.models['Model-1'].parts['twoD'].projectReferencesOntoSketch(filter=COPLANAR_EDGES,
sketch=mdb.models['Model-1'].sketches['__profile__'])

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.421 ,
5.775 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.468 , 5.864 ), point2=( 6.451 ,
5.760 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.555 , 5.820 ), point2=( 6.511 ,
5.731 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 9.970 , 4.127 ), point2=( 9.926 ,
4.037 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 10.000 , 4.112 ), point2=( 10.000 ,
4.000 ) );
```

```

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.468 ,
5.864 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.421 , 5.775 ), point2=( 6.451 ,
5.760 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.468 , 5.864 ), point2=( 6.555 ,
5.820 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.451 , 5.760 ), point2=( 6.511 ,
5.731 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.555 , 5.820 ), point2=( 9.970 ,
4.127 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.511 , 5.731 ), point2=( 9.926 ,
4.037 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 9.970 , 4.127 ), point2=( 10.000 ,
4.112 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 9.926 , 4.037 ), point2=( 10.000 ,
4.000 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.468 ,
5.864 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.368 , 5.860 ), point2=( 6.465 ,
5.897 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.360 , 5.955 ), point2=( 6.460 ,
5.963 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.034 , 9.959 ), point2=( 6.134 ,
9.967 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.031 , 10.000 ), point2=( 6.131 ,
10.000 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.368 ,
5.860 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.468 , 5.864 ), point2=( 6.465 ,
5.897 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.368 , 5.860 ), point2=( 6.360 ,
5.955 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.465 , 5.897 ), point2=( 6.460 ,
5.963 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.360 , 5.955 ), point2=( 6.034 ,
9.959 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.460 , 5.963 ), point2=( 6.134 ,
9.967 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.034 , 9.959 ), point2=( 6.031 ,
10.000 ) );

```

```

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.134 , 9.967 ), point2=( 6.131 ,
10.000 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.356 , 4.717 ), point2=( 4.363 ,
4.775 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.300 , 4.693 ), point2=( 4.333 ,
4.788 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.231 , 4.724 ), point2=( 4.272 ,
4.815 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 0.030 , 6.591 ), point2=( 0.071 ,
6.682 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 0.000 , 6.604 ), point2=( 0.000 ,
6.714 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.356 , 4.717 ), point2=( 4.300 ,
4.693 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.363 , 4.775 ), point2=( 4.333 ,
4.788 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.300 , 4.693 ), point2=( 4.231 ,
4.724 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.333 , 4.788 ), point2=( 4.272 ,
4.815 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.231 , 4.724 ), point2=( 0.030 ,
6.591 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.272 , 4.815 ), point2=( 0.071 ,
6.682 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 0.030 , 6.591 ), point2=( 0.000 ,
6.604 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 0.071 , 6.682 ), point2=( 0.000 ,
6.714 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.356 , 4.717 ), point2=( 4.404 ,
4.683 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.300 , 4.693 ), point2=( 4.391 ,
4.653 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.273 , 4.631 ), point2=( 4.365 ,
4.591 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 2.314 , 0.070 ), point2=( 2.406 ,
0.031 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 2.284 , 0.000 ), point2=( 2.393 ,
0.000 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.356 , 4.717 ), point2=( 4.300 ,
4.693 ) );

```

```

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.404 , 4.683 ), point2=( 4.391 ,
4.653 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.300 , 4.693 ), point2=( 4.273 ,
4.631 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.391 , 4.653 ), point2=( 4.365 ,
4.591 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.273 , 4.631 ), point2=( 2.314 ,
0.070 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.365 , 4.591 ), point2=( 2.406 ,
0.031 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 2.314 , 0.070 ), point2=( 2.284 ,
0.000 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 2.406 , 0.031 ), point2=( 2.393 ,
0.000 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.421 ,
5.775 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.368 , 5.860 ), point2=( 6.392 ,
5.759 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.286 , 5.816 ), point2=( 6.333 ,
5.728 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.451 , 4.822 ), point2=( 4.499 ,
4.734 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.392 , 4.790 ), point2=( 4.404 ,
4.683 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.363 , 4.775 ), point2=( 4.356 ,
4.717 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.419 , 5.833 ), point2=( 6.368 ,
5.860 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.421 , 5.775 ), point2=( 6.392 ,
5.759 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.368 , 5.860 ), point2=( 6.286 ,
5.816 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.392 , 5.759 ), point2=( 6.333 ,
5.728 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.286 , 5.816 ), point2=( 4.451 ,
4.822 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 6.333 , 5.728 ), point2=( 4.499 ,
4.734 ) );
mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.451 , 4.822 ), point2=( 4.392 ,
4.790 ) );

```

```

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.499 , 4.734 ), point2=( 4.404 ,
4.683 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.392 , 4.790 ), point2=( 4.363 ,
4.775 ) );

mdb.models['Model-1'].sketches['__profile__'].Line(point1=( 4.404 , 4.683 ), point2=( 4.356 ,
4.717 ) );

mdb.models['Model-1'].parts['twoD'].PartitionFaceBySketch(faces=mdb.models['Model-
1'].parts['twoD'].faces[0:1], sketch=mdb.models['Model-1'].sketches['__profile__'])

mdb.models['Model-1'].Material(name='M5-290deg')
mdb.models['Model-1'].materials['M5-290deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M5-290deg'].Density(table=((8.000000e-015,)),)
mdb.models['Model-1'].materials['M5-290deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.352,
0.902, -0.251, 0.0, 0.0, 0.0, 1.0, 0.0, -0.601, 0.012, -0.799,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

mdb.models['Model-1'].HomogeneousSolidSection(material='M5-290deg', name='Section-M5-
290', thickness=1.0)

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt(( (6.219, 1.396, 0.000), ), )),
sectionName='Section-M5-290')

```

```

mdb.models['Model-1'].Material(name='M2-741deg')
mdb.models['Model-1'].materials['M2-741deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M2-741deg'].Density(table=((8.000000e-015,)),)
mdb.models['Model-1'].materials['M2-741deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.571,
    0.727, -0.380, 0.0, 0.0, 0.0, 1.0, 0.0, 0.760, -0.644, -0.090,
    0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

mdb.models['Model-1'].HomogeneousSolidSection(material='M2-741deg', name='Section-M2-
741', thickness=1.0)
mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt(( (9.830, 8.676, 0.000), ), )),
sectionName='Section-M2-741')

mdb.models['Model-1'].Material(name='M1-525deg')
mdb.models['Model-1'].materials['M1-525deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M1-525deg'].Density(table=((8.000000e-015,)),)
mdb.models['Model-1'].materials['M1-525deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
    0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,

```

```

0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, -0.016,
-0.153, -0.988, 0.0, 0.0, 0.0, 1.0, 0.0, 0.195, -0.970, 0.147,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M1-525deg', name='Section-M1-525', thickness=1.0)

```

```

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt((2.596, 8.088, 0.000), ), ),
sectionName='Section-M1-525')

```

```

mdb.models['Model-1'].Material(name='M1-881deg')

```

```

mdb.models['Model-1'].materials['M1-881deg'].Depvar(n=135)

```

```

mdb.models['Model-1'].materials['M1-881deg'].Density(table=((8.000000e-015,)),)

```

```

mdb.models['Model-1'].materials['M1-881deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(

```

```

193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, -0.370,
-0.354, 0.859, 0.0, 0.0, 0.0, 1.0, 0.0, -0.910, -0.046, -0.411,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,

```



```

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M1-881deg', name='Section-M1-881', thickness=1.0)

```

```

mdb.models['Model-1'].parts['twoD'].SectionAssignment(region=Region(faces=
mdb.models['Model-1'].parts['twoD'].faces.findAt((0.679, 3.775, 0.000), ), ),
sectionName='Section-M1-881')

```

```

mdb.models['Model-1'].Material(name='Cohesive')

```

```

mdb.models['Model-1'].materials['Cohesive'].Density(table=((8.000000e-015,)),)

```

```

mdb.models['Model-1'].materials['Cohesive'].Elastic(table=((210000.0, 210000.0, 210000.0)),,
type=TRACTION)

```

```

mdb.models['Model-1'].materials['Cohesive'].MaxsDamageInitiation(table=((260.0, 260.0,
260.0)),)

```

```

mdb.models['Model-1'].materials['Cohesive'].maxsDamageInitiation.DamageEvolution(table=((75.0,)),,type=ENERGY)

```

```

mdb.models['Model-1'].CohesiveSection(initialThickness=0.100, initialThicknessType=SPECIFY, material='Cohesive',
name='Cohesive', outOfPlaneThickness=None, response=TRACTION_SEPARATION)

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.420186, 5.804142, 0.)),)

```

```

myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.436084, 5.824029, 0.)),)

```

```

mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)

```

```

mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0], sense=REVERSE)

```

```

mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4, elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI, elemLibrary=EXPLICIT)), regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.436084, 5.824029, 0.)),), ))

```

```

mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0, offsetField="", offsetType=MIDDLE_SURFACE, region=
Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.436084, 5.824029, 0.)),), ),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.459543,5.812126 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.491483,5.814889,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.491483,5.814889,0.)),) ,))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.491483,5.814889,0.)),) ),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.533148,5.775621 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((7.678811,5.226024,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((7.678811,5.226024,0.)),) ,))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((7.678811,5.226024,0.)),) ),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((9.947922,4.082036 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((9.965281,4.092030,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT)

```

```
IT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((9.965281,4.092030,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((9.965281,4.092030,0.)),),),sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.443453,5.848415,0.)),)
```

```
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.418254,5.852282,0.)),)
```

```
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP, elemShape=QUAD)
```

```
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0], region=myRegion[0],sense=REVERSE)
```

```
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.418254,5.852282,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.418254,5.852282,0.)),),),sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.416516,5.878521,0.)),)
```

```
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.397709,5.904134,0.)),)
```

```
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP, elemShape=QUAD)
```

```
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0], region=myRegion[0],sense=REVERSE)
```

```
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.397709,5.904134,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.397709,5.904134,0.)),),),sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.409931,5.959417,0.)),)
```

```
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.176089,8.626934,0.)),)
```

```
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
```

```
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
```

```
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.176089,8.626934,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.176089,8.626934,0.)),),),
sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.084085,9.962720,0.)),)
```

```
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.065451,9.986221,0.)),)
```

```
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
```

```
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
```

```
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.065451,9.986221,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.065451,9.986221,0.)),),),
sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.359334,4.745831,0.)),)
```

```
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.339472,4.728319,0.)),)
```

```
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
```

```
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
```

```
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.339472,4.728319,0.)),),))
```

```
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.339472,4.728319,0.)),),),
sectionName='Cohesive')
```

```
n=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.339472,4.728319,0.)),),
sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.316143,4.740723 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.287764,4.735097,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.287764,4.735097,0.)),),))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.287764,4.735097,0.)),),),
sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.251312,4.769536 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.844362,5.376586,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.844362,5.376586,0.)),),))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.844362,5.376586,0.)),),),
sectionName='Cohesive')
```

```
myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((0.050767,6.636377 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((0.033845,6.625659,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP,
elemShape=QUAD)
```

```

mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)

mdb.models['Model-
1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLIC
IT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-
1'].parts['twoD'].faces.findAt(((0.033845,6.625659,0.)),),))

mdb.models['Model-
1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,regio
n=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((0.033845,6.625659,0.)),),),
sectionName='Cohesive')


myEdge=mdb.models['Model-
1'].parts['twoD'].edges.getClosest(coordinates=((4.379963,4.700142,0.)),)

myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.353225,4.697859,0.)),)

mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)

mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)

mdb.models['Model-
1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLIC
IT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-
1'].parts['twoD'].faces.findAt(((4.353225,4.697859,0.)),),))

mdb.models['Model-
1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,regio
n=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.353225,4.697859,0.)),),),
sectionName='Cohesive')


myEdge=mdb.models['Model-
1'].parts['twoD'].edges.getClosest(coordinates=((4.345426,4.672951,0.)),)

myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.321255,4.658904,0.)),)

mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)

mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)

mdb.models['Model-
1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLIC
IT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-
1'].parts['twoD'].faces.findAt(((4.321255,4.658904,0.)),),))

mdb.models['Model-
1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,regio
n=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.321255,4.658904,0.)),),),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.318856,4.611080 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((3.650685,3.097416,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((3.650685,3.097416,0.)),) ,))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((3.650685,3.097416,0.)),) ),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((2.360286,0.050358 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.330554,0.023363,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.330554,0.023363,0.)),) ,))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((2.330554,0.023363,0.)),) ),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.420186,5.804142 ,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.402743,5.822767,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)

```

```

mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.402743,5.822767,0.)),),))

mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.402743,5.822767,0.)),),),sectionName='Cohesive')


myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.379944,5.809702,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.348562,5.811665,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP, elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0], region=myRegion[0],sense=REVERSE)

mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.348562,5.811665,0.)),),))

mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((6.348562,5.811665,0.)),),),sectionName='Cohesive')


myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((6.309601,5.771619,0.)),)
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((5.078428,5.124032,0.)),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion, technique=SWEEP, elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0], region=myRegion[0],sense=REVERSE)

mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT), ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((5.078428,5.124032,0.)),),))

mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField='',offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((5.078428,5.124032,0.)),),),sectionName='Cohesive')

```



```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.474744,4.778253 ,0.),))
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.447267,4.782330,0.),),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.447267,4.782330,0.),),),))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.447267,4.782330,0.),),),),
sectionName='Cohesive')

```

```

myEdge=mdb.models['Model-1'].parts['twoD'].edges.getClosest(coordinates=((4.398286,4.736860 ,0.),))
myRegion=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.370327,4.760715,0.),),)
mdb.models['Model-1'].parts['twoD'].setMeshControls(regions=myRegion,    technique=SWEEP,
elemShape=QUAD)
mdb.models['Model-1'].parts['twoD'].setSweepPath(edge=myEdge[0][0],
region=myRegion[0],sense=REVERSE)
mdb.models['Model-1'].parts['twoD'].setElementType(elemTypes=(ElemType(elemCode=COH2D4,elemLibrary=EXPLICIT),
ElemType(elemCode=UNKNOWN_TRI,elemLibrary=EXPLICIT)),regions=(mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.370327,4.760715,0.),),),))
mdb.models['Model-1'].parts['twoD'].SectionAssignment(offset=0.0,offsetField="",offsetType=MIDDLE_SURFACE,region=Region(faces=mdb.models['Model-1'].parts['twoD'].faces.findAt(((4.370327,4.760715,0.),),),),
sectionName='Cohesive')

```

A.2.2 Model in ABAQUS/CAE

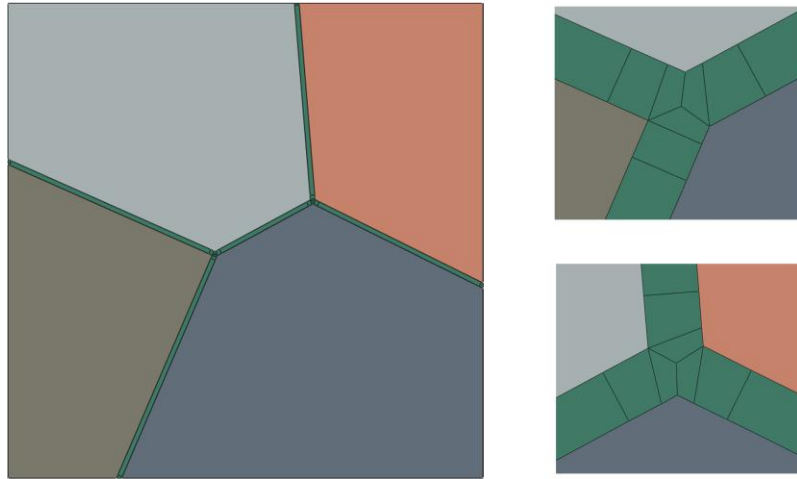


Figure A.2. A 2D grain structure with cohesive zones consisting of four grains represented in ABAQUS, where details of the two junctions are highlighted.

A.3 3D grain structure

A.3.1 Script file

```

from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=(100)+200)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0,0), point2=(10,10))
mdb.models['Model-1'].Part(dimensionality=THREE_D, name='Part-1', type=
DEFORMABLE_BODY)
vP=mdb.models['Model-1'].parts['Part-1']
vP.BaseSolidExtrude(depth=10, sketch= mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 5.3808455011, 0.0000000000, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.1369960401, 0.0000000000 ))

```

```

vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.0000000000 ))
vP.DatumPlaneByThreePoints( point1=vP.datums[2], point2=vP.datums[3], point3=vP.datums[4])
vP.DatumAxisByTwoPoint(point1=vP.datums[2], point2=vP.datums[3])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[6],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[6])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 0.0000000000, 0.0000000000, 7.6869293761 ))
vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPointByCoordinate(coords=( 5.3808455011, 0.0000000000, 0.0000000000 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[8],          point2=vP.datums[9],
point3=vP.datums[10])
vP.DatumAxisByTwoPoint(point1=vP.datums[8], point2=vP.datums[9])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[12],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[12])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 6.8788336041 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 0.0000000000, 7.6869293761 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[14],          point2=vP.datums[15],
point3=vP.datums[16])
vP.DatumAxisByTwoPoint(point1=vP.datums[14], point2=vP.datums[15])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[18],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,

```

```

    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[18])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.5285634866, 0.4158847691 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.1369960401, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 5.3808455011, 0.0000000000, 0.0000000000 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[20],          point2=vP.datums[21],
point3=vP.datums[22])
vP.DatumAxisByTwoPoint(point1=vP.datums[20], point2=vP.datums[21])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[24], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[24])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 0.0000000000, 0.0000000000, 7.6869293761 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 6.8788336041 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[26],          point2=vP.datums[27],
point3=vP.datums[28])
vP.DatumAxisByTwoPoint(point1=vP.datums[26], point2=vP.datums[27])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[30], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[30])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 6.8788336041 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[32],          point2=vP.datums[33],
point3=vP.datums[34])
vP.DatumAxisByTwoPoint(point1=vP.datums[32], point2=vP.datums[33])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1,

```

```

    sketchUpEdge=vP.datums[36],sketchOrientation=RIGHT, origin=(0,0,0))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[36])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.1369960401, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.5285634866, 0.4158847691 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[38],          point2=vP.datums[39],
point3=vP.datums[40])
vP.DatumAxisByTwoPoint(point1=vP.datums[38], point2=vP.datums[39])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[41], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[42],sketchOrientation=RIGHT, origin=(0,0,0))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[41], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[42])
del mdb.models['Model-1'].sketches['__profile__']

mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=(100)+200)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0,0), point2=(10,10))
mdb.models['Model-1'].Part(dimensionality=THREE_D,          name='Part-2',          type=
DEFORMABLE_BODY)
vP=mdb.models['Model-1'].parts['Part-2']
vP.BaseSolidExtrude(depth=10, sketch= mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.1369960401, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.5285634866, 0.4158847691 ))
vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPlaneByThreePoints( point1=vP.datums[2], point2=vP.datums[3], point3=vP.datums[4])
vP.DatumAxisByTwoPoint(point1=vP.datums[2], point2=vP.datums[3])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[6],sketchOrientation=RIGHT, origin=(0,0,0))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

```

```

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[6])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 0.0000000000, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.1369960401, 0.0000000000 ))
vP.DatumPointByCoordinate(coords=( 5.3808455011, 0.0000000000, 0.0000000000 ))
vP.DatumPlaneByThreePoints(
    point1=vP.datums[8], point2=vP.datums[9],
    point3=vP.datums[10])
vP.DatumAxisByTwoPoint(point1=vP.datums[8], point2=vP.datums[9])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[12], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES, sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[12])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 0.0000000000, 10.0000000000 ))
vP.DatumPlaneByThreePoints(
    point1=vP.datums[14], point2=vP.datums[15],
    point3=vP.datums[16])
vP.DatumAxisByTwoPoint(point1=vP.datums[14], point2=vP.datums[15])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[18], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES, sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[18])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 1.5704708872, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 8.5285634866, 0.4158847691 ))
vP.DatumPlaneByThreePoints(
    point1=vP.datums[20], point2=vP.datums[21],
    point3=vP.datums[22])
vP.DatumAxisByTwoPoint(point1=vP.datums[20], point2=vP.datums[21])

```

```

mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[24],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[24])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 1.5704708872, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[26],          point2=vP.datums[27],
point3=vP.datums[28])
vP.DatumAxisByTwoPoint(point1=vP.datums[26], point2=vP.datums[27])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[30],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[30])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 1.5704708872, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 0.0000000000, 10.0000000000 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[32],          point2=vP.datums[33],
point3=vP.datums[34])
vP.DatumAxisByTwoPoint(point1=vP.datums[32], point2=vP.datums[33])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[36],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[36])
del mdb.models['Model-1'].sketches['__profile__']

```

```

mdb.models['Model-1'].ConstrainedSketch(name='__profile__', sheetSize=(100)+200)
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(0,0), point2=(10,10))
mdb.models['Model-1'].Part(dimensionality=THREE_D, name='Part-3', type=
DEFORMABLE_BODY)
vP=mdb.models['Model-1'].parts['Part-3']
vP.BaseSolidExtrude(depth=10, sketch= mdb.models['Model-1'].sketches['__profile__'])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 6.8788336041 ))
vP.DatumPointByCoordinate(coords=( -0.0000000000, -0.0000000000, 7.6869293761 ))
vP.DatumPlaneByThreePoints( point1=vP.datums[2], point2=vP.datums[3], point3=vP.datums[4])
vP.DatumAxisByTwoPoint(point1=vP.datums[2], point2=vP.datums[3])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
transform=vP.MakeSketchTransform(sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1,
sketchUpEdge=vP.datums[6],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES, sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
sketchPlane=vP.datums[5], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[6])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 1.5704708872, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPlaneByThreePoints( point1=vP.datums[8], point2=vP.datums[9],
point3=vP.datums[10])
vP.DatumAxisByTwoPoint(point1=vP.datums[8], point2=vP.datums[9])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
transform=vP.MakeSketchTransform(sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1,
sketchUpEdge=vP.datums[12],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES, sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
sketchPlane=vP.datums[11], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[12])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 1.5704708872, 10.0000000000 ))

```



```

vP.DatumPlaneByThreePoints(          point1=vP.datums[14],          point2=vP.datums[15],
point3=vP.datums[16])
vP.DatumAxisByTwoPoint(point1=vP.datums[14], point2=vP.datums[15])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[18],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[17], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[18])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 10.0000000000 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[20],          point2=vP.datums[21],
point3=vP.datums[22])
vP.DatumAxisByTwoPoint(point1=vP.datums[20], point2=vP.datums[21])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[24],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[23], sketchPlaneSide=SIDE1,sketchUpEdge=vP.datums[24])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( -0.0000000000, -0.0000000000, 7.6869293761 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[26],          point2=vP.datums[27],
point3=vP.datums[28])
vP.DatumAxisByTwoPoint(point1=vP.datums[26], point2=vP.datums[27])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__',sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[30],sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,

```

```

    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[29], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[30])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 0.2969786065 ))
vP.DatumPointByCoordinate(coords=( 10.0000000000, 10.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 0.0000000000, 10.0000000000, 10.0000000000 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[32],          point2=vP.datums[33],
point3=vP.datums[34])
vP.DatumAxisByTwoPoint(point1=vP.datums[32], point2=vP.datums[33])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[36], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[35], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[36])
del mdb.models['Model-1'].sketches['__profile__']

vP.DatumPointByCoordinate(coords=( 6.9858961735, 0.0000000000, 10.0000000000 ))
vP.DatumPointByCoordinate(coords=( 1.9628295258, 0.0000000000, 6.3950234437 ))
vP.DatumPointByCoordinate(coords=( -0.0000000000, -0.0000000000, 7.6869293761 ))
vP.DatumPlaneByThreePoints(          point1=vP.datums[38],          point2=vP.datums[39],
point3=vP.datums[40])
vP.DatumAxisByTwoPoint(point1=vP.datums[38], point2=vP.datums[39])
mdb.models['Model-1'].ConstrainedSketch(gridSpacing=0.12, name='__profile__', sheetSize=5.02,
    transform=vP.MakeSketchTransform(sketchPlane=vP.datums[41], sketchPlaneSide=SIDE1,
    sketchUpEdge=vP.datums[42], sketchOrientation=RIGHT, origin=(0,0,0)))
mdb.models['Model-1'].sketches['__profile__'].sketchOptions.setValues(sheetSize=100)
vP.projectReferencesOntoSketch(filter=COPLANAR_EDGES,          sketch=mdb.models['Model-
1'].sketches['__profile__'])
mdb.models['Model-1'].sketches['__profile__'].rectangle(point1=(-100, -100), point2=(100, 100))

vP.CutExtrude(depth=100, flipExtrudeDirection=OFF,
    sketch=mdb.models['Model-1'].sketches['__profile__'], sketchOrientation=RIGHT,
    sketchPlane=vP.datums[41], sketchPlaneSide=SIDE1, sketchUpEdge=vP.datums[42])
del mdb.models['Model-1'].sketches['__profile__']

mdb.models['Model-1'].rootAssembly.DatumCsysByDefault(CARTESIAN)
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-1-1',
    part=mdb.models['Model-1'].parts['Part-1'])
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-2-1',
    part=mdb.models['Model-1'].parts['Part-2'])
mdb.models['Model-1'].rootAssembly.Instance(dependent=ON, name='Part-3-1',
    part=mdb.models['Model-1'].parts['Part-3'])

```



```

0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M5-138deg', name='Section-M5-138', thickness=1.0)
mdb.models['Model-1'].parts['Part-total'].SectionAssignment(region=Region(cells=mdb.models['Model-1'].parts['Part-total'].cells.findAt(((9.830544, 2.682826, 3.532354),))), sectionName='Section-M5-138')

```

```

mdb.models['Model-1'].Material(name='M1-529deg')
mdb.models['Model-1'].materials['M1-529deg'].Depvar(n=135)
mdb.models['Model-1'].materials['M1-529deg'].Density(table=((8.000000e-015,)),)
mdb.models['Model-1'].materials['M1-529deg'].UserMaterial(unsymm=ON,
mechanicalConstants=(
193000.000, 0.300, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, -0.247,
-0.175, -0.953, 0.0, 0.0, 0.0, 1.0, 0.0, 0.850, 0.433, -0.299,
0.0, 0.0, 20.000, 0.001, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 225.000, 330.000, 50.000, 0.0,
0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 1.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 1.0, 10.0, 1.0e-5, 0.0, 0.0, 0.0, 0.0, 0.0))

```

```

mdb.models['Model-1'].HomogeneousSolidSection(material='M1-529deg', name='Section-M1-529', thickness=1.0)
mdb.models['Model-1'].parts['Part-total'].SectionAssignment(region=Region(cells=mdb.models['Model-1'].parts['Part-total'].cells.findAt(((7.805622, 6.569127, 6.353819),))), sectionName='Section-M1-529')

```

A.3.2 Model in ABAQUS/CAE

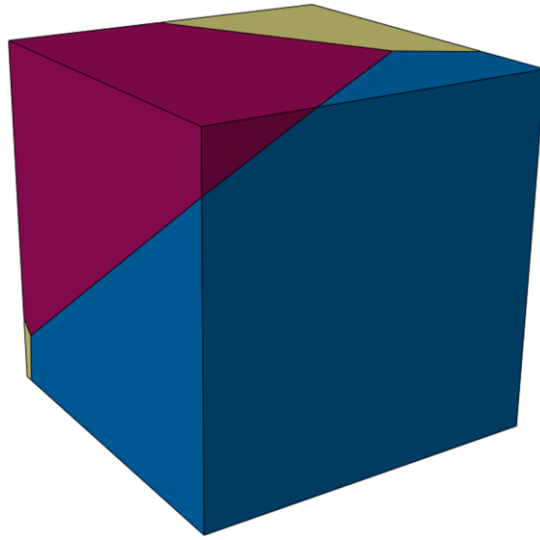


Figure A.3. A 3D grain structure consisting of three grains represented in ABAQUS.

References

- [1] Geiger M, Kleiner M, Eckstein R, Tiesler N, Engel U. Microforming. *CIRP Annals-Manufacturing Technology* 2001; **50**(2):445–462.
- [2] Vollertsen F, Hu Z, Niehoff H, Theiler C. State of the art in micro forming and investigations into micro deep drawing. *Journal of Materials Processing Technology* 2004; **151**(1-3):70–79.
- [3] Hill R, Rice JR. Constitutive analysis of elastic-plastic crystals at arbitrary strain. *Journal of the Mechanics and Physics of Solids* 1972; **20**(6):401–413.
- [4] Asaro RJ, Rice JR. Strain localization in ductile single crystals. *Journal of the Mechanics and Physics of Solids* 1977; **25**(5):309–338.
- [5] Asaro RJ. Micromechanics of crystals and polycrystals. *Advances in applied mechanics* 1983; **23**(1):1–115.
- [6] Harewood FJ, McHugh PE. Investigation of finite element mesh independence in rate dependent materials. *Computational Materials Science* 2006; **37**(4):442–453.
- [7] Peirce D, Asaro RJ, Needleman A. An analysis of nonuniform and localized deformation in ductile single crystals. *Acta metallurgica* 1982; **30**(6):1087–1119.
- [8] Asaro R, Needleman A. Texture development and strain hardening in rate dependent polycrystals. *Acta Metall.* 1985; **33**(6):923–953.
- [9] Zhao Z, Roters F, Mao W, Raabe D. Introduction of a texture component crystal plasticity finite element method for anisotropy simulations. *Advanced Engineering Materials* 2001; **3**(12):984–990.
- [10] Raabe D, Roters F. Using texture components in crystal plasticity finite element simulations. *International Journal of Plasticity* 2004; **20**(3):339–361.
- [11] Roters F, Eisenlohr P, Hantcherli L, Tjahjanto D, Bieler T, Raabe D. Overview of constitutive laws, kinematics, homogenization and multiscale methods in crystal plasticity finite-element modeling: Theory, experiments, applications. *Acta Materialia* 2010; **58**(4):1152–1211.
- [12] Ritz H, Dawson PR. Sensitivity to grain discretization of the simulated crystal stress distributions in FCC polycrystals. *Modelling and Simulation in Materials Science and Engineering* 2009; **17**(1):1–21.
- [13] Yu T, Shi H. Effects of grain size distribution on the creep damage evolution of polycrystalline materials. *Journal of Physics D: Applied Physics* 2010; **43**:165401.
- [14] Berbenni S, Favier V, Berveiller M. Micro-macro modelling of the effects of the grain size distribution on the plastic flow stress of heterogeneous materials. *Computational Materials Science* 2007; **39**(1):96–105.

- [15] Berbenni S, Favier V, Berveiller M. Impact of the grain size distribution on the yield stress of heterogeneous materials. *International Journal of Plasticity* 2007; **23**(1):114–142.
- [16] Balint DS, Deshpande VS, Needleman A, Giessen E. Discrete dislocation plasticity analysis of the grain size dependence of the flow strength of polycrystals. *International Journal of Plasticity* 2008; **24**(12):2149–2172.
- [17] Balint DS, Deshpande VS, Needleman A, Giessen E. Size effects in uniaxial deformation of single and polycrystals: a discrete dislocation plasticity analysis. *Modelling and Simulation in Materials Science and Engineering* 2006; **14**:409–422.
- [18] Wang S, Zhuang W, Balint D, Lin J. A crystal plasticity study of the necking of micro-films under tension. *Journal of Multiscale Modelling* 2009; **3**(4):331–345.
- [19] Cailletaud G, Forest S, Jeulin D, Feyel F, Galliet I, Mounoury V, Quilici S. Some elements of microstructural mechanics. *Computational Materials Science* 2003; **27**(3):351–374.
- [20] Kim BN, Hiraga K, Morita K, Chen IW. Rate of creep due to grain-boundary diffusion in polycrystalline solids with grain-size distribution. *Philosophical Magazine* 2005; **85**(20):2281–2292.
- [21] Yang F, Yang W. Crack growth versus blunting in nanocrystalline metals with extremely small grain size. *Journal of the Mechanics and Physics of Solids* 2009; **57**(2):305–324.
- [22] Boots BN. The arrangement of cells in random networks. *Metallography* 1982; **15**(1):53–62.
- [23] Aboav DA. The arrangement of grains in a polycrystal. *Metallography* 1970; **3**(4):383–390.
- [24] Schwertel J, Stamm H. Analysis and modelling of tessellations by means of image analysis methods. *Journal of Microscopy* 1997; **186**(2):198–209.
- [25] Serra J. *Image Analysis and Mathematical Morphology*. Academic Press: London, 1988.
- [26] Cremona M, Mauricio MH, Scavarda D, Prioli R, Nunes VB, Zanette SI, Caride AO, Albuquerque MP. Grain size distribution analysis in polycrystalline thin films by mathematical morphology techniques on afm images and x-ray diffraction data. *Journal of microscopy* 2000; **197**(3):260.
- [27] Den Toonder J, Van Dommelen J, Baaijens F. Relation between single crystal elasticity and the effective elastic behaviour of polycrystalline materials: Theory, measurement and computation. *Modelling and Simulation in Materials Science and Engineering* 1999; **7**(6):909–928.
- [28] Clayton JD. Dynamic plasticity and fracture in high density polycrystals: constitutive modeling and numerical simulation. *Journal of the Mechanics and Physics of Solids* 2005; **53**(2):261–301.
- [29] Vander Voort G. *Metallography and Microstructures, Vol 9, ASM Handbook*, chap. Color Metallography. ASM International, 2004; 493–512.
- [30] Adams BL. Orientation imaging microscopy: application to the measurement of grain boundary structure. *Materials Science and Engineering: A* 1993; **166**(1-2):59–66.
- [31] Prangnell PB, Heason CP. Grain structure formation during friction stir welding observed by the 'stop action technique'. *Acta materialia* 2005; **53**(11):3179–3192.

- [32] Mironov S, Zhang Y, Sato Y, Kokawa H. Development of grain structure in [beta]-phase field during friction stir welding of ti-6al-4v alloy. *Scripta Materialia* 2008; **59**(1):27–30.
- [33] Raabe D, Sachtleber M, Zhao Z, Roters F, Zaefferer S. Micromechanical and macromechanical effects in grain scale polycrystal plasticity experimentation and simulation. *Acta Materialia* 2001; **49**(17):3433–3441.
- [34] Sachtleber M, Zhao Z, Raabe D. Experimental investigation of plastic grain interaction. *Materials Science and Engineering A* 2002; **336**(1-2):81–87.
- [35] Becker R, Richmond O. Incorporation of microstructural geometry in material modelling. *Modelling and Simulation in Materials Science and Engineering* 1994; **2**:439.
- [36] Panchanadeeswaran S, Doherty RD. Direct observation of orientation change by channel die compression polycrystalline aluminum - use of split sample. *SCR METALL MATER.* 1993; **28**(2):213–218.
- [37] Delaire F, Raphanel JL, Rey C. Plastic heterogeneities of a copper multicrystal deformed in uniaxial tension: experimental study and finite element simulations. *Acta materialia* 2000; **48**(5):1075–1087.
- [38] Cheong K, Busso E. Effects of lattice misorientations on strain heterogeneities in fcc polycrystals. *Journal of the Mechanics and Physics of Solids* 2006; **54**(4):671–689.
- [39] Zhao Z, Ramesh M, Raabe D, Cuitino A, Radovitzky R. Investigation of three-dimensional aspects of grain-scale plastic surface deformation of an aluminum oligocrystal. *International Journal of Plasticity* 2008; **24**(12).
- [40] Groeber MA, Rowenhorst DJ, Uchic MD. Collection, processing, and analysis of three-dimensional ebsd data sets. *Electron Backscatter Diffraction in Materials Science*. Springer, 2009; 123–137.
- [41] Wolfsdorf TL, Bender WH, Voorhees PW. The morphology of high volume fraction solid-liquid mixtures: An application of microstructural tomography. *Acta materialia* 1997; **45**(6):2279–2295.
- [42] Alkemper J, Voorhees PW. Quantitative serial sectioning analysis. *Journal of Microscopy* 2001; **201**(3):388–394.
- [43] Spowart J, Mullens H, Puchala B. Collecting and analyzing microstructures in three dimensions: A fully automated approach. *JOM* 2003; **55**(10):35–37.
- [44] Uchic MD, Groeber MA, Dimiduk DM, Simmons JP. 3D microstructural characterization of nickel superalloys via serial-sectioning using a dual beam FIB-SEM. *Scripta Materialia* 2006; **55**(1):23–28.
- [45] Groeber MA, Haley BK, Uchic MD, Dimiduk DM, Ghosh S. 3D reconstruction and characterization of polycrystalline microstructures using a FIB-SEM system. *Materials Characterization* 2006; **57**(4-5):259–273.
- [46] Zaefferer S, Wright SI, Raabe D. Three-dimensional orientation microscopy in a focused ion beam–scanning electron microscope: A new dimension of microstructure characterization. *Metallurgical and Materials Transactions A* 2008; **39**(2):374–389.

- [47] Rowenhorst DJ, Gupta A, Feng CR, Spanos G. 3D crystallographic and morphological analysis of coarse martensite: combining ebsd and serial sectioning. *Scripta Materialia* 2006; **55**(1):11–16.
- [48] Uchic MD, Groeber MA, Rollett AD. Automated serial sectioning methods for rapid collection of 3-d microstructure data. *JOM* 2011; **63**(3):25–29.
- [49] Spowart JE. Automated serial sectioning for 3-d analysis of microstructures. *Scripta materialia* 2006; **55**(1):5–10.
- [50] Mulders J, Day AP. Three-dimensional texture analysis. *Materials Science Forum*, vol. 495, 2005; 237–244.
- [51] Langer SA, Fuller Jr ER, Carter WC. OOF: an image-based finite-element analysis of material microstructures. *Computing in Science & Engineering* 2001; **3**(3):15–23.
- [52] Shan Z, Gokhale AM. Digital image analysis and microstructure modeling tools for microstructure sensitive design of materials. *International Journal of Plasticity* 2004; **20**(7):1347–1370.
- [53] Chawla N, Ganesh VV, Wunsch B. Three-dimensional (3d) microstructure visualization and finite element modeling of the mechanical behavior of sic particle reinforced aluminum composites. *Scripta materialia* 2004; **51**(2):161–165.
- [54] Chawla N, Deng X. Microstructure and mechanical behavior of porous sintered steels. *Materials Science and Engineering A* 2005; **390**(1-2):98–112.
- [55] Spanos G, Rowenhorst DJ, Lewis AC, Geltmacher AB. Combining serial sectioning, ebsd analysis, and image-based finite element modeling. *MRS Bulletin* 2008; **33**(6):597–602.
- [56] Lewis AC, Bingert JF, Rowenhorst DJ, Gupta A, Geltmacher AB, Spanos G. Two-and three-dimensional microstructural characterization of a super-austenitic stainless steel. *Materials Science and Engineering A* 2006; **418**(1-2):11–18.
- [57] Lewis AC, Geltmacher AB. Image-based modeling of the response of experimental 3D microstructures to mechanical loading. *Scripta materialia* 2006; **55**(1):81–85.
- [58] Lewis AC, Qidwai SM, Geltmacher AB. Slip systems and initiation of plasticity in a body-centered-cubic titanium alloy. *Metallurgical and Materials Transactions A* 2010; :1–10.
- [59] Bhandari Y, Sarkar S, Groeber M, Uchic MD, Dimiduk DM, Ghosh S. 3D polycrystalline microstructure reconstruction from FIB generated serial sections for FE analysis. *Computational Materials Science* 2007; **41**(2):222–235.
- [60] Ghosh S, Bhandari Y, Groeber M. CAD-based reconstruction of 3D polycrystalline alloy microstructures from FIB generated serial sections. *Computer-Aided Design* 2008; **40**(3):293–310.
- [61] Poulsen HF, Garbe S, Lorentzen T, Juul Jensen D, Poulsen FW, Andersen NH, Frello T, Feidenhans'l R, Graafsma H. Applications of high-energy synchrotron radiation for structural studies of polycrystalline materials. *Journal of Synchrotron Radiation* 1997; **4**(3):147–154.
- [62] Poulsen HF. 3DXRD - a new probe for materials science. PhD Thesis, Risø National Laboratory, Roskilde, Denmark 2004.

- [63] Reimers W, Pyzalla AR, Schreyer A, Clemens H. *Neutrons and Synchrotron Radiation in Engineering Materials Science*. Wiley, 2008.
- [64] Nielsen SF, Poulsen HF, Beckmann F, Thorning C, Wert JA. Measurements of plastic displacement gradient components in three dimensions using marker particles and synchrotron x-ray absorption microtomography. *Acta materialia* 2003; **51**(8):2407–2415.
- [65] Martins RV, Margulies L, Schmidt S, Poulsen HF, Leffers T. Simultaneous measurement of the strain tensor of 10 individual grains embedded in an Al tensile sample. *Materials Science and Engineering: A* 2004; **387**:84–88.
- [66] Aydiner C, Bernier J, Clausen B, Lienert U, Tomé C, Brown D. Evolution of stress in individual grains and twins in a magnesium alloy aggregate. *Physical Review B* 2009; **80**(2):024113.
- [67] Oddershede J, Schmidt S, Poulsen HF, Sorensen HO, Wright J, Reimers W. Determining grain resolved stresses in polycrystalline materials using three-dimensional x-ray diffraction. *Journal of Applied Crystallography* 2010; **43**:539–549.
- [68] Schmidt S, Nielsen SF, Gundlach C, Margulies L, Huang X, Jensen DJ. Watching the growth of bulk grains during recrystallization of deformed metals. *Science* 2004; **305**(5681):229.
- [69] Lauridsen EM, Schmidt S, Nielsen SF, Margulies L, Poulsen HF, Jensen DJ. Non-destructive characterization of recrystallization kinetics using three-dimensional x-ray diffraction microscopy. *Scripta Materialia* 2006; **55**(1):51–56.
- [70] Ludwig W, King A, Reischig P, Herbig M, Lauridsen E, Schmidt S, Proudhon H, Forest S. New opportunities for 3d materials science of polycrystalline materials at the micrometre lengthscale by combined use of x-ray diffraction and x-ray imaging. *Materials Science and Engineering: A* 2009; **524**(1-2):69–76.
- [71] Martin T, Koch A. Recent developments in x-ray imaging with micrometer spatial resolution. *Journal of Synchrotron Radiation* 2006; **13**(2):180–194.
- [72] Ludwig W, Reischig P, King A, Herbig M, Lauridsen E, Johnson G. Three-dimensional grain mapping by x-ray diffraction contrast tomography and the use of friedel pairs in diffraction data analysis. *Review of Scientific Instruments* 2009; **80**:033905.
- [73] Cahn JW, Hilliard JE. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of Chemical Physics* 1958; **28**:258–267.
- [74] Cahn JW. On spinodal decomposition. *Acta Metallurgica* 1961; **9**(9):795–801.
- [75] Allen SM, Cahn JW. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica* 1979; **27**(6):1085–1095.
- [76] Collins J, Levine H. Diffuse interface model of diffusion-limited crystal growth. *Physical Review B* 1985; **31**(9):6119–6122.
- [77] Kobayashi R. Modeling and numerical simulations of dendritic crystal growth. *Physica D: Nonlinear Phenomena* 1993; **63**(3-4):410–423.
- [78] Chen LQ, Yang W. Computer simulation of the domain dynamics of a quenched system with a large number of nonconserved order parameters: The grain-growth kinetics. *Physical Review B* 1994; **50**(21):15752–56.

- [79] Fan D, Geng C, Chen LQ. Computer simulation of topological evolution in 2-d grain growth using a continuum diffuse-interface field model. *Acta materialia* 1997; **45**(3):1115–1126.
- [80] Krill III C, Chen LQ. Computer simulation of 3-d grain growth using a phase-field model. *Acta Materialia* 2002; **50**(12):3059–3075.
- [81] Suwa Y, Saito Y, Onodera H. Three-dimensional phase field simulation of the effect of anisotropy in grain-boundary mobility on growth kinetics and morphology of grain structure. *Computational materials science* 2007; **40**(1):40–50.
- [82] Steinbach I, Pezzolla F, Nestler B, Seeßelberg M, Prieler R, Schmitz G, Rezende J. A phase field concept for multiphase systems. *Physica D* 1996; **94**(3):135–147.
- [83] Hesselbarth HW, Gobel IR. Simulation of recrystallization by cellular automata. *Acta Metallurgica et Materialia* 1991; **39**(9):2135–2143.
- [84] Raabe D. Mesoscale simulation of recrystallization textures and microstructures. *Advanced Engineering Materials* 2001; **3**(10):745–752.
- [85] Brown SGR, Williams T, Spittle JA. Cellular automaton model of the steady-state free growth of a non-isothermal dendrite. *Acta Metallurgica et Materialia* 1994; **42**:2893–2898.
- [86] Geiger J, Roosz A, Barkoczy P. Simulation of grain coarsening in two dimensions by cellular-automaton. *Acta materialia* 2001; **49**(4):623–629.
- [87] Ding HL, He YZ, Liu LF, Ding WJ. Cellular automata simulation of grain growth in three dimensions based on the lowest-energy principle. *Journal of crystal growth* 2006; **293**(2):489–497.
- [88] Kumar M, Sasikumar R, Kesavan Nair P. Competition between nucleation and early growth of ferrite from austenite—studies using cellular automaton simulations. *Acta materialia* 1998; **46**(17):6291–6303.
- [89] Raabe D, Roters F, Barlat F, Chen LQ. Cellular, lattice gas, and boltzmann automata. *Continuum Scale Simulation of Engineering Materials*. Wiley, 2004; 57–76.
- [90] Raabe D, Becker RC. Coupling of a crystal plasticity finite-element model with a probabilistic cellular automaton for simulating primary static recrystallization in aluminium. *Modelling and Simulation in Materials Science and Engineering* 2000; **8**:445–462.
- [91] Lan YJ, Xiao NM, Li DZ, Li YY. Mesoscale simulation of deformed austenite decomposition into ferrite by coupling a cellular automaton method with a crystal plasticity finite element model. *Acta materialia* 2005; **53**(4):991–1003.
- [92] Anderson M, Srolovitz D, Grest G, Sahni P. Computer simulation of grain growth—i. kinetics. *Acta Metallurgica* 1984; **32**(5):783–791.
- [93] Binder K, Young A. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern physics* 1986; **58**(4):801.
- [94] Anderson MP, Grest GS, Srolovitz DJ. Computer simulation of normal grain growth in three dimensions. *Philosophical Magazine Part B* 1989; **59**(3):293–329.
- [95] Yang W, Chen L, Messing G. Computer simulation of anisotropic grain growth. *Materials Science and Engineering A* 1995; **195**:179–187.

- [96] Holm E, Srolovitz D, Cahn J. Microstructural evolution in two-dimensional two-phase polycrystals. *Acta metallurgica et materialia* 1993; **41**(4):1119–1136.
- [97] Holm E, Zacharopoulos N, Srolovitz D. Nonuniform and directional grain growth caused by grain boundary mobility variations. *Acta materialia* 1998; **46**(3):953–964.
- [98] Srolovitz D, Grest G, Anderson M. Computer simulation of recrystallization—I. homogeneous nucleation and growth. *Acta metallurgica* 1986; **34**(9):1833–1845.
- [99] Zhu P, Smith RW. Dynamic simulation of crystal growth by monte carlo method—I. model description and kinetics. *Acta metallurgica et materialia* 1992; **40**(4):683–692.
- [100] Holm E, Glazier J, Srolovitz D, Grest G. Effects of lattice anisotropy and temperature on domain growth in the two-dimensional potts model. *Physical Review A* 1991; **43**(6):2662.
- [101] Senechal M. Which tetrahedra fill space? *Mathematics Magazine* 1981; **54**(5):227–243.
- [102] Bronkhorst CA, Kalidindi SR, Anand L. Polycrystalline plasticity and the evolution of crystallographic texture in FCC metals. *Philosophical Transactions: Physical Sciences and Engineering* 1992; **341**(1662):443–477.
- [103] Kalidindi S, Bronkhorst C, Anand L. Crystallographic texture evolution in bulk deformation processing of FCC metals. *Journal of Mechanics and Physics of Solids* 1992; **40**:537–569.
- [104] Delannay L, Jacques PJ, Kalidindi SR. Finite element modeling of crystal plasticity with grains shaped as truncated octahedrons. *International Journal of Plasticity* 2006; **22**(10):1879–1898.
- [105] Savage P, O'Donnell BP, McHugh PE, Murphy BP, Quinn DF. Coronary stent strut size dependent stress–strain response investigated using micromechanical finite element models. *Annals of Biomedical Engineering* 2004; **32**(2):202–211.
- [106] McGarry JP, O'Donnell BP, McHugh PE, O'Cearbhaill E, McMeeking RM. Computational examination of the effect of material inhomogeneity on the necking of stent struts under tensile loading. *Journal of applied mechanics* 2007; **74**(5):978–989.
- [107] Mika DP, Dawson PR. Effects of grain interaction on deformation in polycrystals. *Materials Science and Engineering A* 1998; **257**(1):62–76.
- [108] Bunge HJ, Kiewel R, Reinert T, Fritsche L. Elastic properties of polycrystals—influence of texture and stereology. *Journal of the Mechanics and Physics of Solids* 2000; **48**(1):29–66.
- [109] Aboav DA, Langdon TG. The shape of grains in a polycrystal. *Metallography* 1969; **2**(2-3):171–178.
- [110] Weaire D. Some remarks on the arrangement of grains in a polycrystal. *Metallography* 1974; **7**(2):157–160.
- [111] Lemaitre J, Gervois A, Trodec JP, Rivier N, Ammi M, Oger L, Bideau D. Arrangement of cells in Voronoi tessellations of monosize packing of discs. *Philosophical Magazine B* 1993; **67**(3):347–362.
- [112] Watanabe O, Zbib HM, Takenouchi E. Crystal plasticity: micro-shear banding in polycrystals using Voronoi tessellation. *International Journal of Plasticity* 1998; **14**(8):771–788.

- [113] Johnson WA, Mehl RF. Reaction kinetics in processes of nucleation and growth. *Trans. AIME* 1939; **135**(8):396–415.
- [114] Evans UR. The laws of expanding circles and spheres in relation to the lateral growth of surface films and the grain-size of metals. *Transactions of the Faraday Society* 1945; **41**:365–374.
- [115] Meijering JL. Interface area, edge length, and number of vertices in crystal aggregates with random nucleation. *Philips Res. Rep* 1953; **8**:270–290.
- [116] Gilbert EN. Random subdivisions of space into crystals. *The Annals of mathematical statistics* 1962; **33**(3):958–972.
- [117] Frost HJ, Thompson CV. The effect of nucleation conditions on the topology and geometry of two-dimensional grain structures. *Acta metallurgica* 1987; **35**(2):529–540.
- [118] Frost HJ, Thompson CV. Development of microstructure in thin films. *Proceedings of SPIE-The international Society. for Optical Engineering*, 1987; 77–87.
- [119] Frost HJ, Thompson CV. Computer simulation of microstructural evolution in thin films. *Journal of Electronic Materials* 1988; **17**(5):447–458.
- [120] Schaudt BF, Drysdale RL. Multiplicatively weighted crystal growth voronoi diagrams (extended abstract) 1991; :214–223.
- [121] Aparicio N, Cocks A. On the representation of random packings of spheres for sintering simulations. *Acta metallurgica et materialia* 1995; **43**(10):3873–3884.
- [122] Inaba M, Katoh N, Imai H. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering. *Proceedings of the tenth annual symposium on Computational geometry*, ACM, 1994; 332–339.
- [123] Mahadevan S, Zhao Y. Advanced computer simulation of polycrystalline microstructure. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**(34):3651–3667.
- [124] Ito O, Fuller Jr ER. Computer modelling of anisotropic grain microstructure in two dimensions. *Acta Metallurgica et Materialia* 1993; **41**(1):191–198.
- [125] Frost HJ, Thompson CV, Walton DT. Simulation of thin film grain structures–i. grain growth stagnation. *Acta Metallurgica et Materialia* 1990; **38**(8):1455–1462.
- [126] Frost H, Thompson C, Walton D. Simulation of thin film grain structures–ii. abnormal grain growth. *Acta metallurgica et materialia* 1992; **40**(4):779–793.
- [127] Frost HJ, Thompson CV. Computer simulation of grain growth. *Current Opinion in Solid State and Materials Science* 1996; **1**(3):361–368.
- [128] Hatem TM, Zikry MA. Dislocation density crystalline plasticity modeling of lath martensitic microstructures in steel alloys. *Philosophical Magazine* 2009; **89**:3087–3109.
- [129] Fu HH, Benson DJ, Andre Meyers M. Computational description of nanocrystalline deformation based on crystal plasticity. *Acta Materialia* 2004; **52**(15):4413–4425.
- [130] Wei YJ, Anand L. Grain-boundary sliding and separation in polycrystalline metals: application to nanocrystalline fcc metals. *Journal of the Mechanics and Physics of Solids* 2004; **52**(11):2587–2616.

- [131] Simonovski I, NILSSON KF, Cizelj L. Crack tip displacements of microstructurally small cracks in 316l steel and their dependence on crystallographic orientations of grains. *Fatigue & Fracture of Engineering Materials & Structures* 2007; **30**(6):463–478.
- [132] Simonovski I, Cizelj L. The influence of grains' crystallographic orientations on advancing short crack. *International Journal of Fatigue* 2007; **29**(9-11):2005–2014.
- [133] Kanit T, Forest S, Galliet I, Mounoury V, Jeulin D. Determination of the size of the representative volume element for random composites: statistical and numerical approach. *International Journal of Solids and Structures* 2003; **40**(13-14):3647–3679.
- [134] Kovac M, Cizelj L. Modeling elasto-plastic behavior of polycrystalline grain structure of steels at mesoscopic level. *Nuclear engineering and design* 2005; **235**(17-19):1939–1950.
- [135] Simonovski I, Kova M, Cizelj L. Estimating the correlation length of inhomogeneities in a polycrystalline material. *Materials Science and Engineering A* 2004; **381**(1-2):273–280.
- [136] Zhuang W, Wang S, Cao J, Lin J, Hartl C. Modelling of localised thinning features in the hydroforming of micro-tubes using the crystal-plasticity FE method. *The International Journal of Advanced Manufacturing Technology* 2010; **47**(9):859–865.
- [137] Wang S, Zhuang W, Balint D, Lin J. A virtual crystal plasticity simulation tool for micro-forming. *Procedia Engineering* 2009; **1**(1):75 – 78.
- [138] Wang S, Zhuang W, Cao J, Lin J. An investigation of springback scatter in forming ultra-thin metal-sheet channel parts using crystal plasticity FE analysis. *The International Journal of Advanced Manufacturing Technology* 2010; **47**(9):845–852.
- [139] Weinzapfel N, Sadeghi F, Bakolas V. An approach for modeling material grain structure in investigations of hertzian subsurface stresses and rolling contact fatigue. *Journal of Tribology* 2010; **132**(4):041404.
- [140] Zhang P, Balint D, Lin J. An integrated scheme for crystal plasticity analysis: Virtual grain structure generation. *Computational Materials Science* 2011; **50**(10):2854–2864.
- [141] Brahme A, Alvi MH, Saylor D, Fridy J, Rollett AD. 3d reconstruction of microstructure in a commercial purity aluminum. *Scripta Materialia* 2006; **55**(1):75–80.
- [142] Groeber M, Ghosh S, Uchic M, Dimiduk D. A framework for automated analysis and simulation of 3D polycrystalline microstructures. part 2: Synthetic structure generation. *Acta Materialia* 2008; **56**(6):1274–1287.
- [143] Okabe A, Boots B, Sugihara K, Chiu SN. *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons Inc, 2000.
- [144] Conrad H, Swintowski M, Mannan S. Effect of cold work on recrystallization behavior and grain size distribution in titanium. *Metallurgical and Materials Transactions A* 1985; **16**(5):703–708.
- [145] Louat N. On the theory of normal grain growth. *Acta Metallurgica* 1974; **22**(6):721–724.
- [146] Hanson HG. Voronoi cell properties from simulated and real random spheres and points. *Journal of statistical physics* 1983; **30**(3):591–605.

- [147] Oger L, Gervois A, Troadec JP, Rivier N. Voronoi tessellation of packings of spheres: Topological correlation and statistics. *Philosophical Magazine Part B* 1996; **74**(2):177–197.
- [148] Kumar S, Kurtz SK, Banavar JR, Sharma MG. Properties of a three-dimensional Poisson-Voronoi tessellation: A Monte Carlo study. *Journal of statistical physics* 1992; **67**(3):523–551.
- [149] Kiang T. Random fragmentation in two and three dimensions. *Zeitschrift fur Astrophysik* 1966; **64**:433.
- [150] Andrade PN, Fortes MA. Distribution of cell volumes in a voronoi partition. *Philosophical Magazine Part B* 1988; **58**(6):671–674.
- [151] Kumar S, Kurtz SK. Properties of a two-dimensional Poisson-Voronoi tessellation: A Monte-Carlo study. *Materials Characterization* 1993; **31**(1):55–68.
- [152] Hinde AL, Miles RE. Monte Carlo estimates of the distributions of the random polygons of the Voronoi tessellation with respect to a Poisson process. *Journal of Statistical Computation and Simulation* 1980; **10**(3):205–223.
- [153] Weaire D, Kermode JP, Wejchert J. On the distribution of cell areas in a Voronoi network. *Philosophical Magazine B* 1986; **53**(5):101–105.
- [154] Zhu H, Thorpe S, Windle A. The geometrical properties of irregular two-dimensional Voronoi tessellations. *Philosophical Magazine A* 2001; **81**(12):2765–2783.
- [155] Wejchert J, Weaire D, Kermode JP. Monte Carlo simulation of the evolution of a two-dimensional soap froth. *Philosophical Magazine B* 1986; **53**(1):15–24.
- [156] Bailey DH, Yozo H, Li XS, Thompson B. ARPPEC: An arbitrary precision computation package. *Technical Report*, Lawrence Berkeley National Laboratory 2002.
- [157] Ho KC, Zhang N, Lin J, Dean TA. An integrated approach for virtual microstructure generation and micromechanics modelling for micro-forming simulation. *Proceedings of ASME MNC2007*, 2007.
- [158] Cao J, Zhuang W, Wang S, Ho KC, Zhang N, Lin J, Dean TA. An integrated crystal plasticity FE system for microforming simulation. *Journal of Multiscale Modelling* 2009; **1**:107–124.
- [159] Cao J, Zhuang W, Wang S, Lin J. Development of a VGRAIN system for CPFE analysis in micro-forming applications. *The International Journal of Advanced Manufacturing Technology* 2010; **47**:981–991.
- [160] Alzer H. On some inequalities for the gamma and psi functions. *Mathematics of Computation* 1997; **66**(217):373–389.
- [161] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 2001; **9**(2):159–195.
- [162] Parasiz SA, Kinsey B, Krishnan N, Cao J, Li M. Investigation of deformation size effects during microextrusion. *Journal of Manufacturing Science and Engineering* 2007; **129**:690.
- [163] Chandra A, Huang Y, Jiang ZQ, Hu KX, Fu G. A model of crack nucleation in layered electronic assemblies under thermal cycling. *Journal of Electronic Packaging* 2000; **122**(9):220–226.

- [164] McClintock FA. A criterion for ductile fracture by growth of holes. *Journal of Applied Mechanics* 1968; **35**:363–371.
- [165] Rice JR, Tracey DM. On the ductile enlargement of voids in triaxial stress fields. *Journal of the Mechanics and Physics of Solids* 1969; **17**(3):201–217.
- [166] Needleman A, Tvergaard V, Hutchinson JW. Void growth in plastic solids. *Topics in fracture and fatigue*. Springer, 1992.
- [167] Bonfoh N, Lipinski P, Carmasol A, Tiem S. Micromechanical modeling of ductile damage of polycrystalline materials with heterogeneous particles. *International Journal of Plasticity* 2004; **20**(1):85–106.
- [168] Hutchinson JW. Singular behaviour at the end of a tensile crack in a hardening material. *Journal of the Mechanics and Physics of Solids* 1968; **16**(1):13–31.
- [169] Rice JR, Rosengren GF. Plane strain deformation near a crack tip in a power-law hardening material. *Journal of the Mechanics and Physics of Solids* 1968; **16**(1):1–12.
- [170] Rice JR. Tensile crack tip fields in elastic-ideally plastic crystals. *Mechanics of Materials* 1987; **6**(4):317–335.
- [171] Rice JR. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics* 1968; **35**:379–386.
- [172] Hadavinia H, Kawashita L, Kinloch AJ, Moore DR, Williams JG. A numerical analysis of the elastic-plastic peel test. *Engineering fracture mechanics* 2006; **73**(16):2324–2335.
- [173] Needleman A. An analysis of tensile decohesion along an interface. *Journal of the Mechanics and Physics of Solids* 1990; **38**(3):289–324.
- [174] de Borst R, Remmers J, Needleman A. Mesh-independent discrete numerical representations of cohesive-zone models. *Engineering fracture mechanics* 2006; **73**(2):160–177.
- [175] Alfano G, Crisfield MA. Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues. *International Journal for Numerical Methods in Engineering* 2001; **50**(7):1701–1736.
- [176] Robinson P, Besant T, Hitchings D. Delamination growth prediction using a finite element approach. *Second ESIS TC4 Conference on Fracture of Polymers, Composites and Adhesives, European Structural Integrity Society*, vol. 27, Williams J, Pavan A (eds.). 2000; 135–147.
- [177] Moes N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**(1):131–150.
- [178] Moes N, Belytschko T. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics* 2002; **69**:813–833.
- [179] Dugdale DS. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids* 1960; **8**(2):100–104.
- [180] Barenblatt GI. The mathematical theory of equilibrium cracks in brittle fracture. *Advances in Applied Mechanics* 1962; **7**:55–129.
- [181] Needleman A. A continuum model for void nucleation by inclusion debonding. *Journal of Applied Mechanics* 1987; **54**:525.

- [182] Rice JR, Wang JS. Embrittlement of interfaces by solute segregation. *Materials Science and Engineering: A* 1989; **107**:23–40.
- [183] Ruiz G, Pandolfi A, Ortiz M. Three-dimensional cohesive modeling of dynamic mixed-mode fracture. *International Journal for Numerical Methods in Engineering* 2001; **52**(1-2):97–120.
- [184] Helms K, Allen DH, Hurtado LD. A model for predicting grain boundary cracking in polycrystalline viscoplastic materials including scale effects. *International Journal of Fracture* 1999; **95**(1):175–194.
- [185] Zavattieri PD, Espinosa HD. Grain level analysis of crack initiation and propagation in brittle materials. *Acta Materialia* 2001; **49**(20):4291–4311.
- [186] Marfia S, Sacco E. Numerical techniques for the analysis of crack propagation in cohesive materials. *International Journal for Numerical Methods in Engineering* 2003; **57**(11):1577–1602.
- [187] Held M. *On the computational geometry of pocket machining*. Lecture Notes In Computer Science, Springer-Verlag New York, Inc., 1991.
- [188] Aggarwal A, Guibas L, Saxe J, Shor P. A linear time algorithm for computing the Voronoi diagram of a convex polygon. *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987; 39–45.
- [189] Zhang P, Balint D, Lin J. Controlled poisson Voronoi tessellation for virtual grain structure generation: a statistical evaluation. *Philosophical Magazine* 2011; doi: 10.1080/14786435.2011.613860.
- [190] Elices M, Guinea G, Gomez J, Planas J. The cohesive zone model: advantages, limitations and challenges. *Engineering fracture mechanics* 2002; **69**(2):137–163.
- [191] Karimpour M, Balint D, Lin J, Farrugia D. A crystal plasticity analysis of the effect of second phase inclusions on damage in free-cutting steels. *To be submitted* ; .
- [192] Boots BN, Murdoch DJ. The spatial arrangement of random Voronoi polygons. *Computers and Geosciences* 1983; **9**(3):351–365.
- [193] Møller J. Random tessellations in R^d . *Advances in Applied Probability* 1989; **21**(1):37–73.
- [194] Kumar S, Kurtz SK, Weaire D. Average number of sides for the neighbours in a Poisson-Voronoi tessellation. *Philosophical Magazine B* 1994; **69**(3):431–435.
- [195] Zhu HX, Hobdell JR, Windle AH. Effects of cell irregularity on the elastic properties of open-cell foams. *Acta materialia* 2000; **48**(20):4893–4900.
- [196] Zhu H, Thorpe S, Elliott J, Windle A. The statistical properties of irregular three-dimensional Voronoi tessellations. *Technical Report*, University of Cambridge 2001.
- [197] Evans M, Hastings N, Peacock B. *Statistical distributions*. John Wiley and Sons, 2000.
- [198] Hutchinson J. Bounds and self-consistent estimates for creep of polycrystalline materials. *Proceedings of the Royal Society of London, Series A*, vol. 348, 1976; 101–127.
- [199] Peirce D, Asaro J, Needleman A. Material rate dependence and localized deformation in crystalline solids. *Acta Metallurgica* 1983; **31**(12):1951–176.

- [200] Huang Y. A user-material subroutine incorporating single crystal plasticity in the abaqus finite element program. *Technical Report*, Division of Applied Sciences, Harvard University 1991.
- [201] Fan J, Chen A, Fu M, Lu J. A novel structural gradient metallic glass composite with enhanced mechanical properties. *Scripta Materialia* 2009; **61**(6):608–611.
- [202] Jin H, Saimoto S, Ball M, Threadgill P. Characterisation of microstructure and texture in friction stir welded joints of 5754 and 5182 aluminium alloy sheets. *Materials science and technology* 2001; **17**(12):1605–1614.
- [203] Lee J, Ehrlich F, Crall L, Collins T. An analysis for the effect of a grain size gradient on torsional and tensile properties. *Metallurgical and Materials Transactions A* 1988; **19**(2):329–335.
- [204] Kokini K, Choules B. Surface thermal fracture of functionally graded ceramic coatings: effect of architecture and materials. *Composites Engineering* 1995; **5**(7):865–877.
- [205] Qin L, Xu J, Lian J, Jiang Z, Jiang Q. A novel electrodeposited nanostructured ni coating with grain size gradient distribution. *Surface and Coatings Technology* 2008; **203**(1-2):142–147.
- [206] Yang Z, Sista S, Elmer J, DebRoy T. Three dimensional monte carlo simulation of grain growth during gta welding of titanium. *Acta materialia* 2000; **48**(20):4813–4825.
- [207] Evans A, Clarke D, Levi C. The influence of oxides on the performance of advanced gas turbines. *Journal of the European Ceramic Society* 2008; **28**(7):1405–1419.