MODELING OF IMPACT DAMAGE IN MAGNESIUM NUCLEATED FROM REALISTIC MICROSTRUCTURES

A Thesis

by

ANGELA MARIE OLINGER

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Justin Wilkerson
Committee Members,	Alan Freed
	Thomas Lacy
	Ankit Srivastava
Head of Department,	Andreas Polycarpou

May 2020

Major Subject: Mechanical Engineering

ABSTRACT

Void growth is known to nucleate from pre-existing defects or particles in a matrix material. When these voids coalesce, ductile failure occurs. Rolled metals can have highly anisotropic particle geometries whose effects are not accurately captured by existing void growth models. This work seeks to better understand ductile damage that is nucleated from particle geometries present in real microstructures.

This work presents calibrated sets of GTN model parameters that match the uniaxial stress response of direct numerical simulations using real, explicitly modeled, particle geometries. Then a method of generating a representative unit cell from a real particle microstructure is presented. This method idealizes real second-phase particles as a distribution of ellipsoids, and generates a representative unit cell from those ellipsoids' sizes and nearest-neighbor spacings. The representative unit cell is then rotated to obtain a unit cell for a certain loading direction.

The nearest-neighbor spacing algorithm has limitations for certain particle spacings, and also tends to over-predict the porosity of the original unit cell compared to the original particle distribution. An improved method of spacing estimation is presented using a Voronoi tessellation which more closely reproduces porosity. The unit cells derived from the Voronoi tessellation were compared to direct numerical simulations of their source void distributions, and more closely reproduce the stress strain response of the direct numerical simulations than the nearest neighbor algorithm.

This process allows any microstructure to be idealized as a representative unit cell. These unit cell calculations are much less computationally expensive than directly simulating a real particle distribution, and can be used both for existing microstructures or for microstructural design.

ACKNOWLEDGEMENTS

A special thank you is owed to my advisor, Dr. Justin Wilkerson. I have learned so much under your guidance over the past year and a half. Working with you has been a pleasure and I am grateful for all of your assistance during my time in graduate school.

I would like to thank Dr. Freed, Dr. Lacy, and Dr. Srivastava for dedicating their time to serving on my thesis committee. I would also like to thank Dr. Jeffery Lloyd of the Army Research Lab. I truly appreciate all of your guidance and mentorship during my time at ARL.

Lastly, I would like to thank my family for all of their support in helping me get here. To my parents, Susan and Terry, my sister, Lee, and my wonderful fiancé Zach, thank you for always believing in me and encouraging me.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Justin Wilkerson [advisor], Dr. Alan Freed, and Dr. Thomas Lacy of the Department of Mechanical Engineering and Dr. Ankit Srivastava of the Department of Materials Science.

The data used for calibration in Chapter 2, and the material microstructure data used in Chapter 3 were provided by Dr. Jeffery Lloyd of the Army Research Lab. The ballistic simulations in Chapter 2 initially set up by the student, with assistance in running and extracting data from subsequent permutations provided by Caleb Foster. All other work conducted for the thesis was completed by the student independently.

Funding Sources

This material is based upon work supported by Army Research Laboratory under the MEDE Collaborative Research Alliance through Cooperative Agreement Number W911NF-12-2-0022. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Portions of this research were conducted with high performance research computing resources provided by Texas A&M University (https://hprc.tamu.edu).

NOMENCLATURE

CRSS	Critical Resolved Shear Stress
DNS	Direct Numerical Simulations
GLD	Gologanu-Leblond-Devoux Model
GTN	Gurson-Tvergaard-Needleman Model
HCP	Hexagonal Close Packed
ND	Normal Direction
RD	Rolling Direction
RVE	Representative Volume Element
TD	Transverse Direction

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1

INTRODUCTION & BACKGROUND

Ductile failure of metals occurs through the nucleation, growth, and coalescence of voids. As voids grow, they coalesce to form microcracks, and eventually macroscopic cracks [1]. Coalescence of voids can occur by necking in the matrix material between voids [2] [3], or by void sheeting due to the presence of smaller, secondary voids [4] [5]. Eventually, these cracks cause macroscopic failure. Voids may already exist in a material, or be nucleated at defects such as precipitates [3]. For voids nucleated at defects, they may nucleate by separating from the inclusion, or due to fracture of the second phase particle inclusion [2]. Since voids tend to nucleate from second phase particles, the particle size and spatial distributions, play a significant role in void nucleation and growth. Better understanding of void nucleation is of interest to many industries that use ductile metals, as knowledge of the impacts of microstructure on ductile failure can enable better microstructural design and selection for a particular application. Of particular interest is the ductile failure of metals subjected to ballistic impacts, specifically for armor material applications.

Recent developments in homogenized ductile damage modeling have focused on extending void growth models to account for non-spherical voids and uneven void spacing ([6],[7]). While these more recent works have made important extensions to the original Gurson-Tvergaard-Needleman (GTN) model ([8]), they all assume voids to be evenly distributed throughout a material, and generally assume all voids to be the same

size. In a real microstructure, voids tend to nucleate from second phase particles, which can have a wide range of sizes in a single material. Depending on the processing route, void nucleating particles can be relatively evenly distributed, or highly aligned with very different spacings in different directions. While the current models may be sufficient for modeling real microstructures that contain similarly sized second phase particles that are relatively evenly distributed, they still require selection of a characteristic particle size, aspect ratio, and spacing. In a real microstructure these parameters may vary greatly within the material, particularly for rolled materials which may contain very small spherical particles, along with much larger, more elongated. Correctly tuning these parameters to match the model response to that of the real microstructure requires comparison of the model response to complex and costly direct numerical simulations of a real microstructure.

The goal of this work is to provide a method of estimating the effects of elongated second phase particle orientation, without explicitly modeling the geometry of those particles. Chapter 2 focuses on the calibration of GTN model parameters to fit the stress strain curves of rolled AZ31B at different orientations that were generated through DNS of the particle geometry. Then, Chapter 3 suggests a method for directly estimating the geometry of a single representative unit cell from a real distribution of second phase particles. Chapter 4 then evaluates the accuracy of those estimated unit cells compared to the DNS, and Chapter 5 suggests an improved method of estimating the representative spacing of particles.

1.1 Magnesium

Magnesium is currently not widely used in structural applications where lightweight metals are needed. Instead aluminum and titanium are often favored. While magnesium has a high specific strength, it is limited by its low ductility compared to other structural metals [9]. Its relatively low density makes it potentially advantageous in applications where weight is important such as in vehicle armor, or spacecraft shielding. However, magnesium also presents certain challenges due to its complex behavior.

Magnesium has a hexagonal-close-packed crystal structure (HCP) with large variations in the critical resolved shear stress (CRSS) along the different slip systems [10]. The primary slip system is basal slip, which has the lowest CRSS. Magnesium also deforms by twinning, where the crystal lattice is reoriented to create a mirror of the original parent lattice [11]. The left and center panels of Figure 1 illustrate the HCP slip systems, and the right panel illustrates the twinning modes [10].

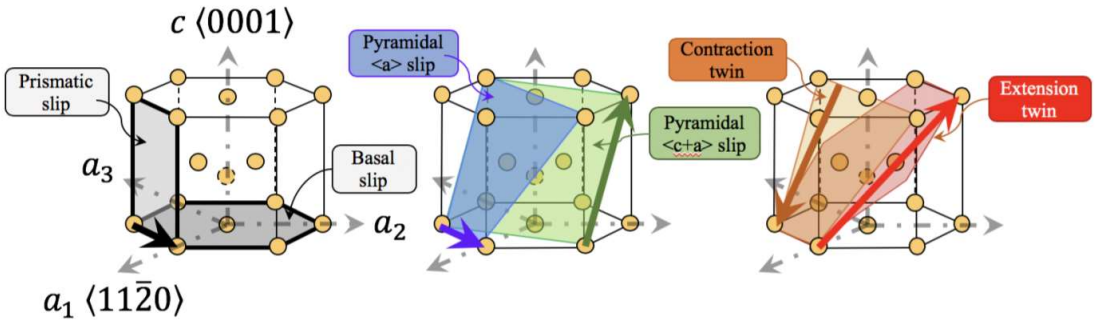

Figure 1: Deformation mechanisms in Mg (a) Basal and prismatic slip (b) Pyramidal slip and (c) Deformation twinning. Reprinted from [10].

When magnesium is rolled, the c-axes of the grains tend to align in the normal direction, which induces strong plastic anisotropy [12]. In addition to this plastic anisotropy, second phase particles present in the matrix material may have anisotropic geometries, particularly in the case of a rolled material, which induce anisotropy in failure strains [9]. Strong anisotropy is also seen in the spall strength, a measure of ballistic performance, of magnesium. Higher spall strengths are generally seen for shock compression along the a-axis, while c-axis compression sees a lower spall strength [10]. This is thought to be influenced by increased twinning for tension along the c-axis [10]. This work studies a special case of magnesium where basal slip and twinning modes are suppressed. This can be achieved through alloying or processing methods and essentially removes the plastic anisotropy, while anisotropic ductility remains due to anisotropic particle geometries.

1.2 Literature review of existing void growth models

The most commonly used void growth model is the GTN, based on the model for the growth of spherical voids by Gurson [13] later modified by Tvergaard and Needleman [8] to include a basic model for loss of stress carrying capacity during coalescence. While commonly used in finite element calculations, the GTN model does have some limitations. It assumes a population of spherical voids that are all the same size and dispersed evenly though the matrix. While Tvergaard and Needleman introduced the "q" parameters that can be tuned to account somewhat for void shape effects, these effects are not well captured [8]. Since real materials may have voids that are not spherical and may be distributed differently in different directions, more recent

models have been proposed to capture more of the effects of void shape and distribution ([6],[14]).

The Gologanu-Leblond-Devaux (GLD) model captures the effects of void shape on void growth. The GLD model captures the growth of spheroidal voids, both prolate and oblate ([15],[16],[6]), and retains the same form as the Gurson model, with the addition of a void aspect ratio parameter to capture the void shape effects and a modified yield surface. The GLD model was initially derived for a confocal unit cell assuming axisymmetric loading, and was later extended to an arbitrary loading case [6]. Later Gologanu extended the GLD model to include coalescence in layers or columns when subject to uniaxial loading ([17],[18]).

Pardeon and Hutchinson extended the GLD model to include a coalescence criterion that accounts for the distribution of voids [14]. They introduced a spacing aspect ratio, and proposed a coalescence model for axisymmetric loading that predicts a critical stress at which localization occurs and account for effects of void spacing and void shape.

Perrin and Leblond additionally proposed a modification of the Gurson model accounting for two populations of voids of different sizes and spacings [19]. Garajeu proposed a model that accounts for ellipsoidal shape effects as well, but also includes void distribution effects by identifying a “void rich” zone and assuming that the matrix is continuous between void rich zones [20]. This model however, does not retain the same functional form as the original Gurson model.

A few different models have been proposed for ellipsoidal voids. Madou and Leblond proposed a model for fully ellipsoidal voids, as opposed to the spheroidal voids assumed by the GLD model [21],[22],[23],[24]. This model includes evolution equations for the orientation of the void axis; however, it is significantly more complex than either the GLD or the Gurson models. Ponte Casteneda and Zaidman proposed a model for aligned ellipsoidal voids and allows for void rotation during deformation [25].

While these models account for various void geometries, they all assume that voids are evenly distributed throughout a material, and do not directly account for distributions where voids may be highly aligned in certain areas where characteristic void sizes and spacing may vary throughout a material.

1.3 Literature review of methods of generating representative volume elements from real microstructures

Recent attempts have been made to capture the effects of particle distribution on the stress-strain response of a material. Pinz et. al. developed a methodology for generating statistically equivalent representative volume elements (RVEs) , by generating a statistical description of the distribution of precipitates in a real microstructure [26]. Then statistically equivalent RVEs can be constructed which have the same statistical description as the full microstructure but contain far fewer precipitates for ease of future simulation [26]. Other methods have also been proposed for generating statistically equivalent RVEs including by using two-point correlation functions to describe distributions [27], [28], [29], and in reconstruction of microstructural features through various tools [30], [31], [32].

Gosh generated statistically equivalent RVEs for a composite material by using Voronoi cells to describe the region around each fiber location, allowing the material to be described by a small sample of the full material with edges described by the Voronoi cells [33]. Gosh also employed Voronoi cells to describe the effects of particle distribution by using finite samples of particles, bounded by their Voronoi cells, as a RVE to calibrate a homogenized model similar to the anisotropic GTN model [34].

These current methods of generating RVEs still have limitations for finite element modelling as they still contain a distribution of multiple voids, rather than reducing to a unit cell with a single void. More complicated RVEs are more time consuming and difficult to use as a potential tool for calibrating model parameters to be used in a ballistic simulations, or other more complex, calculation.

CHAPTER 2

CALIBRATION OF GTN MODEL TO AZ31B DNS

2.1 Review of AZ31b Direct Numerical Simulations.

Rolled magnesium is a highly anisotropic material. This comes from two different sources. The first is anisotropy in the HCP crystals, which tend to align with their c-axes in the normal direction which induces plastic anisotropy. The second is anisotropy in the distribution of precipitates, that tend to align in long stringers along the rolling direction as illustrated by Figure 2. These stringers do not affect the plastic response of the material, but rather induce anisotropy in the damage evolution.

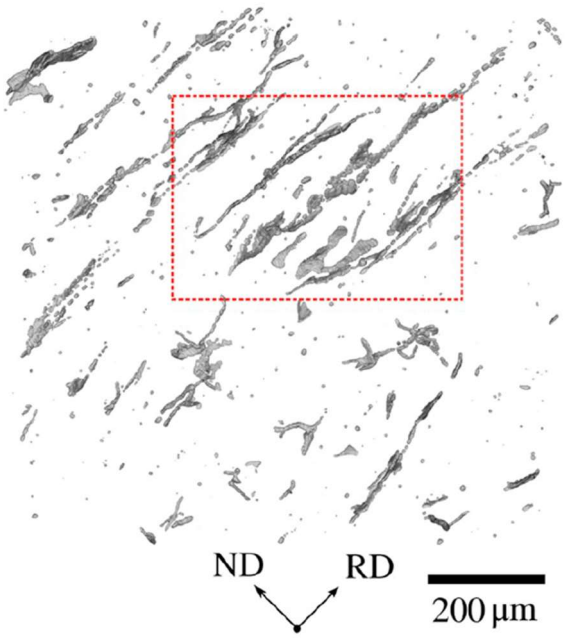

Figure 2: Rolled AZ31B precipitate morphology from a micro-CT scan. Reprinted from [9].

Lloyd [9] conducted uniaxial stress simulations using explicitly modeled realistic second phase particles at various orientations. In these simulations, $\theta_l = 0°$ corresponds with the loading axis aligned with the normal direction (across the second phase particle stringers), and $\theta_l = 90°$ corresponds with the loading axis aligned with the rolling direction (along the second phase particle stringers) as illustrated by Figure 3.

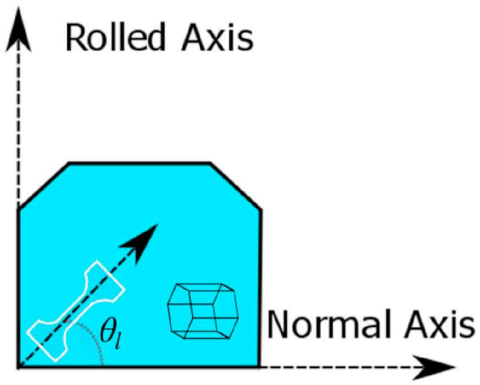

Figure 3: Schematic of simulation loading orientation with respect to rolling and normal directions. Reprinted from [9].

To isolate the effects of the precipitate morphology, an isotropic plasticity model was used, while explicitly modeling the second phase particles. The fracture surfaces for these simulations are shown in Figure 4 from [9].

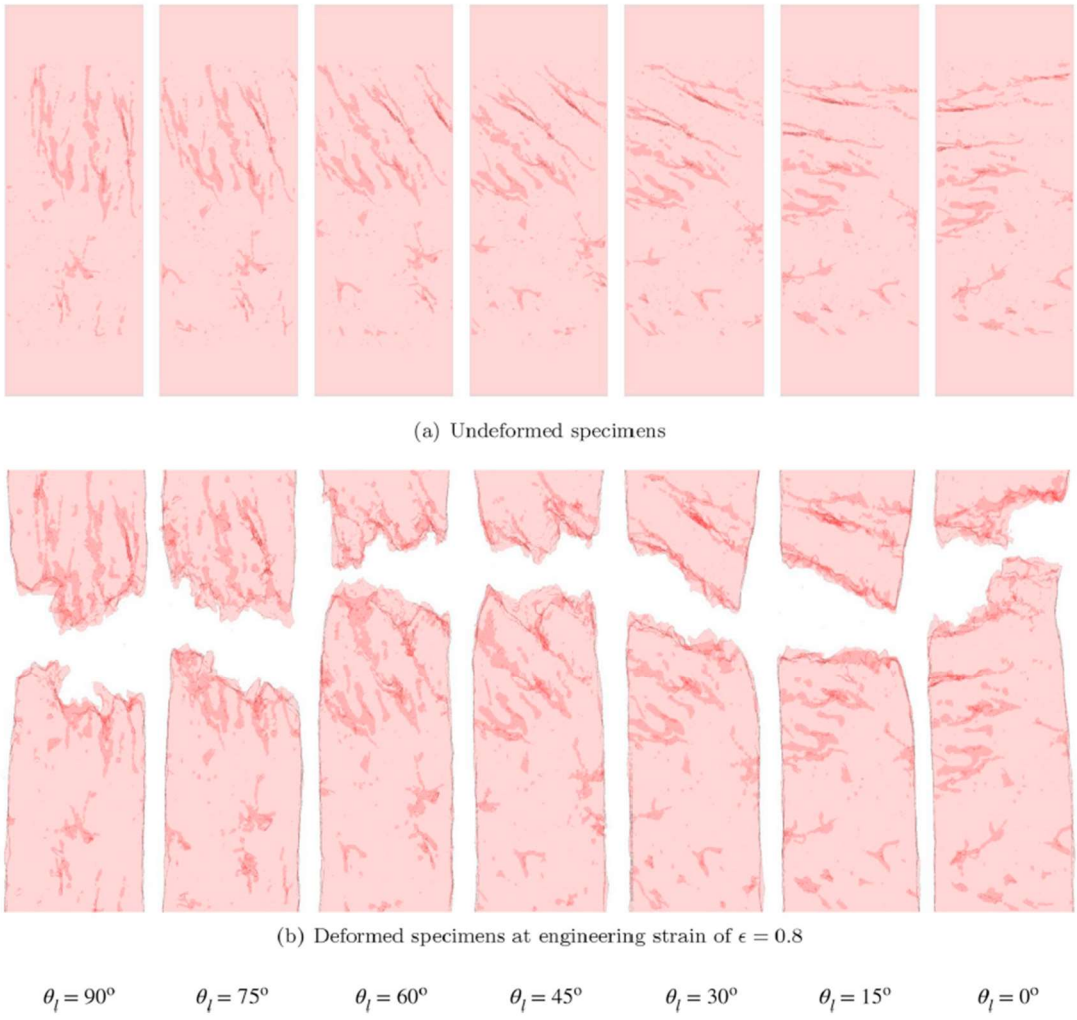

Figure 4: Simulated fracture surfaces under uniaxial stress loading using explicitly modeled precipitates. Reprinted from [9].

While the matrix stress strain behavior for each orientation is the same, damage is nucleated from the particles, and the different particle orientations lead to different fracture behavior. The failure strains are higher for greater θ_l (loading direction more aligned with the RD) where the projected area of the second phase particle stringers is lowest [9].

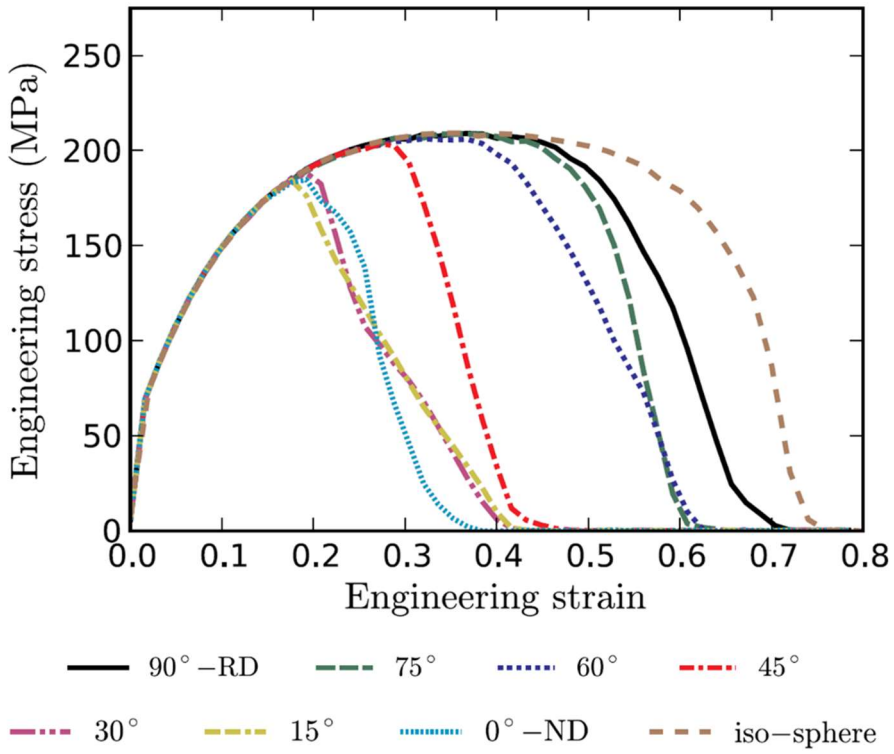

Figure 5: DNS results using isotropic plasticity and realistic, explicitly modeled, precipitates showing orientation dependence. Reprinted from [9].

2.2 Fit of GTN model parameters to AZ31B DNS

While the direct numerical simulations (DNS) with explicitly modeled particles accurately captures the orientation effects, this method is not sustainable for incorporating into a ballistics calculation because of the number of elements required. Instead, a model is needed that captures the effects of orientation without the computational demands of a DNS calculation. To capture these effects, without having to explicitly model the individual particles, a set of GTN model parameters can be fit to each stress strain curve given in Figure 5 for uniaxial stress of an isotropic matrix with realistic microstructure.

2.2.1 GTN model parameter effects

The GTN model yield condition is given by

$$\Phi = \left(\frac{q}{\sigma_y}\right)^2 + 2q_1 f^* \cosh\left(-q_2 \frac{3p}{2\sigma_y}\right) - (1 + q_3 f^{*2}) = 0 \tag{1}$$

where the yield strength is a function of plastic strain.

$$\sigma_y = \sigma_y\left(\bar{\varepsilon}_m^{pl}\right) \tag{2}$$

The hydrostatic pressure, p, is given by

$$p = -\frac{1}{3}\boldsymbol{\sigma} : \mathbf{I} \tag{3}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor. The von Mises stress, q, is given by

$$q = \sqrt{\frac{3}{2}\mathbf{S} : \mathbf{S}} \tag{4}$$

where

$$\mathbf{S} = p\mathbf{I} + \boldsymbol{\sigma}\,. \tag{5}$$

The constants q_1, q_2, and q_3 were selected to be $q_1 = 1.5$, $q_2 = 1$, and $q_3 = q_1^2 = 2.25$.

The void growth rate has two components, the growth rate of existing voids $\dot{f}_{gr}$, and the void nucleation rate $\dot{f}_{nucl}$.

$$\dot{f} = \dot{f}_{gr} + \dot{f}_{nucl} \tag{6}$$

The void nucleation rate is given by

$$\dot{f}_{nucl} = A\dot{\bar{\varepsilon}}^{pl} \tag{7}$$

where A is given by

$$A = \frac{f_N}{s_N\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{\bar{\varepsilon}^{pl} - \varepsilon_N}{s_N}\right)^2\right]. \tag{8}$$

The nucleation strain has a normal distribution with a mean ε_N and standard deviation s_N. Voids are nucleated in tension with a volume fraction of f_N. Figure 6 shows possible effects of precipitate orientation on the void nucleation function. The peak stress is reached at a lower strain for the 0° (Normal Direction) orientation, so voids likely begin nucleating at a lower strain. Figure 6 illustrates how the void nucleation parameters may vary with orientation.

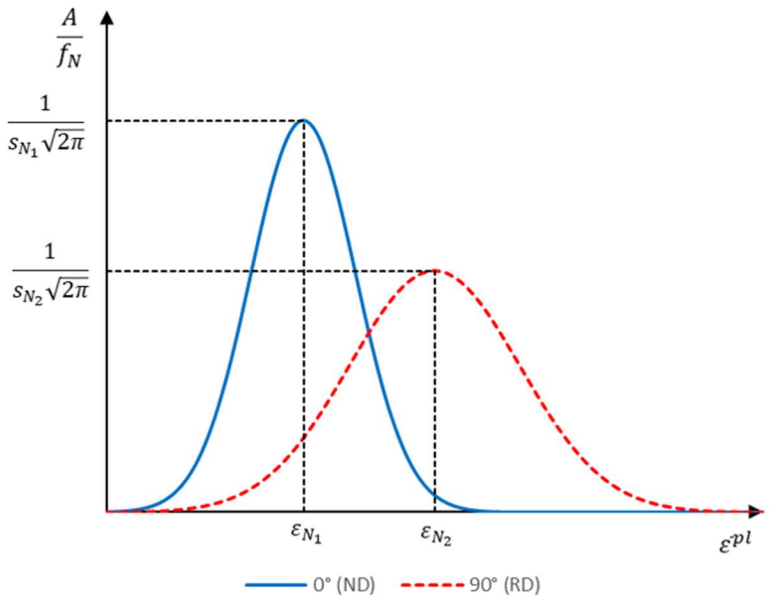

Figure 6: Proposed dependence of void nucleation function of precipitate orientation

Coalescence is governed by two porosities: critical porosity, f_c, and failure porosity, f_F. The function f^* models the loss of stress carrying capacity during void coalescence between f_c and f_F.

$$f^* = \begin{cases} f & \text{if } f \le f_c \\ f_c + \dfrac{\overline{f_F} - f_c}{f_F - f_c}\left(f - f_c\right) & \text{if } f_c < f < f_{\mathrm{F}} \\ \overline{f_F} & \text{if } f \ge f_{\mathrm{F}} \end{cases} \tag{9}$$

$\overline{f_F}$ is a constant determined by the parameters q_1, q_2, and q_3.

$$\overline{f_F} = \frac{q_1 + \sqrt{q_1^2 - q_3}}{q_3} \tag{10}$$

In an ABAQUS, an element is considered failed and is deleted when $f \ge f_{\mathrm{F}}$.

If a material has a higher critical porosity, it will reach a higher strain before the stress starts to drop. For the same critical porosity, a higher failure porosity means the slope between the initial stress drop and total failure (zero stress) will be less steep and the failure strain will be higher. Figure 7 shows how the void coalesce parameters may vary with orientation.

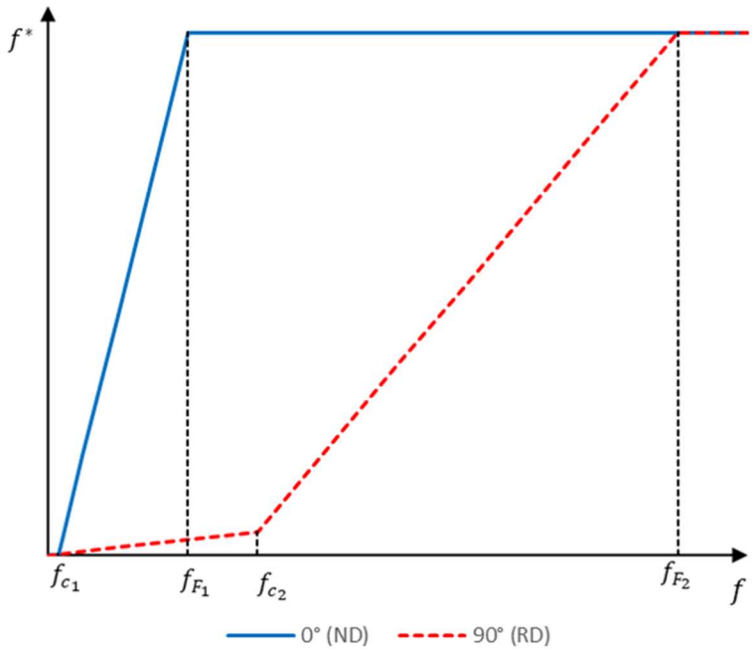

Figure 7: Proposed dependence of coalescence function on precipitate orientation

2.3 Fitting to DNS stress-strain curve

For each precipitate orientation, a set of GTN model parameters was fit to the corresponding engineering stress-strain curve in Figure 5. Single element, uniaxial stress, simulations were run in ABAQUS at $\dot{\varepsilon} = 10^4 s^{-1}$ using the same isotropic plasticity model as in Lloyd 2019 [9], given in Equation 11, and using the GTN damage model.

$$\sigma_s = \left\{ \sigma_{0,s} + \sigma_s^{\infty} \left[1 - \exp\left(\frac{-h_{1,s}}{\sigma_s^{\infty}} \bar{\varepsilon}^{pl} \right) \right] \right\} \tag{11}$$

The values of the parameters in Equation 11 are given in Table 1.

Table 1: Isotropic strain hardening model parameters

Parameter	Value	Units
$\sigma_{0,s}$	40	MPa
σ_s^{∞}	180	MPa
$h_{1,s}$	2500	MPa

For each orientation, f_F, ε_N, and S_N were independently calibrated to fit the strain at peak stress and strain to failure of the DNS calculations with realistic precipitates. f_c and f_N were fixed for all orientations.

Table 2: Calibrated GTN model parameters as a function of orientation

θ_l	**0**	**15**	**30**	**45**	**60**	**75**	**90**
f_F	*0.08*	*0.11*	*0.108*	0.075	*0.07*	0.048	*0.06*
f_c	*0.008*	*0.008*	*0.008*	0.008	*0.008*	0.008	*0.008*
ε_N	*0.18*	*0.14*	*0.15*	0.25	*0.4*	0.5	*0.5*
s_N	*0.05*	*0.025*	*0.015*	0.045	*0.15*	0.2	*0.2*
f_N	*0.04*	*0.04*	*0.04*	0.04	*0.04*	0.04	*0.04*

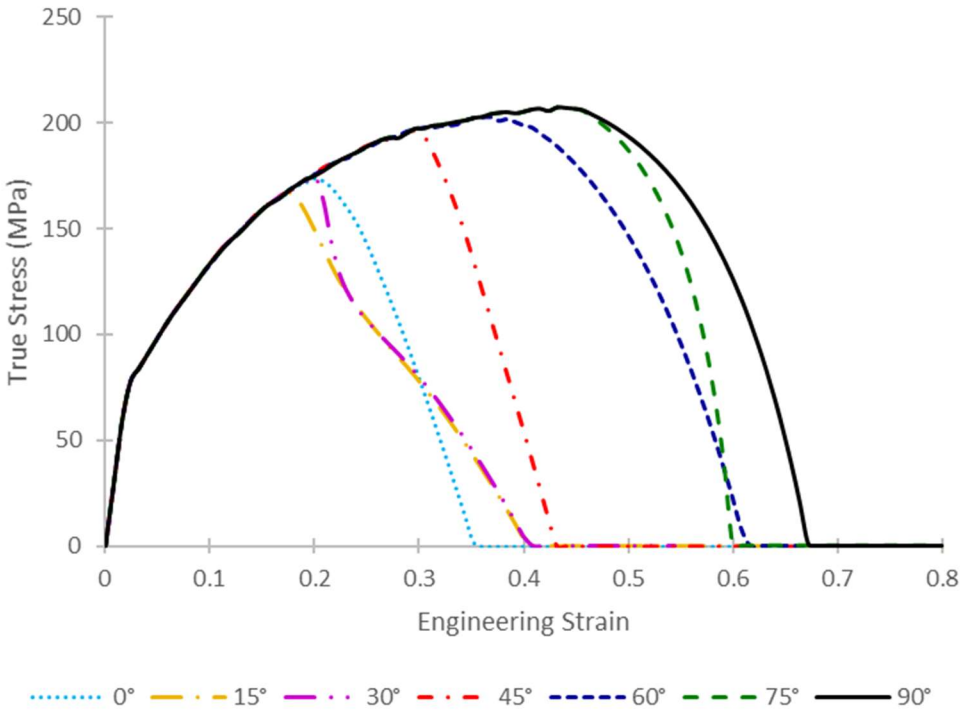

Figure 8: Simulation results using isotropic plasticity and homogenized damage, with f_F, ε_N, and S_N parameters fit to the results of Figure 5

Figure 8 shows the stress strain responses for each orientation using the calibrated GTN parameters. The strain at peak stress and strain to failure are reproduced accurately for each orientation. Although there are some slight discrepancies in the exact slopes between the peak stress and total failure, the overall trends between orientations are accurately reproduced. Figure 9 compares the strain at peak stress and strain to failure of the calibrated GTN stress strain curves (Figure 8) to those of the DNS using isotropic plasticity and realistic precipitates (Figure 5).

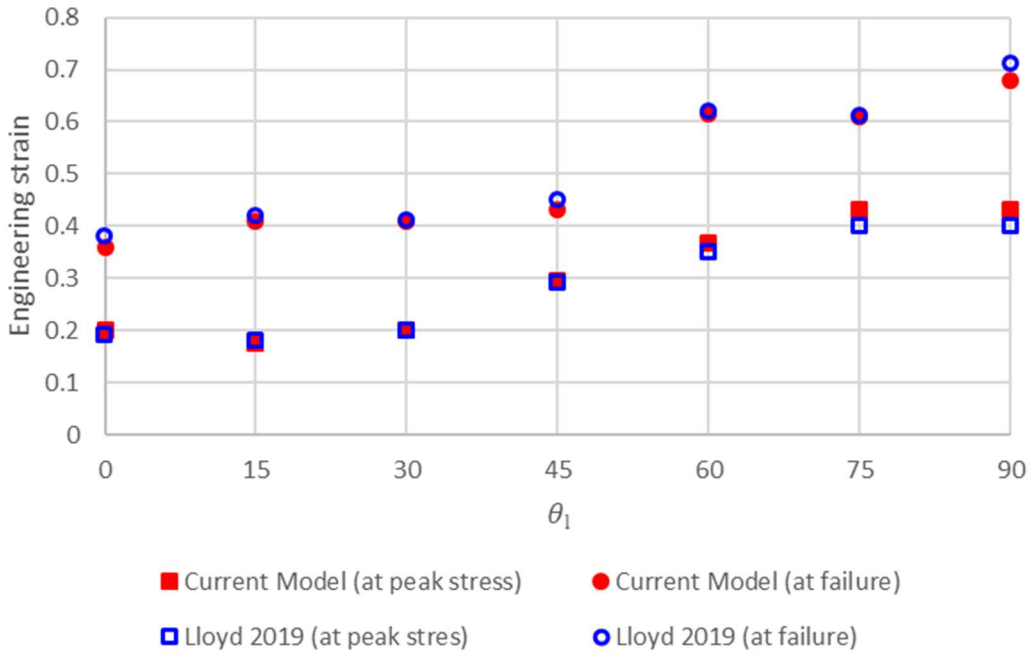

Figure 9: Comparison of calibrated GTN model to DNS simulations with realistic precipitates

The error between the strains of the calibrated model and the strains of the direct numerical simulations is small for both the strain at peak stress and strain-to-failure. The errors in the strains for each material orientation are tabulated in Table 3.

Table 3: Error between strains of calibrated model and DNS

	Engineering Strain at Peak Stress			Engineering Strain at Failure		
θ_l	Current Model	Lloyd 2019 [9]	Error (%)	Current Model	Lloyd 2019 [9]	Error (%)
0°	0.20	0.19	5.3	0.36	0.38	-5.2
15°	0.18	0.18	-2.2	0.41	0.42	-2.8
30°	0.20	0.20	0.0	0.41	0.41	-0.4
45°	0.30	0.29	2.1	0.43	0.45	-4.0
60°	0.37	0.35	5.1	0.62	0.62	-0.6
75°	0.43	0.40	8.0	0.61	0.61	-0.1
90°	0.43	0.40	8.0	0.68	0.71	-4.2

2.3.1 Generalizing the anisotropic damage model

The anisotropic GTN parameters outlined above are applicable for the special case in which the uniaxial tensile loading axis is fixed, i.e. $\boldsymbol{e}_Z = \boldsymbol{e}_{ND} cos\theta_l + \boldsymbol{e}_{RD} sin\theta_l$, with θ_l being a fixed loading axis. For any fixed loading axis, the appropriate GTN parameters for AZ31B may be obtained via interpolation of values in Table 2. The question becomes, what GTN parameters are appropriate in the case of a tensile loading axis that changes during deformation. For example, consider a forming process in which the loading axis is initially along the rolling direction ($\boldsymbol{e}_Z = \boldsymbol{e}_{RD}$ with $\theta_l = 90°$) until the accumulated plastic strain is 50%, then the tensile loading axis is abruptly changed to

align with the normal direction ($\boldsymbol{e}_Z = \boldsymbol{e}_{ND}$ with $\theta_l = 0°$). In this case, is it appropriate to abruptly convert the GTN parameters from the $\theta_l = 90°$ values to the $\theta_l = 0°$ values? With respect to the nucleation parameters, this can be argued to be adequately representative of the actual physics. In this case, the revectoring of the loading axis from $\theta_l = 90°$ to $\theta_l = 0°$ would simply result in accelerated void nucleation rates.

On the other hand, there are serious logic issues with this approach in handling the orientation dependence of the coalescence porosity f_c and the failure porosity f_F. Consider for example, a forming process in which the material is first loaded in tension along $\theta_l = 15°$ until the porosity reaches $f = 6\%$. At this point, the material has roughly 40% of its load carrying capacity since $f_F = 11\%$, i.e. $1 - f_c/f_F \sim 40\%$. Subsequently, assume the material is unloaded, then reloaded along the rolling direction ($\theta_l = 90°$) with an infinitesimally small tensile stress, e.g. 1 Pa. Since $f = f_c = 6\%$, prior to this reloading, the material now has no load carrying capacity. Clearly this is a logical flaw in this simple approach of selecting orientation dependent values of the coalescence porosity f_c and the failure porosity f_F based only on the current tensile axis. To overcome this flaw, it is necessary to introduce a parameter to capture the history dependence of the loading axis, which is proposed below.

For any general stress state $\boldsymbol{\sigma}$, we denote the eigenvectors of the stress tensor as $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$, which correspond to the three principal stresses σ_1, σ_2, and σ_3, respectively. Here we adopt the typical ordering convention of the principal stresses such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$. Thus, for the special case of uniaxial tension discussed above, the loading angle can be equivalently defined as the projection of the eigenvector associated

with the largest principal stress onto the rolling direction unit vector, i.e. $\boldsymbol{e}_1 \cdot \boldsymbol{e}_{RD} = sin\theta_l$. We utilize this same definition, to define the dominant tensile loading axis under any arbitrary general loading state. Even for three-dimensional stress states, $\boldsymbol{e}_1$ and thus θ_l are constant for any proportional loading, e.g. $\boldsymbol{\sigma(t)} = \beta(t)\,\boldsymbol{\sigma_{ref}}$, where $\boldsymbol{\beta(t)}$ is any time-varying scalar that proportionally amplifies the stress tensor from a reference time denoted here as $\boldsymbol{\sigma_{ref}}$. For non-proportional loading, $\boldsymbol{e}_1$ and thus θ_l are varying with loading, and a history dependent effective loading angle (denoted here as $\bar{\theta}_l$) is required. Unfortunately, the authors are unaware of any non-proportional loading experiments or DNS simulations of AZ31B that could be utilized to inform the construction of $\bar{\theta}_l$. For lack of any better formulation, we propose the following expression to compute the effective loading angle

$$\int_{f_0}^{f^*} f^{*\alpha-1}\mathbf{e}_1 \cdot \mathbf{e}_{RD} df^* = \alpha^{-1}(f^{*\alpha} - f_0^{\alpha}))sin\overline{\theta_l} \tag{12}$$

where $\alpha \neq 0$ is a scalar exponent that can be tuned to non-proportional loading data, if available. In particular, α can be selected in such a way that $\bar{\theta}_l$ is uniformly depends on the history of f^*, i.e. $\alpha = 1$. Alternatively, $\alpha < 1$ can be selected such that θ_l and $\bar{\theta}_l$ deviate most starkly at large f^*. Note that the above expression is constructed such that $\bar{\theta}_l = \theta_l$ for any proportional loading case. For any time increment, we propose using $\bar{\theta}_l$ to update the anisotropic values of the coalescence porosity f_c and the failure porosity f_F. For AZ31B, this can be achieved via interpolation of the values in Table 2. Note that this procedure will avoid the logic issues discussed above, because $\bar{\theta}_l$ is a smoothly varying function for any arbitrary loading path. On the other hand, as discussed earlier,

we recommend selecting nucleation parameters (f_N, ϵ_N, s_N) based on the instantaneous value of the dominant loading angle θ_l. In the following section, we will carry out some impact simulations under more general loading states. Even though these cases are not necessarily proportional loading, they are special cases in which the dominant tensile direction, i.e. $\boldsymbol{e}_1$, is known a priori with respect to the impact velocity vector.

2.4 Ballistic performance as a function of material orientation

2.4.1 Spherical impactor

To capture the effects of orientation on the ballistic response of magnesium, for each orientation the corresponding GTN model was used in ABAQUS to simulate a magnesium plate impacted by a steel sphere at a velocity of 300 m/s. The boundary conditions of these axisymmetric finite element simulations are shown in Figure 10. The steel projectile had a diameter of 3.5 mm and the magnesium plate had a thickness of 3.5 mm.

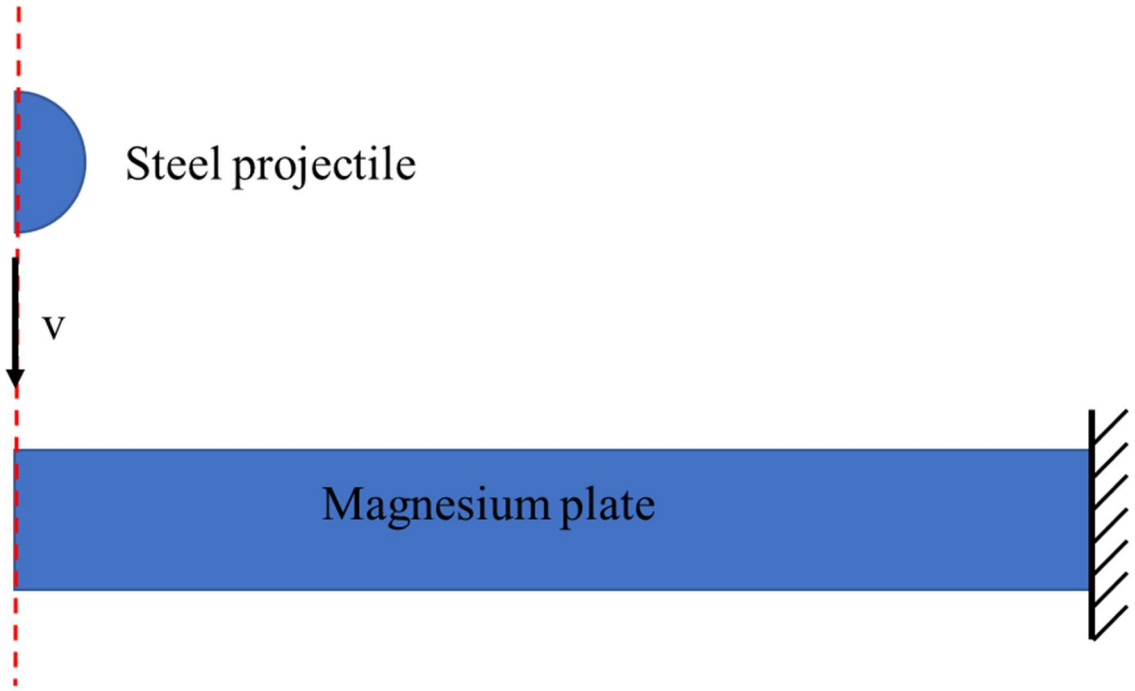

Figure 10: Diagram of ballistics problem

The mesh size was chosen such that the elements had a similar size (0.5mm) to the dimensions of the simulation domain used in the Lloyd et al. DNS simulations. In doing so, we can ensure that energy dissipated in the complete failure of particular volume of a material is approximately the same for both DNS and the homogenized GTN model. In particular, this choice of element size ensures that the energy dissipated in macroscopic crack formation in the homogenized simulation is never less than the corresponding energy dissipated in the DNS simulations. This energy argument does not necessarily hold for significantly finer meshes, which could lead to an artificial mesh-dependence, in which the material is artificially more brittle at finer mesh sizes.

After impact, damage accumulates in the elements on the back face, which are eventually deleted for the weaker orientations. Elements are deleted when the void volume fraction reaches the failure porosity, f_F. The damaged plates at 0° and 90° are shown in Figure 11.

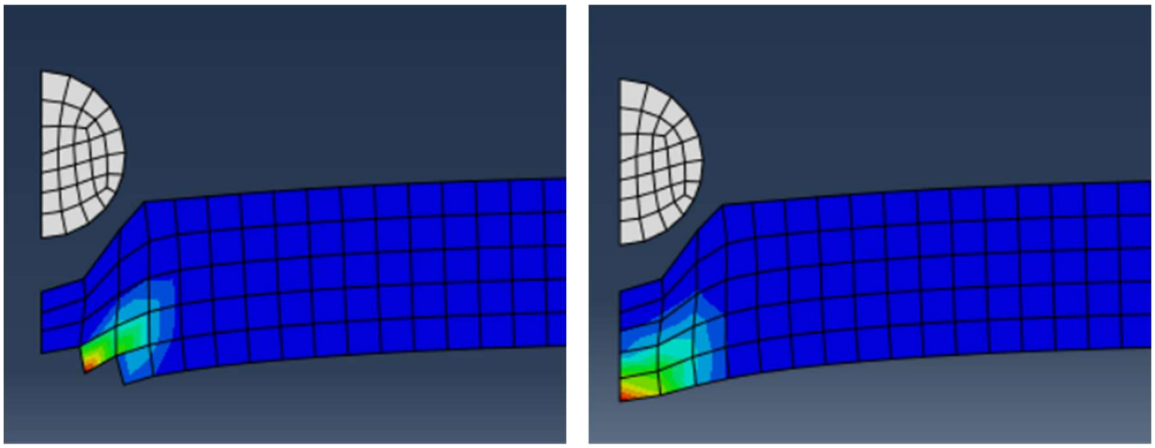

Figure 11: (a) Deformed plate using 0° material model showing void volume fraction (b) Deformed plate using 90° material model showing void volume fraction

Figure 12 shows the free surface velocity taken at the center node on the back face of the magnesium plate. The free surface velocity is plotted as constant after the element has been deleted. The weaker material orientations (0°-30°) reach the highest free-surface velocity, while the stronger orientations reach a lower velocity.

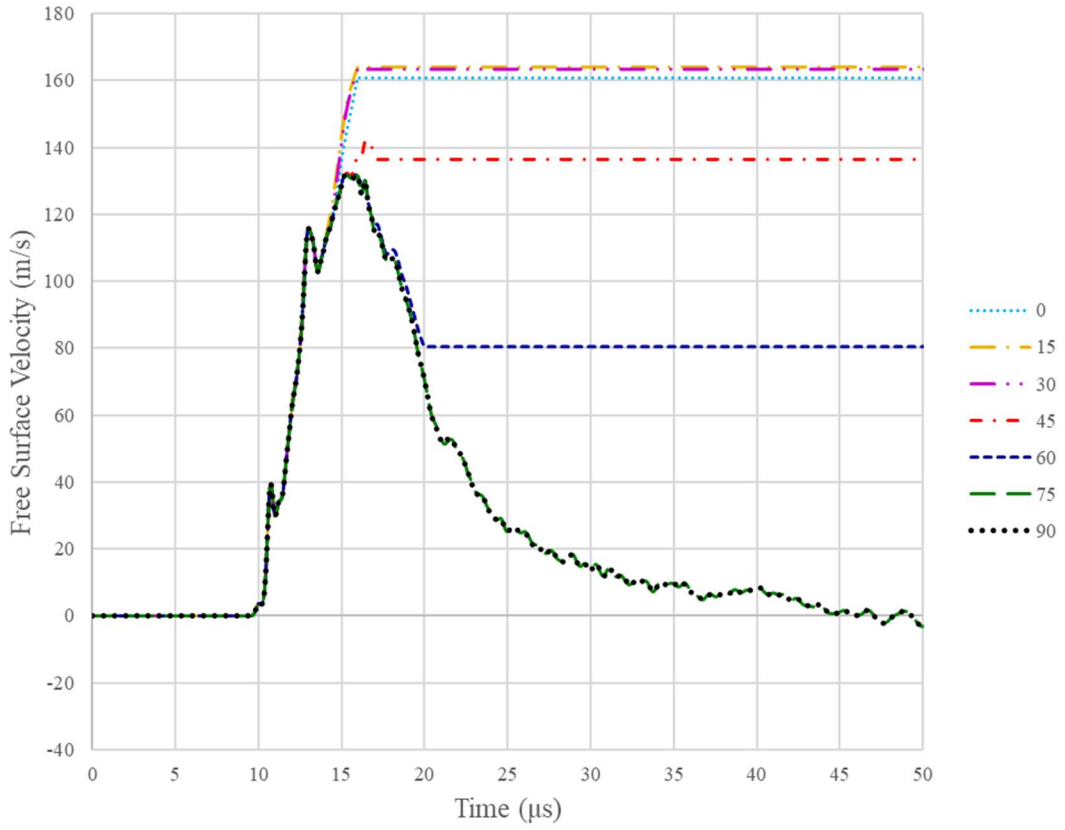

Figure 12: Free surface velocity at the center of the plate for each orientation subjected to impact by steel sphere

Figure 13 plots the number of deleted elements as a function of time for each material model. The strongest orientations never have an element accumulate enough damage to be deleted. In general, the weaker orientations have more deleted elements. This implies that the orientations showing no element deletion (75° and 90°) are better for armor applications since they would not result in any flaking off of material on the back face of a plate of armor.

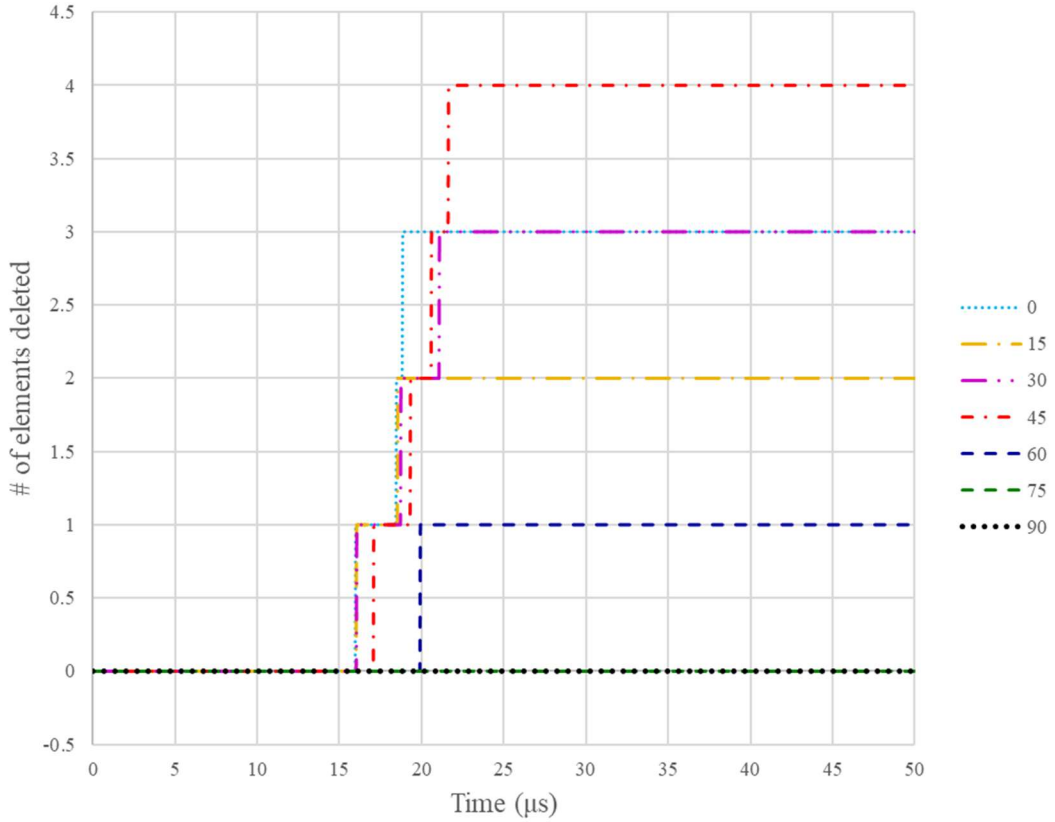

Figure 13: Number of elements deleted for different material orientations subjected to impact by steel sphere

2.4.2 Plate impact

Similar axisymmetric finite element simulations to those in the previous section were also conducted for the case of a steel plate impacting a magnesium plate. For these simulations a circular steel plate of 3.5mm thick impacts a 7mm thick magnesium plate at a velocity of 300 m/s. A diagram of the plate impact problem is shown in Figure 14.

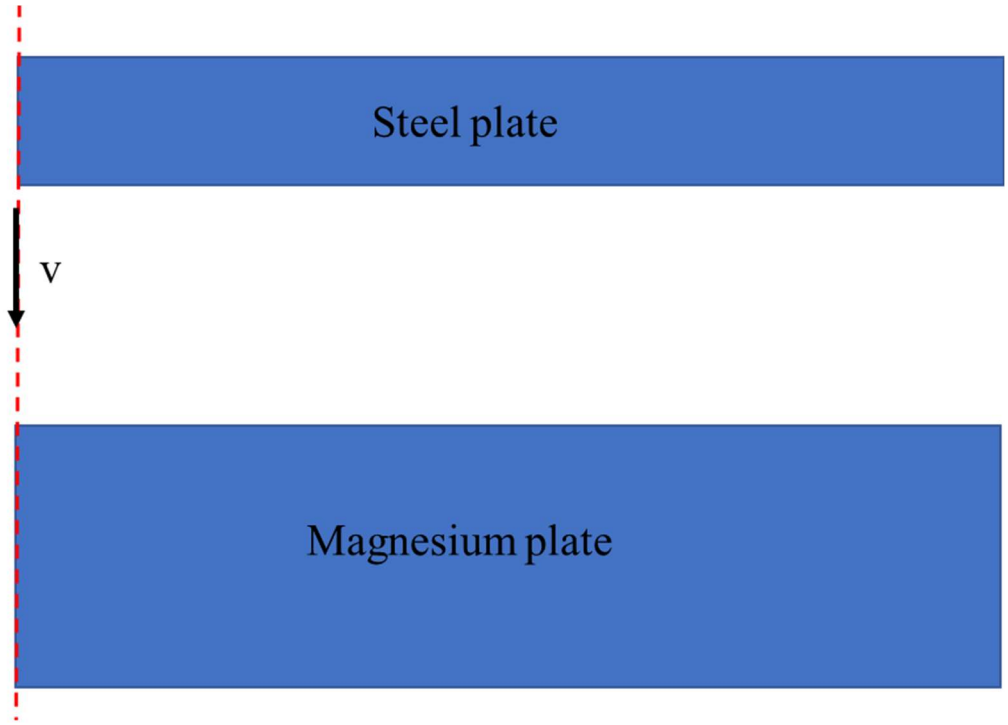

Figure 14: Diagram of plate impact problem

Figure 15 shows the damaged magnesium plate for the 0° material model. Damage develops near the center line of the plate until the elements fail and the back face of the magnesium plate separates. There are some edge effects seen at the contact between the two plates that are due to the coarseness of the mesh. The damage pattern is similar for the other material orientations.

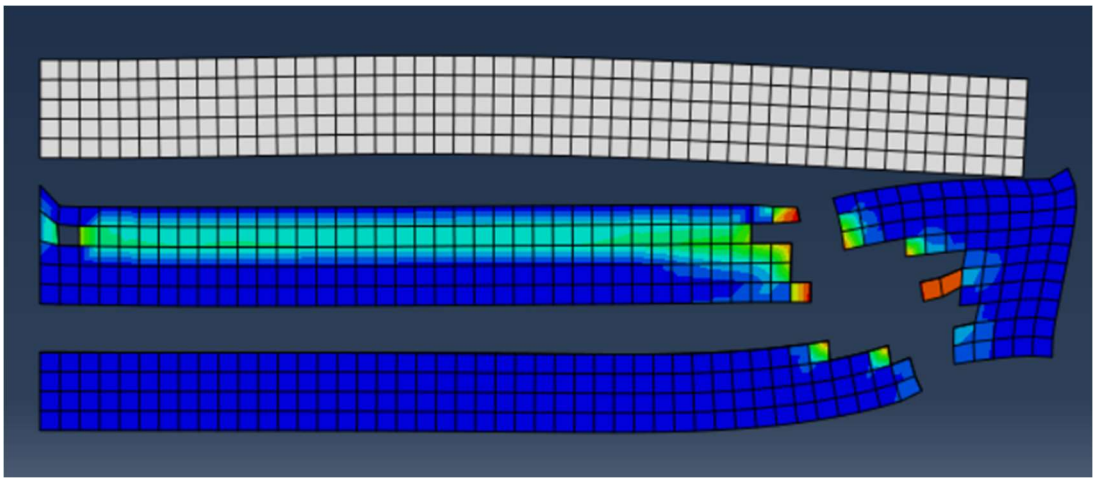

Figure 15: Deformed plate using 0° material model showing spall failure. Color scale denotes void volume fraction

The free surface velocities of the center node on the back face of the magnesium plate is plotted in Figure 16. The responses for each different material model are the same up until the point where the plate separates. The separated piece of the back face is ejected from the plate at a higher velocity that the initial impact due to the difference in mass of the two plates.

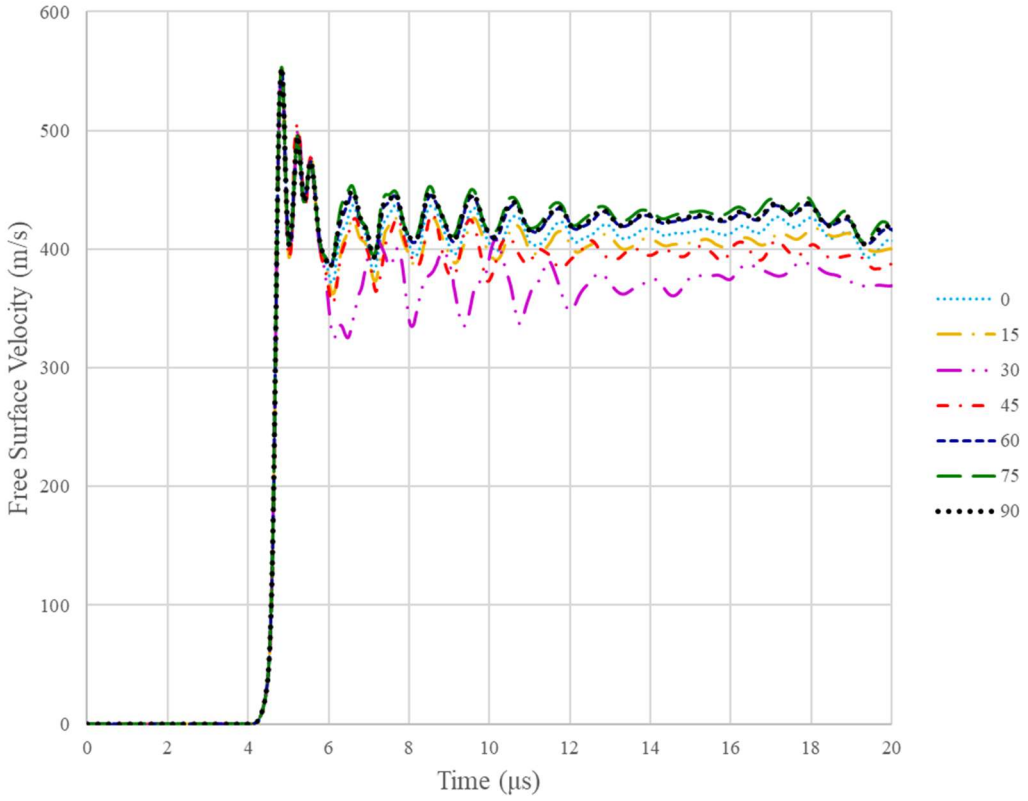

Figure 16: Free surface velocity at the center of the plate for each orientation subjected to impact by steel plate

After separation, the final velocities of the back face were the highest for the 75° and 90° models. The lowest final velocity occurred for the 30° orientation, which also experienced the most damage. This is because for that orientation the back face does not fully separate from the rest of the plate. Overall, the rest of the orientations have similar velocities.

The spall strength is related to the pull-back velocity in a plate impact simulation by Equation 13, where ΔU_{fs} is the pull-back velocity.

$$\Sigma^* = \frac{1}{2}\rho_0 C_0 \Delta U_{fs} \tag{13}$$

The spall strengths for each orientation are listed in Table 4 for the density $\rho_0 = 1770 kg/m^3$ and wave speed $C_0 = 4540 m/s$. The predicted spall strength is slightly lower along the rolling direction (90°) than the normal direction (0°), with the highest spall strengths occurring between $\theta_l = 15°$ and $\theta_l = 45°$.

Table 4: Spall strength as a function of material orientation

θ_l	Σ^* (MPa)
0°	615
15°	643
30°	643
45°	643
60°	603
75°	602
90°	602

The strain rate of the magnesium plate during impact is given by Equation 14, where Δt is the pullback velocity history duration from the peak to the local minimum [10]. The strain rate for this plate impact problem was approximately $8.6 \text{x} 10^4 s^{-1}$.

$$\dot{\varepsilon} = \frac{1}{2C_0}\frac{\Delta U_{fs}}{\Delta t} \tag{14}$$

As shown in the results of Mallick et. al. [10] in Figure 47, spall strength increases with strain rate. The results of Figure 47 range from a spall strength of 0.4 GPa at a strain rate of $10^4 s^{-1}$ to a spall strength of about 1.8 GPa at a strain rate of $10^6 s^{-1}$. The blue dashed line shows the trend in AZ31B. The spall strengths in Table 4 lie approximately along the blue line.

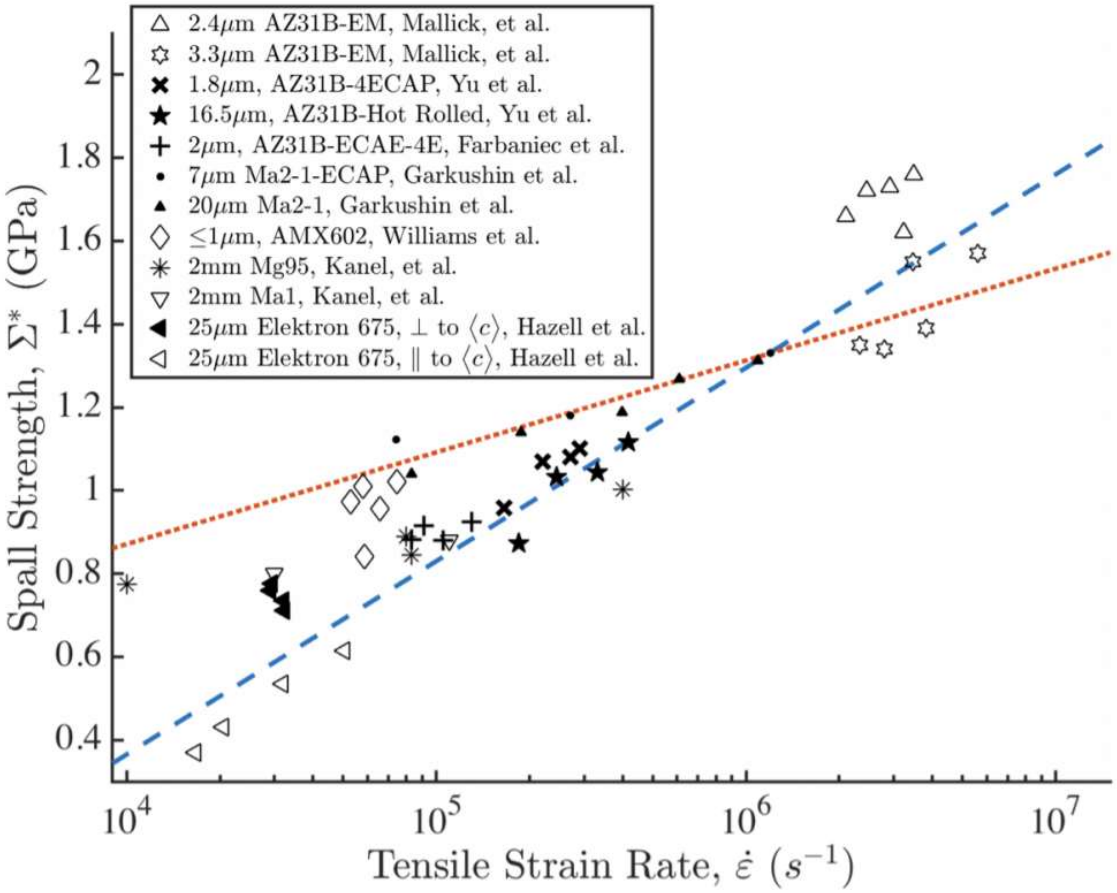

Figure 17: Spall strength of polycrystalline pure and alloyed magnesium. The blue line denotes the trend in AZ31b, while the red line denotes the trend in Ma2-1. Reprinted from [10].

Figure 18 shows the number of deleted elements as a function of time for each orientation. The plates experience a rapid increase in the number of deleted elements when failure occurs at the spall plane, followed by a slower increase as elements continue to accumulate enough damage to be deleted.

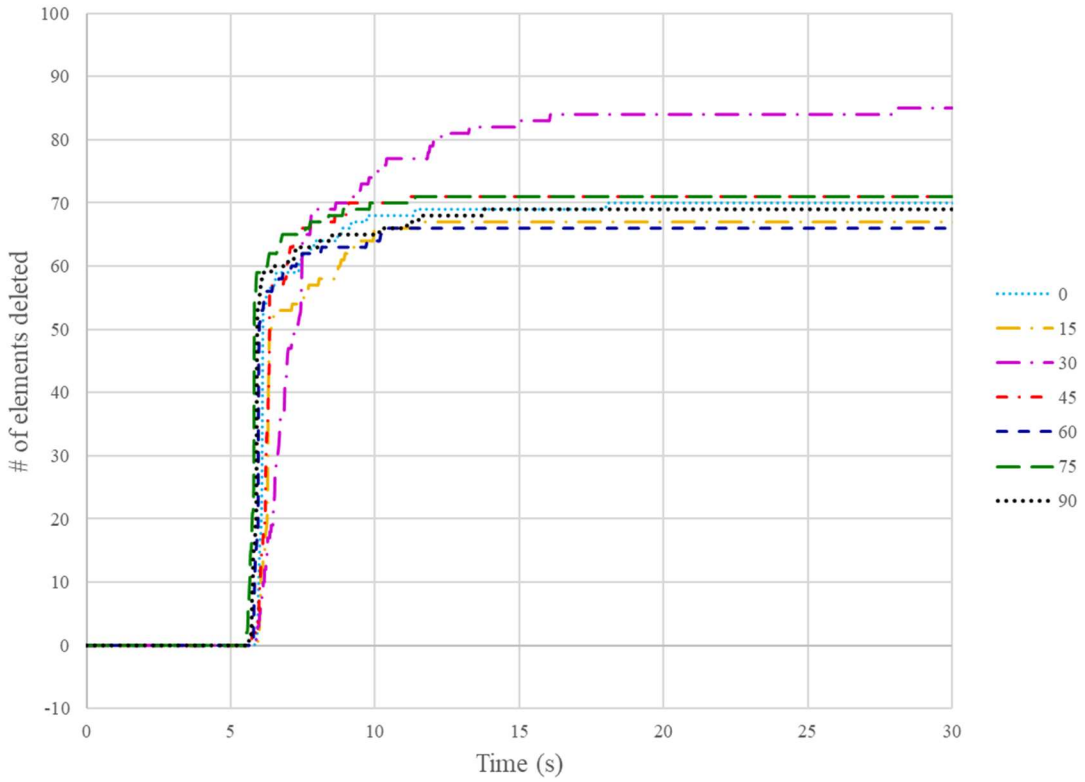

Figure 18: Number of elements deleted for different material orientations subjected to impact by steel plate

CHAPTER 3

CONVERSION OF MICROSTRUCTURE TO REPRESENTATIVE UNIT CELL

In the previous section, the effects of anisotropic particles were captured by calibrating GTN model parameters to stress-strain curves from direct numerical simulations with explicitly modeled particles. In this section, a method for determining a representative unit cell directly from a real particle distribution will be outlined. This will enable skipping the time-intensive step of directly modeling the particles individually and instead using results from the representative unit cell to calibrate model parameters to.

3.1 Fit of ellipsoid distribution to real microstructure

To convert a real microstructure into a representative, axisymmetric, unit cell with an ellipsoidal void, the microstructure needs to be reduced to four representative dimensions, L_z, L_r, R_z, and R_z as shown in Figure 19.

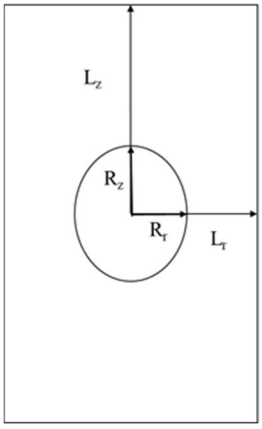

Figure 19: Axisymmetric unit cell geometry

A MATLAB algorithm was created to fit ellipsoidal shapes to real microstructure data and extract the spatial and size distribution of those ellipsoidal shapes. 2D images of a real microstructure are obtained from μ-CT scans of a material. These binary image slices showing the second phase particles present in the matrix material are then stacked together to reconstruct the 3D microstructure as illustrated in Figure 20.

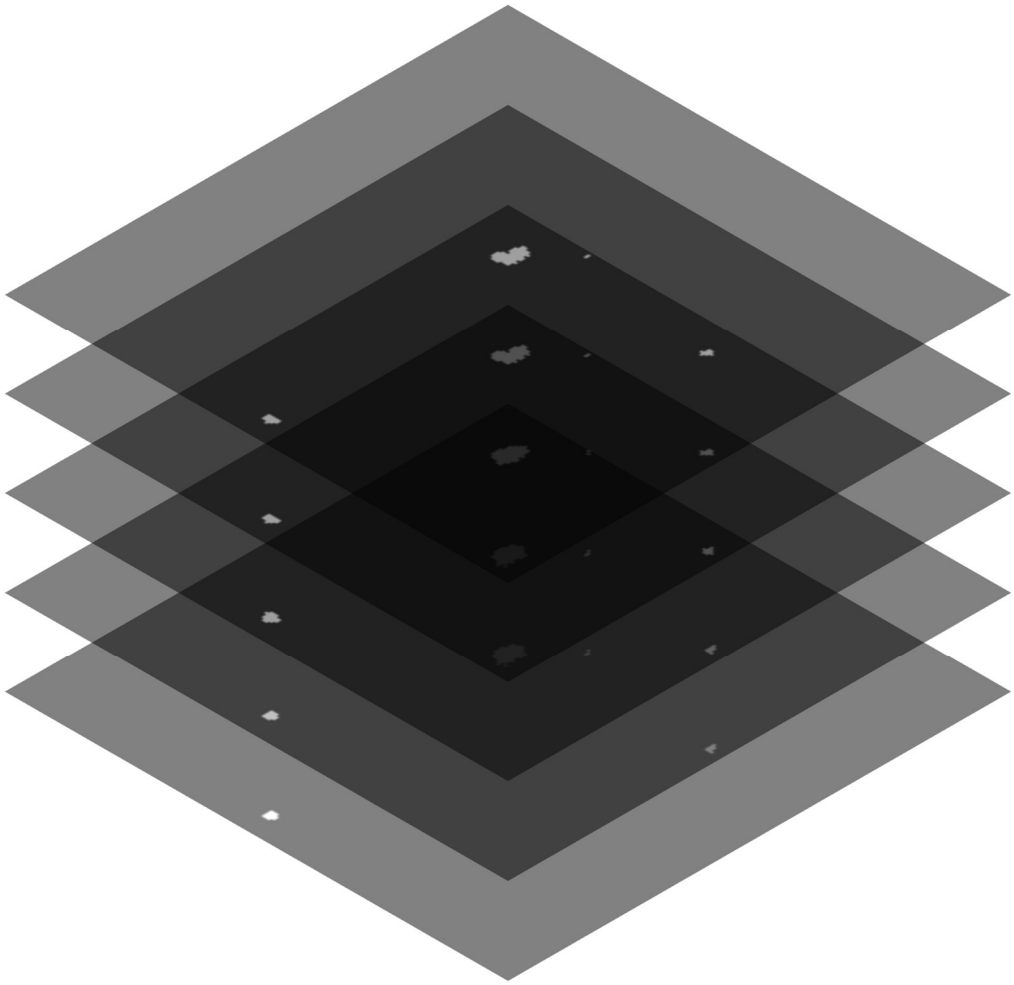

Figure 20: Reconstruction of 3D microstructure from μ-CT binary slices

The MATLAB code uses a function that will find all connected regions in a binary volume, and determine the centroid and axis lengths of the ellipsoid which has the same centroid and second moment as the original particle as illustrated in Figure 21. Additionally, the MATLAB algorithm also calculates the orientation of each individual ellipsoid with respect to the material axes.

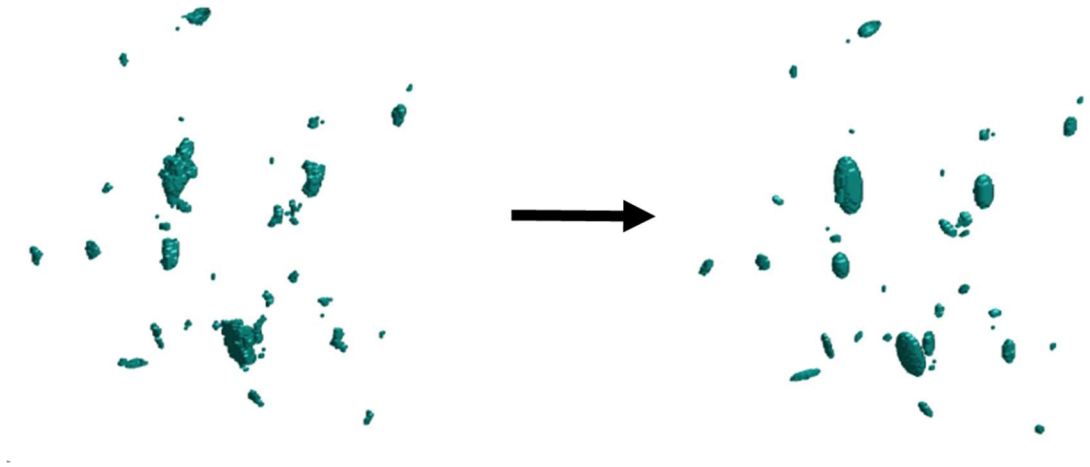

Figure 21: Fit of ellipsoids to real second phase particles

Each particle is defined by a connected region of voxels in the binary volume. Pixels are considered connected only if their faces are in contact as shown in Figure 22. If two voxels are connected by only an edge or a corner, they are considered separate connected regions representing separate particles.

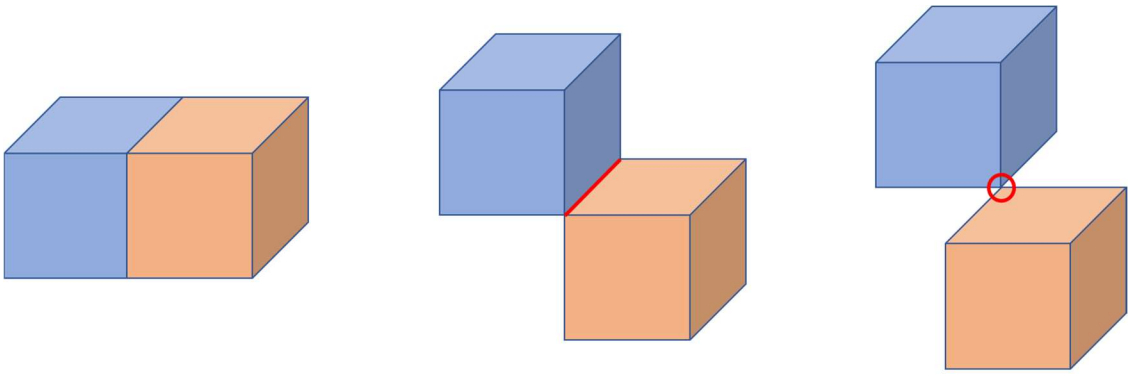

Figure 22: Voxel connectivity conditions

The MATLAB function regionprops3 converts each arbitrary connected binary region to an ellipsoid with the same normalized second central moment as the original region [35]. The ellipsoid orientations and axis lengths are determined from the covariance matrix of the binary connected region. In 2D, the central moments, μ , are given by

$$\mu_{pq} = \sum_{x}\sum_{y}(x-\bar{x})^{p}(y-\bar{y})^{q}b(x,y) \tag{15}$$

and $b(x,y)$ is equal to 1 if the voxel is part of the region, and 0 if not [36]. The coordinates $\{\bar{x},\bar{y}\}$ represent the centroid of the connected binary region. The covariance of the image, I, is given by Equation 16 [37].

$$\text{cov}[I(x,y)] = \frac{1}{\mu_{00}}\begin{pmatrix}\mu_{20} & \mu_{11}\\ \mu_{11} & \mu_{02}\end{pmatrix} \tag{16}$$

The eigenvectors of the covariance matrix correspond to the directions of the major and minor axes of the ellipse, and the eigenvalues represent the squared length of the major and minor axes. The three axes of an ellipsoid representing a 3D binary region are extracted in a similar manner, but there will instead be a set of three eigenvectors and eigenvalues corresponding to the three axes' directions and lengths.

Figure 23 and Figure 24 shows a single particle converted to a representative ellipsoid.

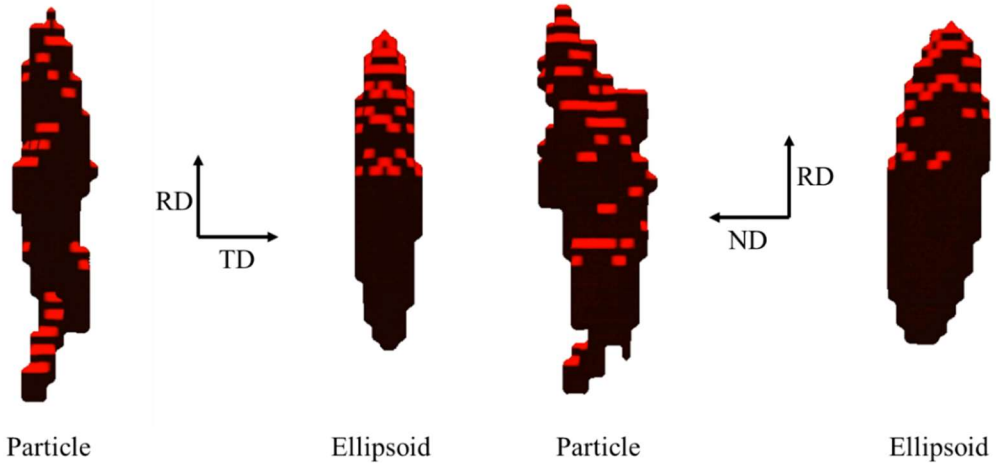

Figure 23: Front and right view of single particle and corresponding ellipsoid

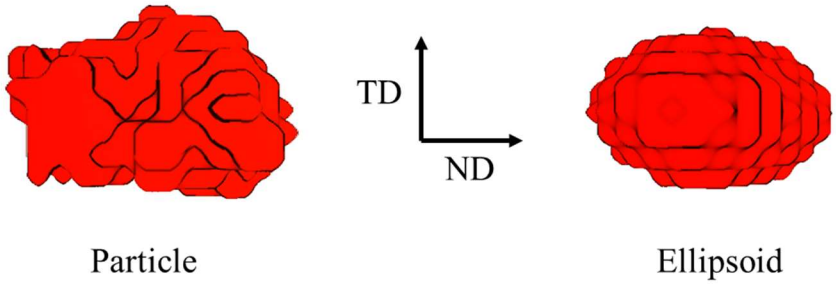

Figure 24: Top view of single particle and corresponding ellipsoid

As shown in Figure 25, Figure 26, and Figure 27 the MATLAB algorithm accurately reproduces the approximate shape, size, and orientation of the real second phase particles. Figure 28 shows a cross section of the original 2D binary volume, and a cross section of the binary volume of the corresponding ellipsoid distribution. The algorithm maintains the exact number of individual particles, since each connected region is assigned an ellipsoid, and the new ellipsoidal microstructure has approximately the same total porosity as the original microstructure.

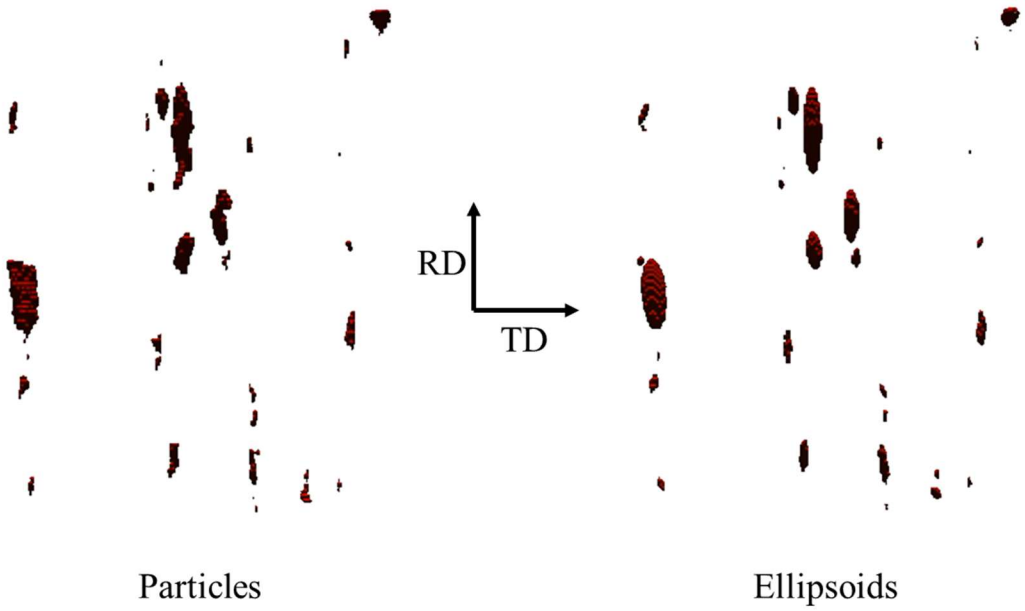

Figure 25: Front view of particles and corresponding ellipsoids

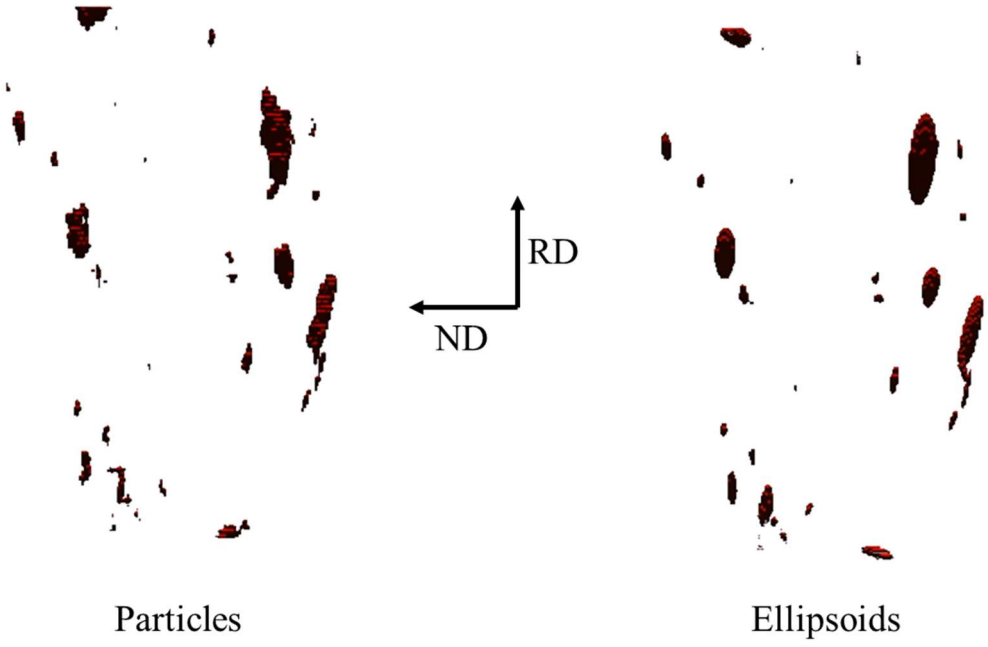

Figure 26: Right view of particles and corresponding ellipsoids

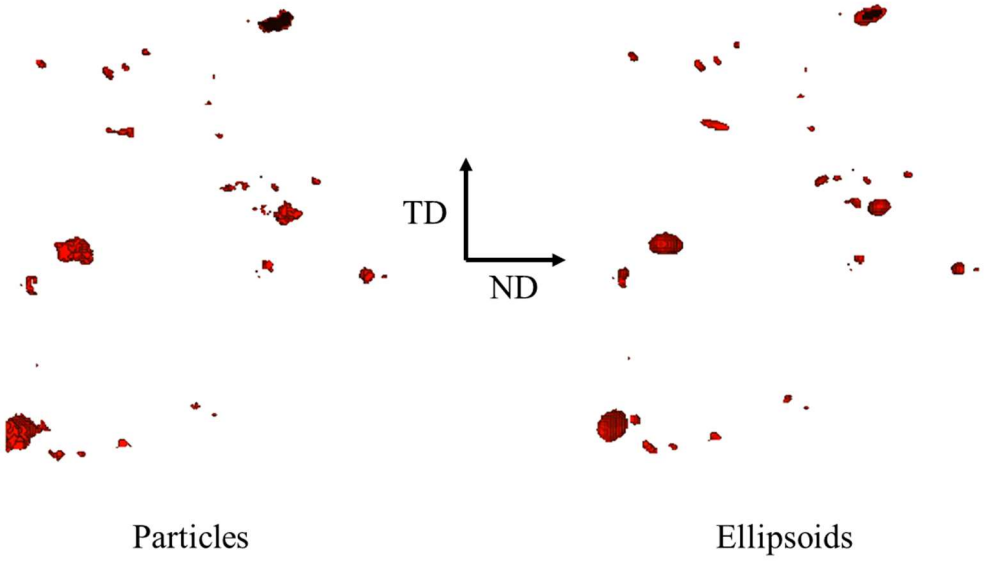

Figure 27: Top view of particles and corresponding ellipsoids

Figure 28: Cross section showing particle and ellipsoid areas at z = 160 voxels

To determine the spacing distribution of the second phase particles, a simple nearest neighbor algorithm was implemented. For each ellipse, the MATLAB algorithm finds the neighboring ellipse which has the shortest radial distance between the ellipse centers.

For a particular ellipse, i, the distances, $r(j)$,between its centroid and the centroid of all other ellipses is calculated.

$$r(j) = \sqrt{(x(i) - x(j))^2 + (y(i) - y(j))^2 + (z(i) - z(j))^2} \tag{17}$$

The coordinates of the centroids of the ellipses are given by $\{x, y, z\}$. Then, the neighboring ellipse with the nearest centroid is selected by taking the minimum of the distances $r(j)$. The spacings in each direction for the nearest ellipsoid are then extracted.

$$\{x_{\min}, y_{\min}, z_{\min}\} = \{|x(i) - x(j_{\min})|, |x(i) - y(j_{\min})|, |z(i) - z(j_{\min})|\} \tag{18}$$

This process is repeated for each ellipsoid, so that the distance in each direction from each ellipse to its nearest neighbor is recorded. This gives a distribution of spacings in the x, y, and z directions, which can be correlated to the RD, ND, and TD for a given material.

The second-phase particles are assumed to have no strength, and as such the particles are treated as pre-existing pores with the same size, shape, and spacing distribution. There are; however, several potentially important factors which are not captured by approximating the microstructure as a distribution of ellipsoids. While the ellipsoids capture the approximate size and shape of the second-phase particles, the real particles have sharp corners which could act as stress concentrators. In addition, there may be additional particles dispersed throughout the material which are smaller than 1 voxel and are not detected. Since these particles are smaller than what can be detected by the micro-CT, there is no way to capture them using this method. Since the microstructures are constructed from micro-CT scans, which detect second phase particles, pre-existing pores in the matrix material are also not captured by this method. The particles are all assumed to be the same precipitate material; however, it is possible that they could be different types of intermetallics, or be composed of different alloying or impurity elements. This could cause different particles to behave differently, perhaps debonding from the matrix material more easily and promoting void growth at that particle.

Two separate microstructures were analyzed using this approach. The first is rolled magnesium AZ31B, discussed previously in Chapter 2, and the second is aluminum 5059.

3.1.1 Rolled AZ31B microstructure

The AZ31B microstructure from Chapter 2 is shown in Figure 29. It has a distribution of small, evenly dispersed second phase particles, as well as long stringer particles that are aligned in the rolling direction.

Figure 29: Rolled AZ31B microstructure showing second phase particles

Figure 30 shows the distribution of second phase particle major and minor axis lengths for the rolled AZ31B sample. The majority of the particles have major axis lengths of less than 10 μm, and are approximately spherical. Very few minor axes are greater than 50 μm, while there are several particles with significantly longer major axes which is consistent with the presence of several large, elongated stringer particles.

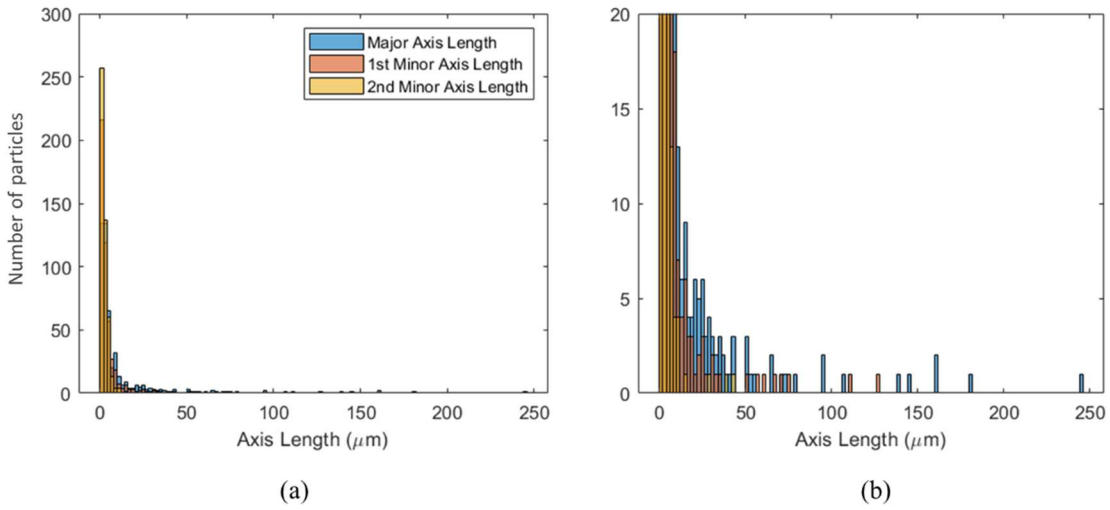

Figure 30: (a) AZ31B second phase particle size distribution. (b) detail of (a)

Figure 31 shows the particle spacing distribution of the AZ31B sample using the nearest neighbor spacing algorithm discussed previously. These spacings are generally small compared to the major axis length since this algorithm by design finds the closest particles to each other. In general, the spacings in the rolling direction are slightly higher than in the normal or transverse directions.

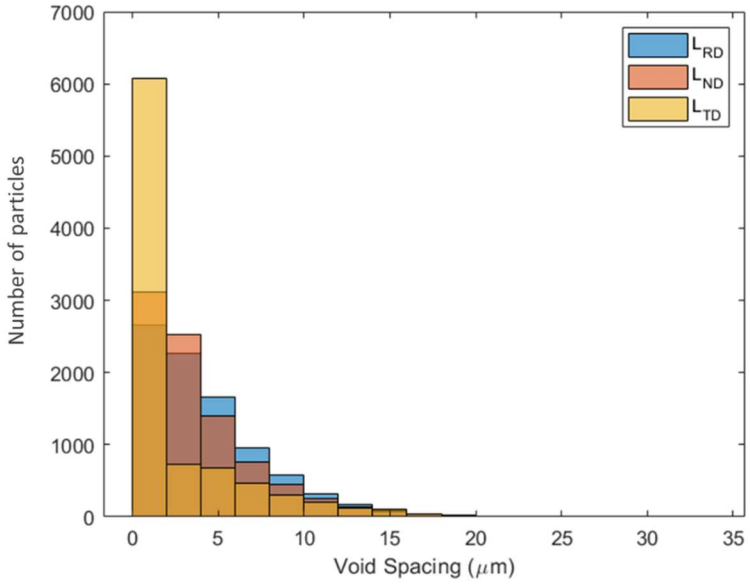

Figure 31: AZ31B second phase particle spacing distribution

Since the nearest-neighbor algorithm by definition locates the closest possible spacing between one void and its neighbor, the spacings in a particular direction can be smaller than the size of the particle in some cases. This may be the case when a large, elongated particle has a much smaller particle located near its centroid, so the nearest-neighbor spacing is smaller than would be expected for the size of the particle. However, when the radial spacing is compared to the smallest minor axis length, Figure 32, it is clear that the particles are not generally intersecting and the spacings are always larger than the smallest axis length.

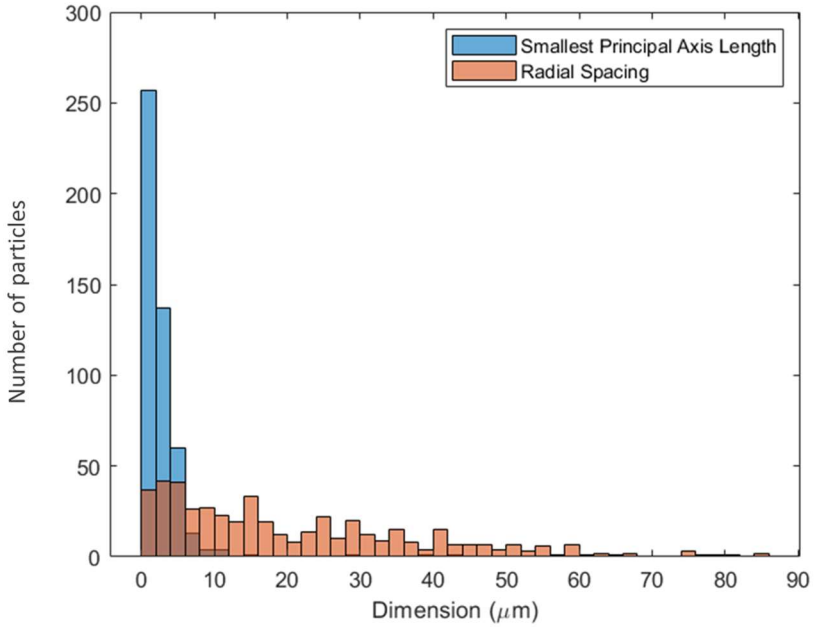

Figure 32: AZ31B smallest minor axis length compared to radial particle spacing

The average, standard deviation, minimum and maximum of the axis lengths, and spacings are tabulated in Table 5.

Table 5: AZ31B size and spacing statistics

		RD	ND	TD
Size	Average	10.12	5.19	2.66
	St. Deviation	22.98	11.15	3.30
	Minimum	1.15	1.15	1.15
	Maximum	244.59	127.54	42.72
Spacing	Average	11.45	9.26	10.62
	St. Deviation	11.46	12.24	11.78
	Minimum	0.00	0.00	0.00
	Maximum	68.64	75.66	59.94

Figure 33 shows the orientations of each of the particles' principal axes. The pole figure plots axes aligned along the RD, ND, and TD as points closer to the corresponding poles. The TD corresponds to the center. The color scale denotes the major axis length. Particles with an axis length of only 1 voxel are not plotted since the orientation of a cube is not meaningful. The smallest particles are not strongly oriented in any particular direction. However, the largest particles (blue) have major axes which are aligned along the RD-TD plane, corresponding to the long stringers seen in Figure 29.

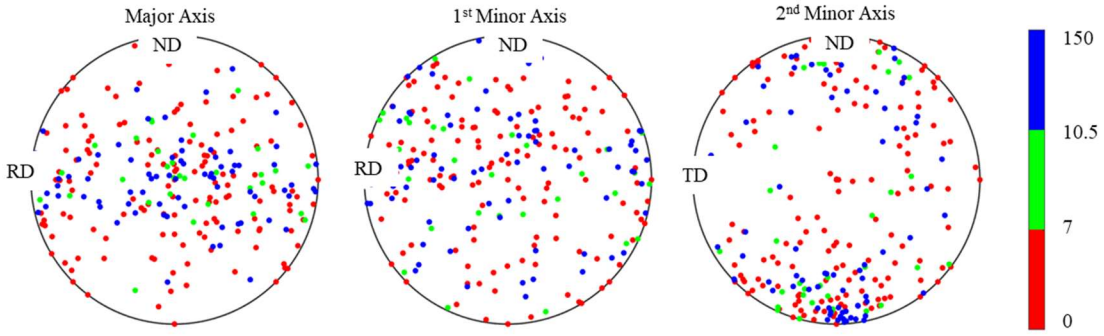

Figure 33: Pole figure showing orientation distribution of AZ31B second phase particles. Color scale denotes major axis length.

3.1.2 Al 5059 microstructure

A sample of rolled Al 5059 was also analyzed. As shown in Figure 34, the microstructure consists of evenly dispersed, approximately ellipsoidal second phase particles. It also does not contain the long stringer particles seen in AZ31B. Figure 35 shows the distribution of second phase particles sizes for the Al 5059 microstructure.

The majority of the particles have major axes of less than 50 µm, and minor axes of less than 20 µm.

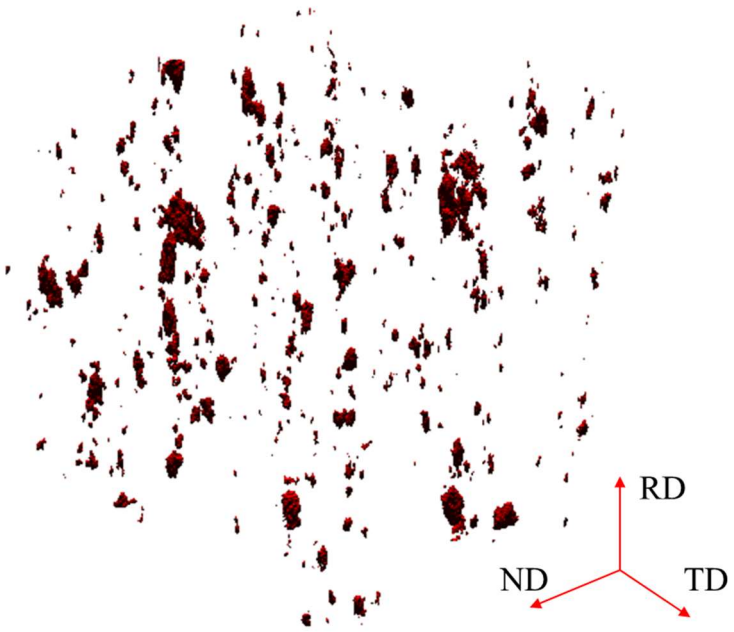

Figure 34: Rolled Al 5059 microstructure showing second phase particles

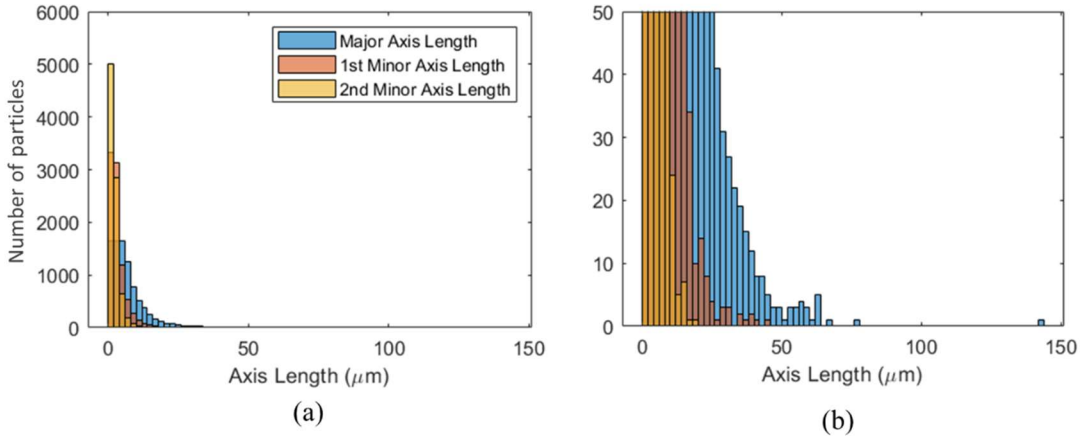

Figure 35: (a) Al 5059 second phase particle size distribution. (b) detail of (a)

Figure 36 shows the distribution of particle spacings. The particles are also fairly evenly distributed. The spacings in the rolling and normal directions are similar, while the spacing is generally smaller in the transverse direction.

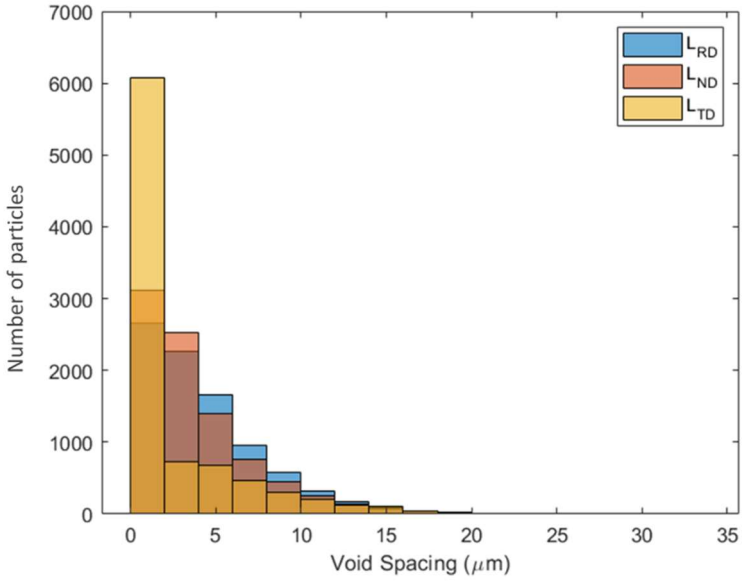

Figure 36: Al 5059 second phase particle spacing distribution

Figure 37 compares the radial spacing to the smallest axis length. As in the case of the AZ31B microstructure, the major axis lengths are generally much longer than the nearest-neighbor spacings due to the arrangement of the particles and the algorithm's bias towards the smallest possible spacing between centroids.

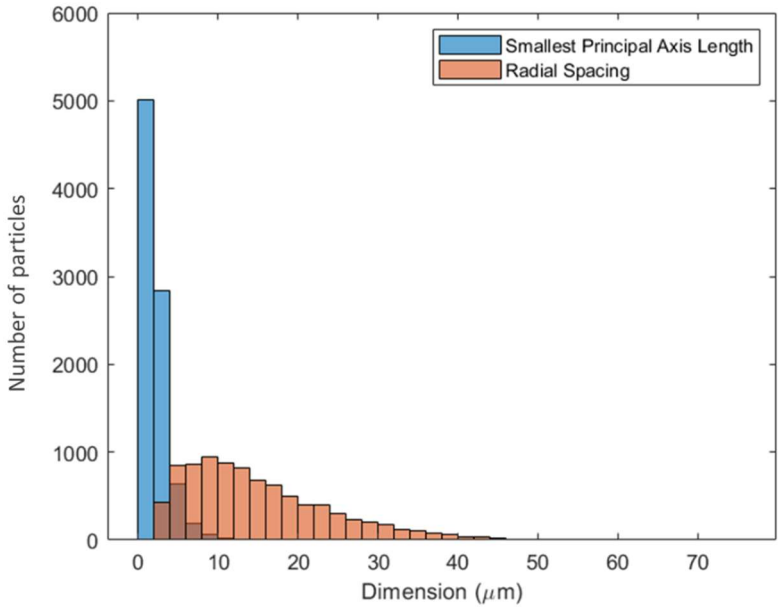

Figure 37: Al 5059 smallest minor axis length compared to radial particle spacing

The average, standard deviation, minimum and maximum of the axis lengths, and spacings are tabulated in Table 6.

Table 6: Al 5059 size and spacing statistics

		RD	ND	TD
Size	Average	3.64	1.71	1.11
	St. Deviation	3.45	1.57	0.81
	Minimum	0.43	0.43	0.43
	Maximum	71.94	22.57	9.85
Spacing	Average	4.33	3.84	2.49
	St. Deviation	3.57	3.40	3.72
	Minimum	0.00	0.00	0.00
	Maximum	33.50	29.14	27.32

Figure 38 shows the orientation distribution of the particles. The major axes are all aligned generally aligned along the rolling direction, with the minor axes distributed about the ND-TD plane. The larger particles, shown in yellow, are more strongly aligned in the rolling direction.

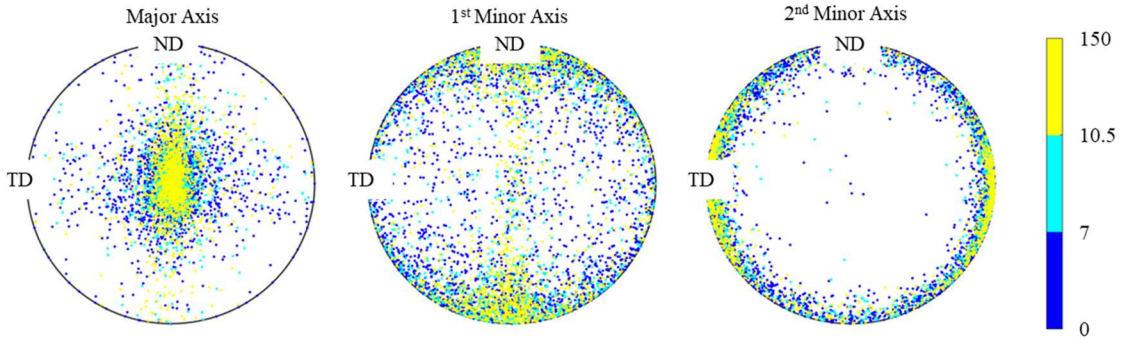

Figure 38: Pole figure showing orientation distribution of Al 5059 second phase particles. Color scale denotes major axis length.

3.2 Conversion of ellipsoidal distribution to representative unit cell

As in Chapter 2, we are interested in the effects of orientation on the stress strain response of the material. To capture the effects of changing the loading orientation, the average spacings and axis lengths were extracted with the Z axis (loading axis) aligned with the rolling direction, and the unit cell was rotated to capture the other orientations. The rotation of the RD-ND into radial coordinates is shown in Figure 39.

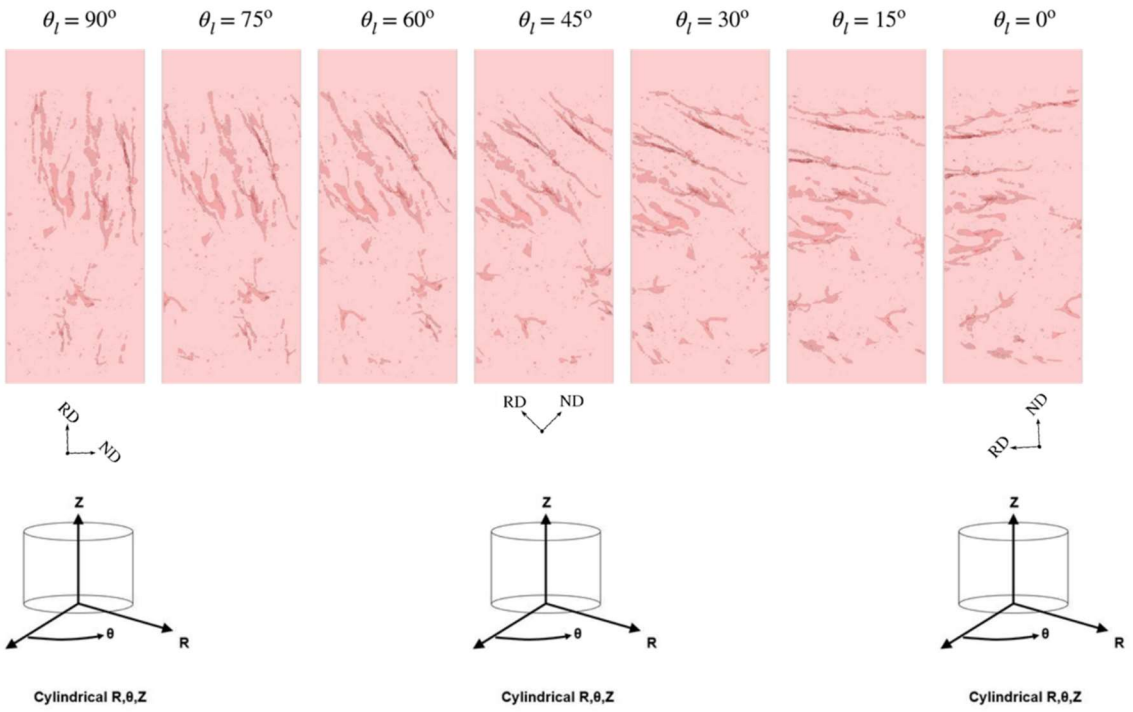

Figure 39: Representative unit cell orientation compared to particle orientation

The average ellipsoidal axis lengths, (R_{RD}^{avg}, R_{ND}^{avg}, R_{TD}^{avg}) and spacings (L_{RD}^{avg}, L_{ND}^{avg}, L_{TD}^{avg}) were extracted at an orientation of θ_l=90°. For each orientation, the spacings are converted from the material coordinate system (RD-ND-TD) to the loading coordinate system (X-Y-Z).

$$L_Z^{avg} = L_{ND}^{avg} cos\theta_l + L_{RD}^{avg} sin\theta_l \tag{19}$$

$$L_X^{avg} = L_{ND}^{avg} sin\theta_l + L_{RD}^{avg} cos\theta_l \tag{20}$$

$$L_Y^{avg} = L_{TD}^{avg} \tag{21}$$

The radii are transformed similarly.

$$R_Z^{avg} = R_{ND}^{avg} cos\theta_l + R_{RD}^{avg} sin\theta_l \tag{22}$$

$$R_X^{avg} = R_{ND}^{avg} sin\theta_l + R_{RD}^{avg} cos\theta_l \tag{23}$$

$$R_Y^{avg} = R_{TD}^{avg} \tag{24}$$

To form an axisymmetric unit cell, the X and Y dimensions are averaged for both the average spacing and average radii.

$$L_R^{avg} = \frac{1}{2} L_X^{avg} + \frac{1}{2} L_Y^{avg} \tag{25}$$

$$R_R^{avg} = \frac{1}{2} R_X^{avg} + \frac{1}{2} R_Y^{avg} \tag{26}$$

CHAPTER 4

REPRESENTATIVE UNIT CELL STRESS-STRAIN RESPONSE

4.1 AZ31B microstructure

For the rolled AZ31B microstructure, the average spacings and radii were extracted in the rolling, normal, and transverse directions. At θ_l=90°, the stringer particles (along the RD) are aligned along the loading (Z) axis, and the average unit cell that is slightly elongated in the Z direction ($L_z / L_r = 1.2$) with a much more elongated void ($R_z / R_r = 2.6$). This reproduces the effect seen in the true microstructure where many small particles are stacked very closely in the rolling direction, shown in Figure 40, such that the spacing between the edges of particles is small.

Figure 40: Small particles forming long stringers in rolled AZ31B

At θ_l=0°, the representative unit cell is wider ($L_z / L_r = 0.8$), with a more disk like void ($R_z / R_r = 0.8$). This is due to modeling the unit cell as axisymmetric about the Z-axis. In the real microstructure at θ_l=0° the stringer particles are aligned across the loading axis. Table 7 details the representative unit cell geometry for each orientation.

Table 7: AZ31B representative unit cell geometry

θ_l	**0**	**15**	**30**	**45**	**60**	**75**	**90**
R_Z	5.19	7.63	9.55	10.82	11.36	11.11	10.12
R_R	6.39	6.89	7.01	6.74	6.11	5.15	3.93
L_Z	9.26	11.91	13.74	14.64	14.55	13.46	11.45
L_R	11.03	12.04	12.58	12.63	12.18	11.26	9.94
R_Z / R_R	0.81	1.11	1.36	1.61	1.86	2.16	2.58
L_Z / L_R	0.84	0.99	1.09	1.16	1.19	1.19	1.15
f_0 (%)	12.53	13.99	14.38	14.04	13.08	11.50	9.19

The porosities of the unit cells in Table 7 are significantly higher than the porosity of the AZ31B DNS, listed in Table 8. However, the higher porosities in each case occurs for the 30° through 60° orientations.

Table 8: Porosity as a function of orientation from Lloyd results [9]

θ_l	**0**	**15**	**30**	**45**	**60**	**75**	**90**
f_0 (%)	0.630	0.633	0.634	0.632	0.657	0.603	0.586

As shown in Figure 41, the unit cell is elongate at 90°, and gradually becomes more square with a more oblate void as the unit cell rotates towards 0°.

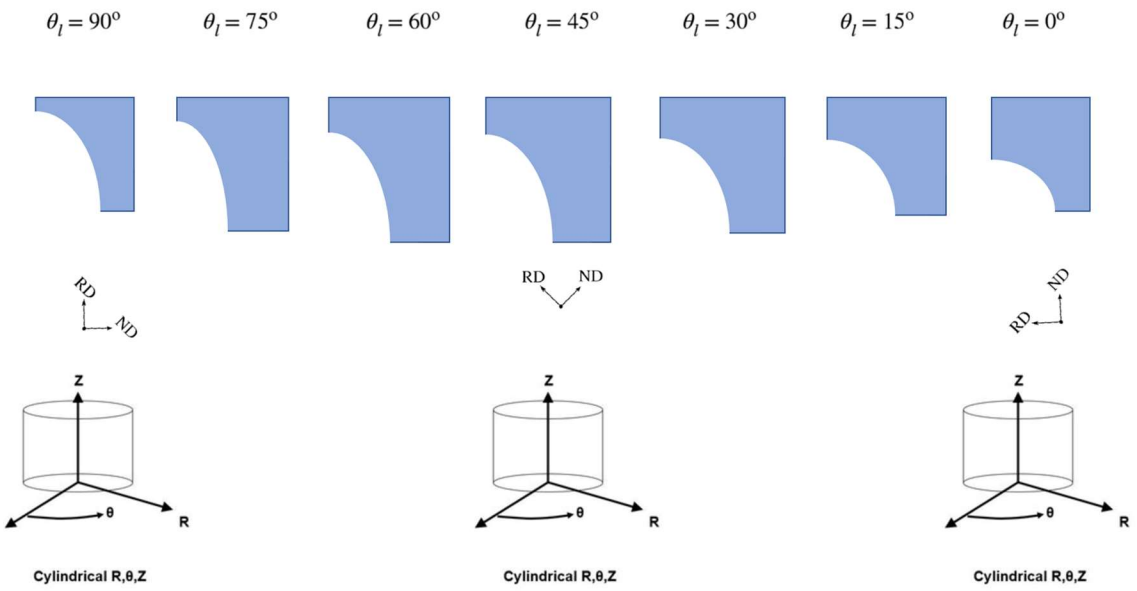

Figure 41: Representative unit cell geometry generated from AZ31B microstructure

These unit cells do not look like a pure rotation because the rotation is done in the X – Z plane, and the spacing in R is an average of the Y (TD) and X spacings. To illustrate what the unit cells would look like if only the rolling direction and normal

direction dimensions accounted for, the unit cells without the transverse direction included are illustrated in Figure 42. In this case the unit cells for 0° and 90° have the same dimensions in opposite directions. Note that without the TD effect, the void becomes spherical at 45°.

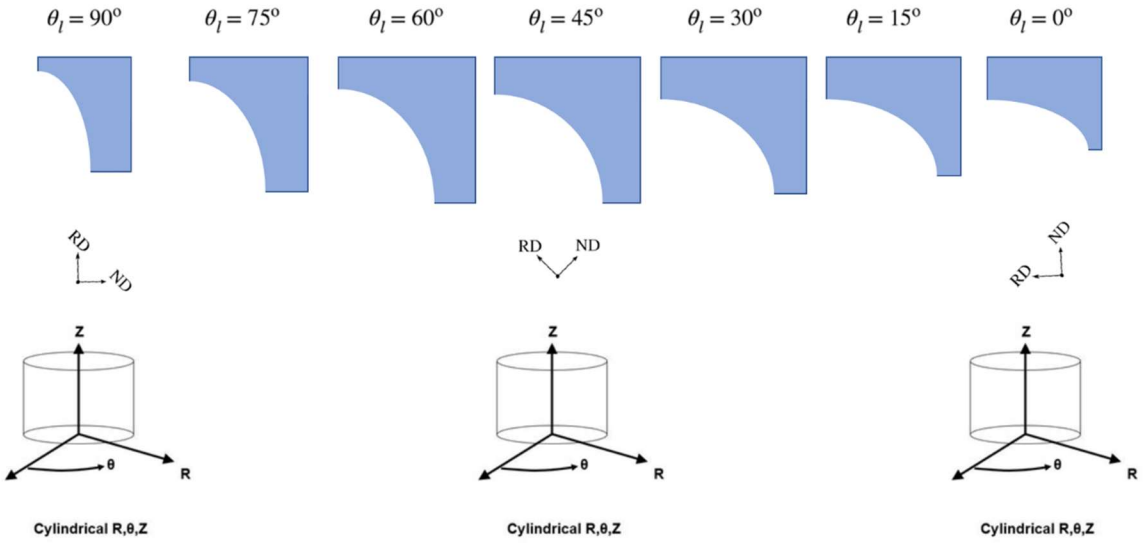

Figure 42: Representative unit cell geometry without TD averaging

These representative unit cells were used to run axisymmetric uniaxial stress finite element simulations with parallel lateral boundaries to compare with the results of Figure 5. These simulations are done with a prescribed strain in the Z direction. The boundary conditions are illustrated in Figure 43.

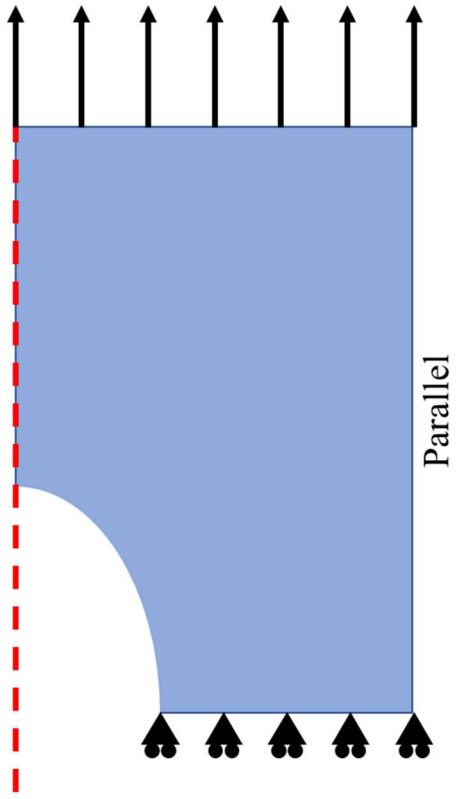

Figure 43: Axisymmetric uniaxial stress boundary conditions

The results of these simulations are shown in Figure 44. While the strains to failure do not match the DNS results exactly, the trends are similar in that the 0°-30° unit cells had the lowest strains to failure while the higher angle orientations had higher strains to failure. The discrepancy in strain-to-failure is likely because these unit cell calculations do not include necking. In the DNS calculations of Figure 5, necking is seen and the maximum true strain in the necked region will be much higher than the nominal strain.

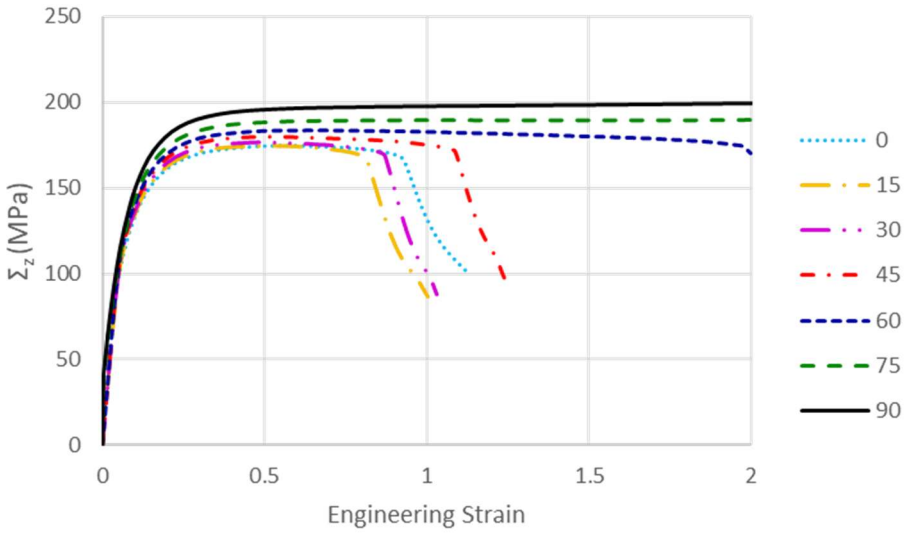

Figure 44: AZ31B representative unit cell engineering stress - strain under uniaxial stress loading

The representative unit cells defined in Table 7 were also simulated under uniaxial strain loading to approximate the conditions seen during a ballistic impact. These boundary conditions are illustrated in Figure 45.

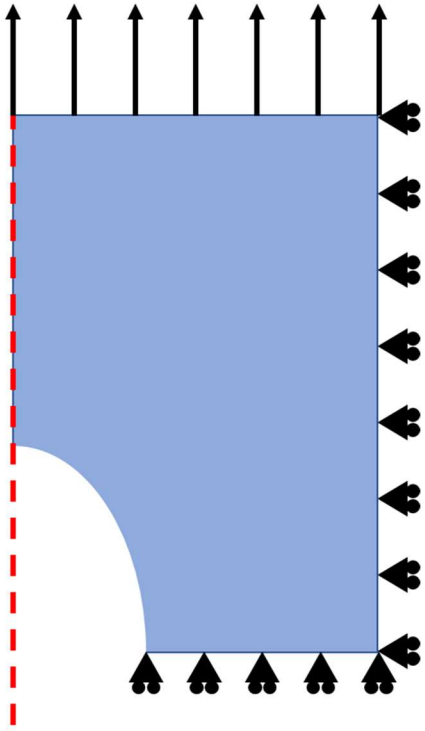

Figure 45: Axisymmetric uniaxial strain boundary conditions

As seen in Figure 46, the 90° orientation, which had the highest strain to failure in the DNS (Figure 5), saw the highest peak stress. Similarly, the orientations with lower failure strains reached a lower peak stress.

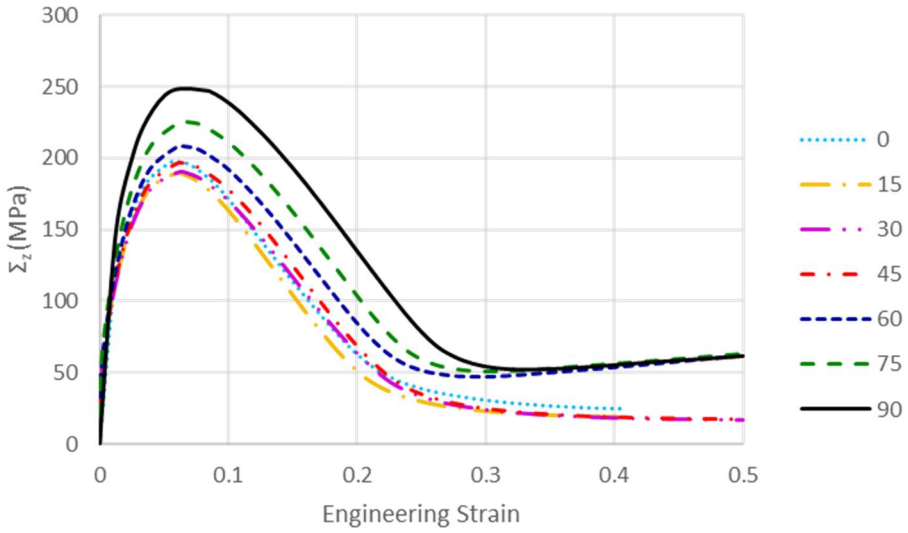

Figure 46: AZ31B representative unit cell engineering stress - strain under uniaxial strain loading

Peak Σ_z in the uniaxial strain unit cell calculations (Figure 46) is essentially spall strength at low strain rate and very low shock pressure. The spall strengths seen are low compared to the experimental results in Figure 47, which is likely due to the high initial porosities of these unit cells ($f_0 \approx 12\%$). The highest spall strength for these unit cells was 247 MPa for the 90° orientation, while the lowest was for the 15° and 30° unit cells at 190 MPa.

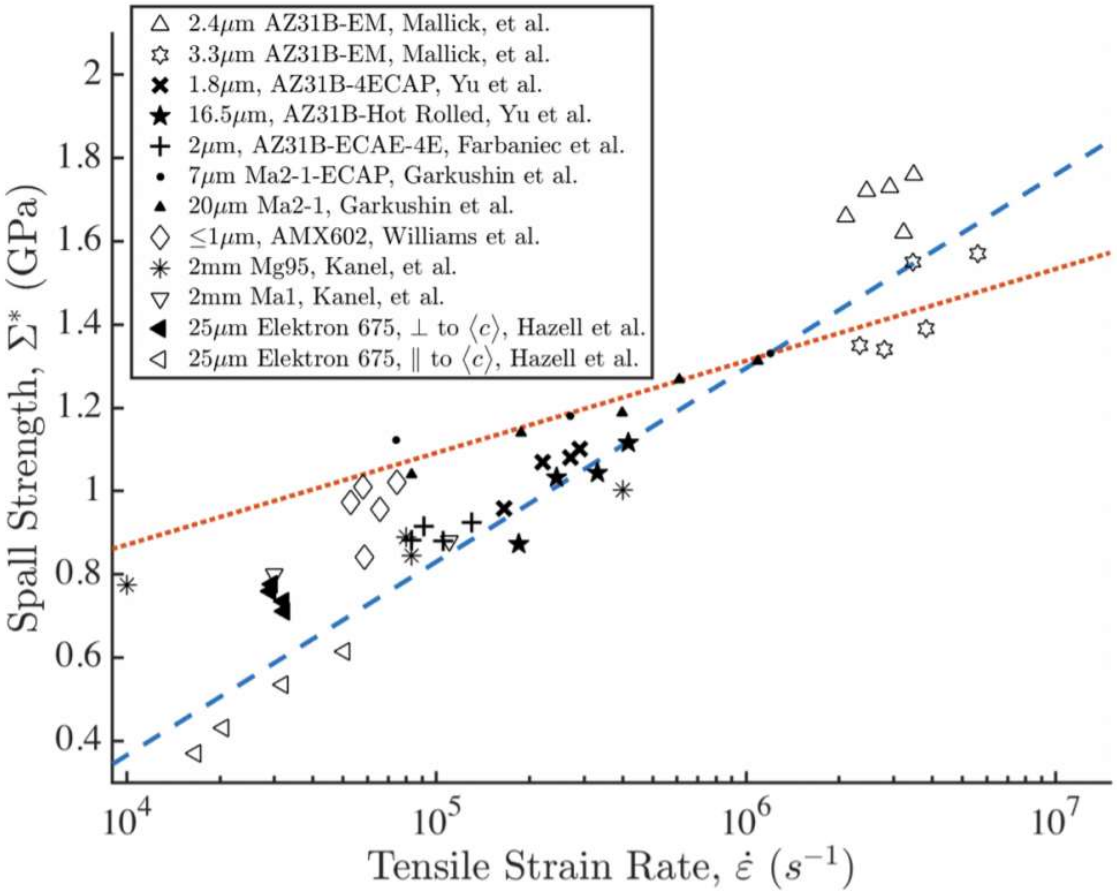

Figure 47: Spall strength of polycrystalline pure and alloyed magnesium. The blue line denotes the trend in AZ31b, while the red line denotes the trend in Ma2-1. Reprinted from [10].

To compare the spall strength estimates for a more realistic initial porosity, the unit cell size and void aspect ratio were kept the same, while the void size was adjusted to use the real porosities given in Table 8. These adjusted unit cell geometries are given in Table 9.

Table 9: AZ31B unit cell geometries with real porosity

θ_l	**0**	**15**	**30**	**45**	**60**	**75**	**90**
R_Z	1.92	2.72	3.38	3.85	4.19	4.16	4.04
R_R	2.36	2.45	2.48	2.40	2.25	1.93	1.57
L_Z	9.26	11.91	13.74	14.64	14.55	13.46	11.45
L_R	11.03	12.04	12.58	12.63	12.18	11.26	9.94
R_Z / R_R	0.81	1.11	1.36	1.61	1.86	2.16	2.58
L_Z / L_R	0.84	0.99	1.09	1.16	1.19	1.19	1.15
f_0 (%)	0.630	0.633	0.634	0.632	0.657	0.603	0.586

The stress-strain responses under uniaxial strain loading for these adjusted unit cells are given in Figure 48. The spall strengths for these unit cells range from 389 MPa (60°) to 401 MPa (0°), which more closely agrees with the experimental results of Figure 47.

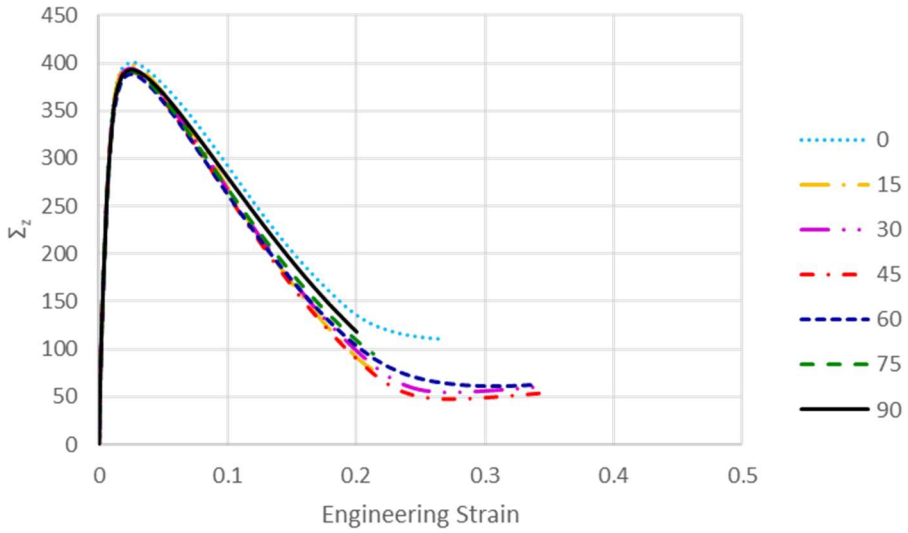

Figure 48: AZ31B representative unit cell with true porosity engineering stress - strain under uniaxial strain loading

4.2 Al microstructure

To determine the response of AZ31B if it had a more dispersed microstructure, the representative unit cells for the Al 5059 microstructure in Chapter 3 were extracted. This microstructure has a more dispersed distribution of particles that are generally smaller, and does not have the long stringer particles seen in rolled AZ31B. For the aluminum microstructure, the average radii and spacings used to construct the unit cells were the volume-weighted averages, to better capture the influence of the larger ellipsoidal particles.

The geometries of the representative unit cells used are given in Table 10. As illustrated in Figure 49, the unit cells generated from the Al microstructure range from very short and wide at 0°, to long and narrow at 90°.

Table 10: Al microstructure representative unit cell geometry

θ_l	**0**	**15**	**30**	**45**	**60**	**75**	**90**
R_Z	2.28	4.77	6.92	8.61	9.70	10.14	9.89
R_R	6.64	6.77	6.55	6.00	5.16	4.08	2.84
L_Z	5.42	8.35	10.72	12.35	13.14	13.03	12.04
L_R	11.01	11.50	11.56	11.16	10.35	9.17	7.70
R_Z / R_R	0.34	0.70	1.06	1.43	1.88	2.48	3.48
L_Z / L_R	0.49	0.73	0.93	1.11	1.27	1.42	1.56
f_0 (%)	10.23	13.17	13.84	13.44	12.25	10.29	7.46

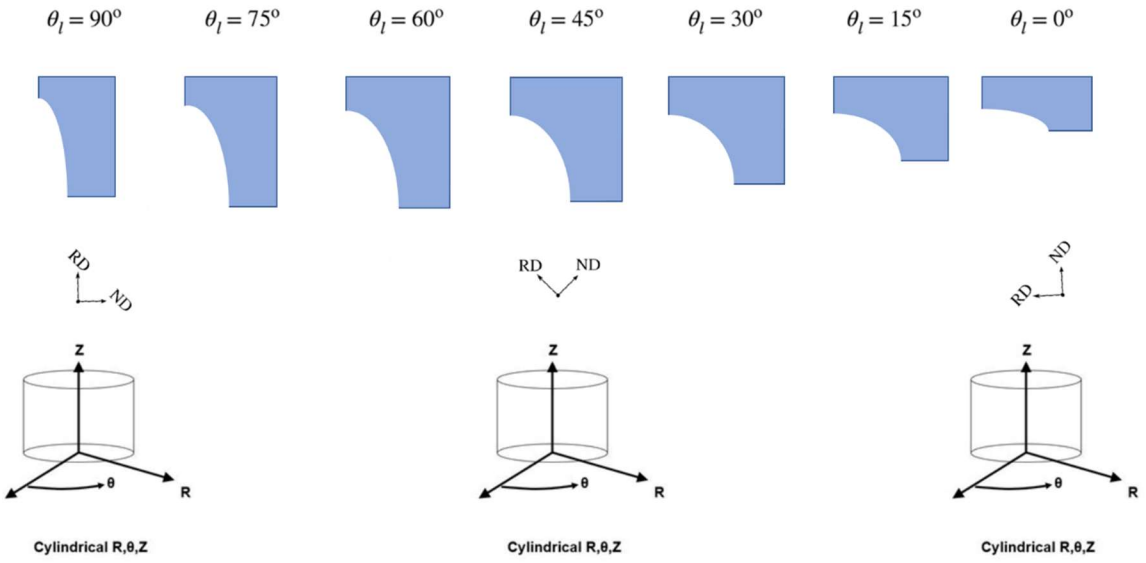

Figure 49: Representative unit cell geometry generated from Al microstructure

The representative unit cell geometries for Al 5059 were simulated using the material properties of AZ31B. Figure 50 shows the response of the Al microstructure unit cells under uniaxial stress loading, with boundary conditions as described in Figure 43. For these unit cells, the 30° had the lowest strain to failure, while the 75° and 90° unit cells never saw the same sharp drop in stress seen in the other unit cells. This may be because the 30° unit cell consisted of a nearly spherical void in a nearly spherical unit cell, while the unit cells for 75° and 90° were very elongated in the Z direction.

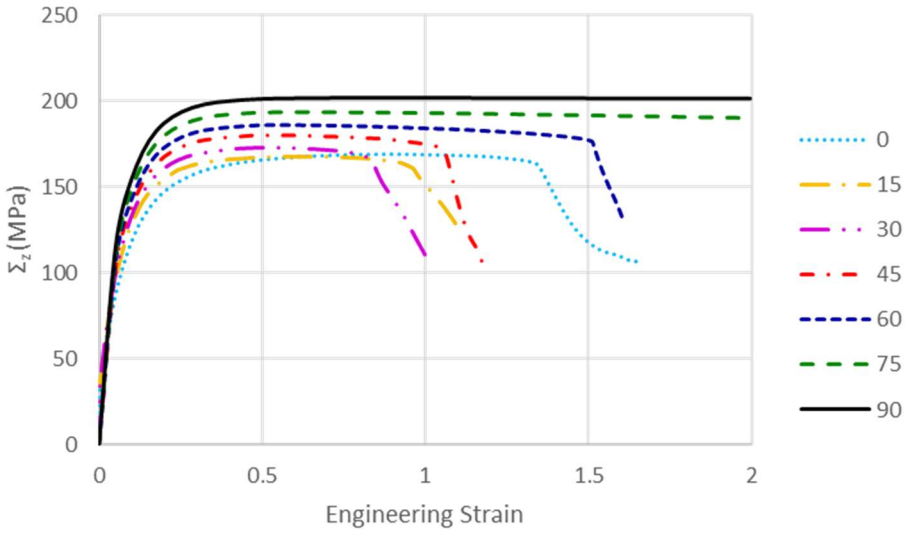

Figure 50: Al microstructure representative unit cell engineering stress - strain under uniaxial stress loading

Figure 51 shows the engineering stress – engineering strain response of the unit cells under the uniaxial strain boundary conditions shown in Figure 45. As with the unit cells generated from the AZ31b microstructure, the ordering of strains to failure of the

uniaxial strain response with orientation are similar to those of the response under uniaxial strain. For these unit cells, the peak stress was lowest for the 30° unit cell, and highest for 90°.

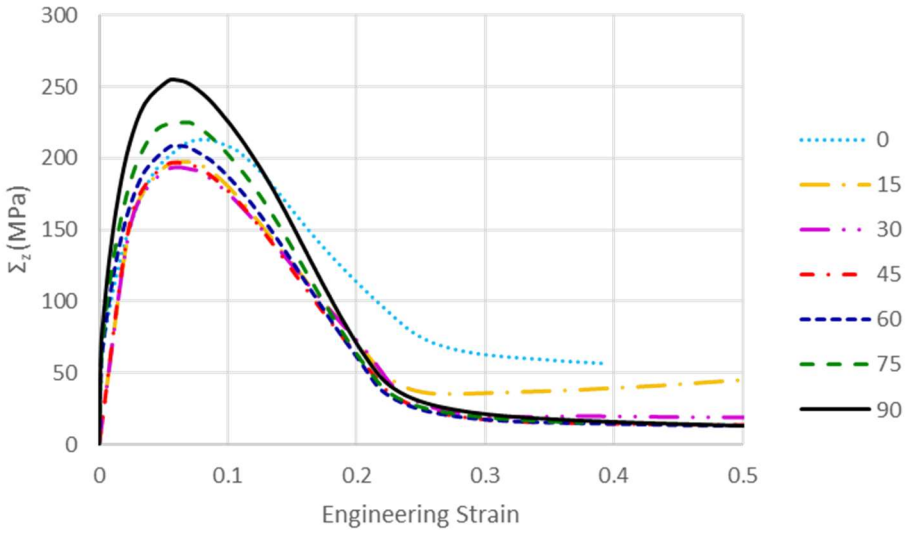

Figure 51: Al representative unit cell engineering stress - strain under uniaxial strain loading

CHAPTER 5

IMPROVEMENTS TO SPACING STATISTICS

In the previous chapters, the void spacing distribution was determined by finding the smallest radial distance between particle spacings [38]. While this method is a simple algorithm, it has limitations. For example, if microstructure consists of horizontal lines of voids, the nearest-neighbor algorithm breaks down.

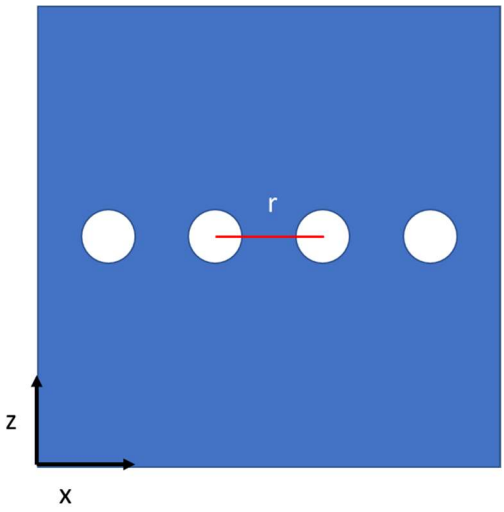

Figure 52: Example of microstructure geometry for which the simple nearest-neighbor algorithm breaks down

Since the algorithm finds the void with the smallest radial distance, then calculates the distance between voids in each direction, the nearest voids will always be directly aligned in the X-direction. This will result in an average spacing of zero in the z-direction, instead of a more intuitive unit cell that has a short L_R^{avg} and long L_Z^{avg}. Even if

the voids are slightly staggered in the Z-direction, this can result in an average spacing that is smaller than the void radius. Because of this problem, an improved method of estimating the representative void spacing is needed.

5.1 Voronoi tessellation approach

The Voronoi tessellation is an algorithm that takes a set of points and calculates a region around each point that is closer to that point than any other [38]. This will produce a set of polygons that encompass the entire region and can be used to approximate a representative unit cell that will reproduce the stress strain curve of the void distribution.

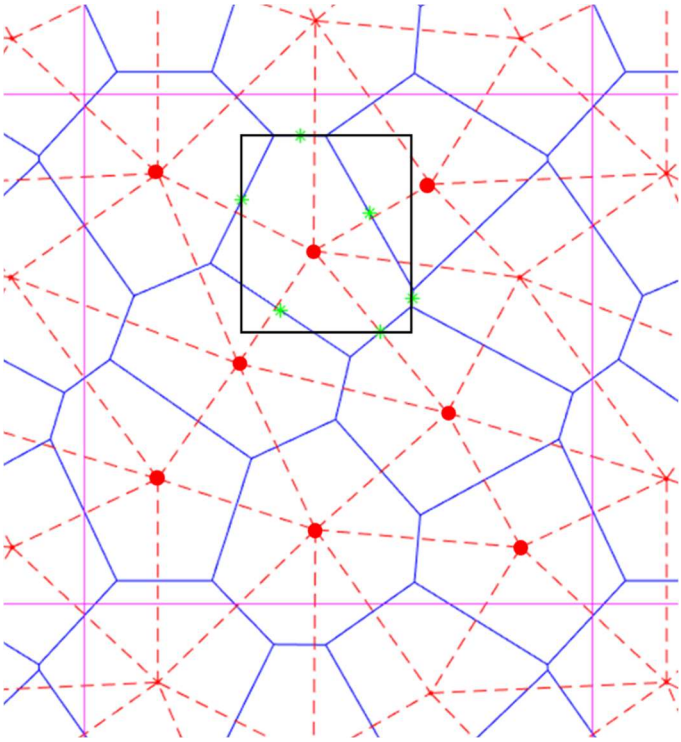

Figure 53: Voronoi diagram showing rectangular spacing extracted from midpoints of the Voronoi polygons. Red dots denote void centers.

To use the Voronoi diagram to extract a representative unit cell, the Voronoi regions need to be converted to rectangular regions. To do this, the regions were approximated as the smallest rectangle that can encompass all of the midpoints of the polygon edges, as shown in Figure 53. This yields a representative spacing for each void in the region. This distribution of spacings yields several potential unit cells, including the average spacing, minimum spacing, maximum spacing, or a combination of the average spacing and the standard deviation.

To evaluate the accuracy of using this method to generate a representative unit cell, and determine which unit cell is best, three different distributions of circular voids were generated. The three different void distributions result in unit cell spacing aspect ratios, $\lambda = L_z/L_x$, of approximately 3, 1, and 0.3.

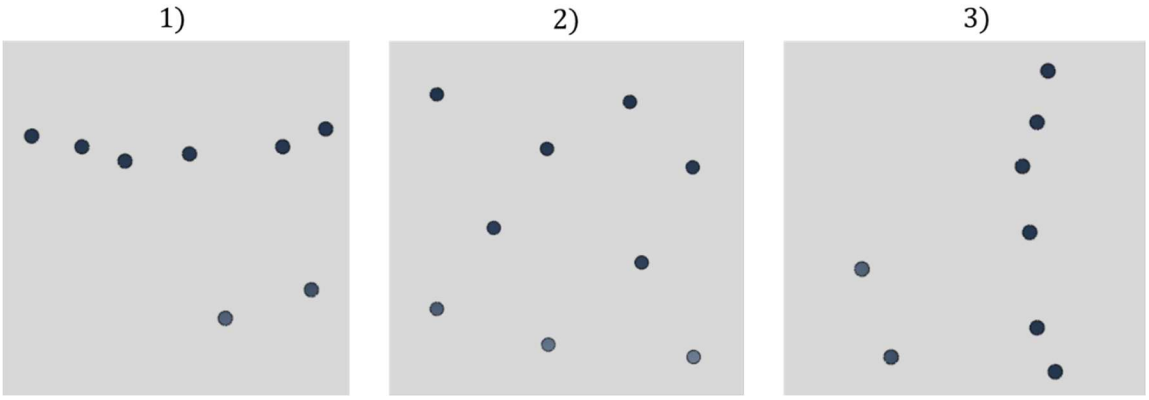

Figure 54: Three void distributions with different spacing aspect ratios $\boldsymbol{\lambda \cong L_z/L_x}$ 1) $\boldsymbol{\lambda \cong 3}$ 2) $\boldsymbol{\lambda \cong 1}$ and 3) $\boldsymbol{\lambda \cong 0.3}$

The Voronoi diagrams for each void distribution are shown in Figure 55. For void distributions 1, and 3, the Voronoi regions are generally long and skinny, capturing the close spacing of the voids in one direction and larger spacing in the other.

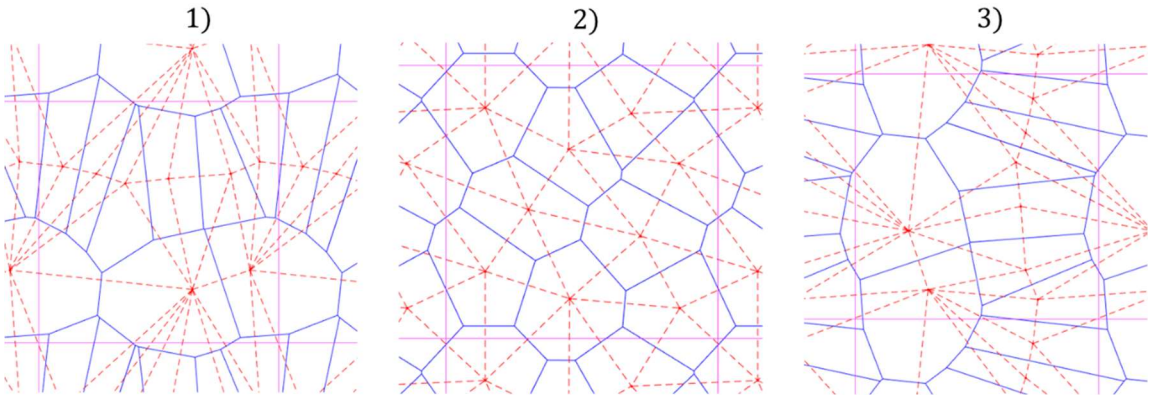

Figure 55: Voronoi diagrams for three void distributions

Six potential unit cells were selected using the distribution of unit cell sizes extracted from each void distribution's Voronoi diagram. These potential unit cells are listed in

Table 11. These potential unit cell formulations use the sizes of the Voronoi unit cells of a particular distribution of circular voids to construct a single, representative, unit cell.

Table 11: Potential unit cell geometries

Unit Cell	L_X	L_Z
1	L_X^{avg}	L_Z^{avg}
2	$L_X^{\min}$	$L_Z^{\min}$
3	$L_X^{\max}$	$L_Z^{\max}$
4	$L_X^{avg} - stdev(L_X)$	$L_Z^{avg} - stdev(L_Z)$
5	$L_X^{avg} - stdev(L_X)$	λL_X $\lambda = L_Z^{avg} / L_X^{avg}$
6	$L_X^{avg} - \frac{1}{2}(stdev(L_X) + stdev(L_Z))$	λL_X $\lambda = L_Z^{avg} / L_X^{avg}$

5.2 Accuracy of Voronoi unit cells

For the three void distributions in Figure 54, the potential unit cell geometries extracted from their Voronoi diagrams are listed in Table 12. The unit cells in Table 12 were calculated for each void distribution using the formulations in Table 11. The initial porosities for all of the random distributions were 1%. The Voronoi unit cells produce porosities that are similar to the initial porosity of the random distribution. This is a significant improvement over the nearest-neighbor approach which produced unit cells in Chapter 4 with significantly higher porosities than the original microstructures.

Table 12: Unit Cell dimensions

	Void distribution 1				Void distribution 2				Void distribution 3			
	L_X	L_Z	λ	f_0 (%)	L_X	L_Z	λ	f_0 (%)	L_X	L_Z	λ	f_0 (%)
1	14.26	27.64	1.94	0.8	18.81	17.02	0.90	0.9	27.64	14.26	0.52	0.8
2	8.60	24.23	2.82	1.5	16.73	11.88	0.71	1.4	24.23	8.60	0.36	1.5
3	27.48	32.66	1.19	0.3	21.92	20.42	0.93	0.6	32.66	27.48	0.84	0.3
4	6.92	24.11	3.49	1.9	16.70	13.67	0.82	1.2	24.11	6.92	0.29	1.9
5	6.92	13.40	1.94	3.4	16.70	15.11	0.90	1.1	24.11	12.44	0.52	1.0
6	8.82	17.10	1.94	2.1	16.08	14.55	0.90	1.2	22.20	11.45	0.52	1.2

Each of the three void distributions shown in Figure 54 were modeled in 2D in ABAQUS under uniaxial strain loading, using plane strain elements. The void radius was selected such that the total porosity was 1%. Figure 56 shows the stress strain responses of each of the three void distributions. Void distribution 2 has voids which are the most evenly distributed, and as expected sees the highest peak stress.

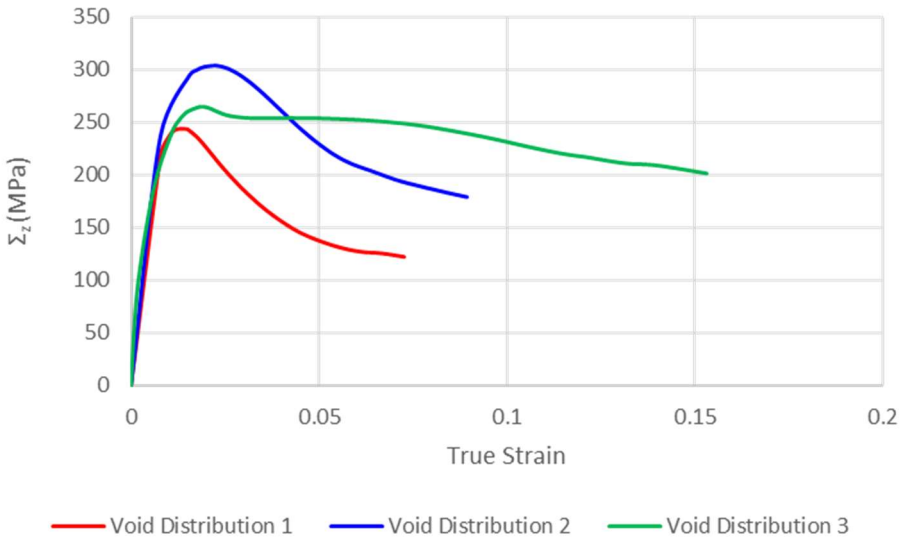

Figure 56: Comparison of stress-strain response of DNS of three void distributions under uniaxial strain

Each of the unit cells in Table 12 were also simulated under uniaxial strain with the same initial void radius. Figure 57, Figure 58 and Figure 59 show the stress-strain curves for the DNS of each void distribution in Figure 54 compared to its representative unit cell results.

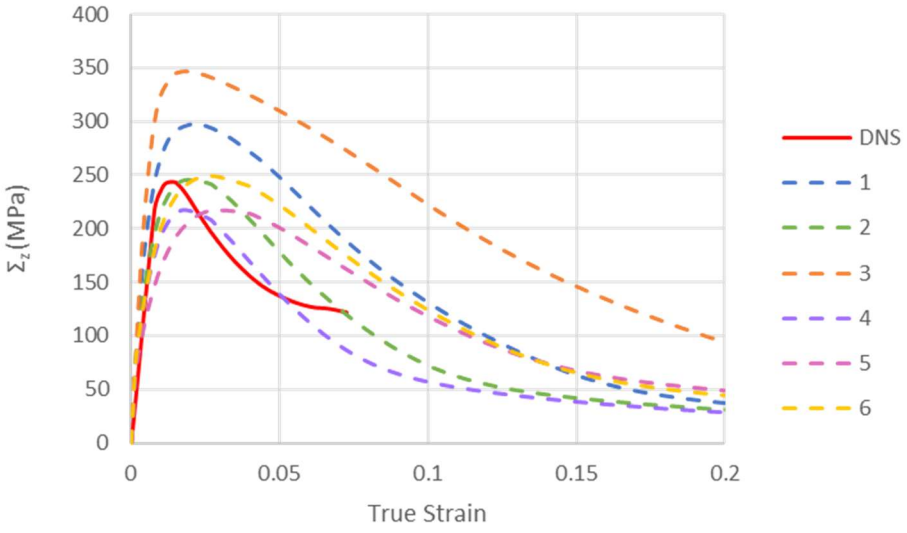

Figure 57: Z-direction stress strain curve for void distribution 1

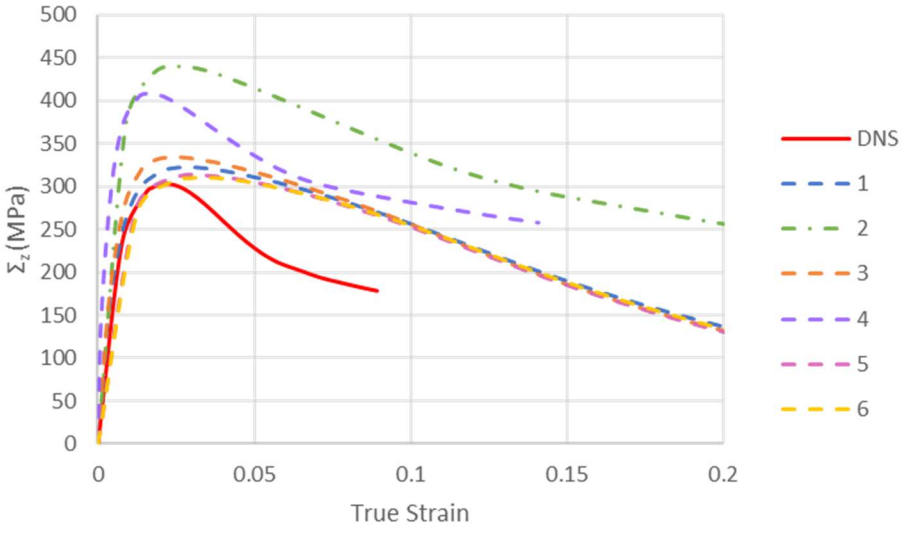

Figure 58: Z-direction stress strain curve for void distribution 2

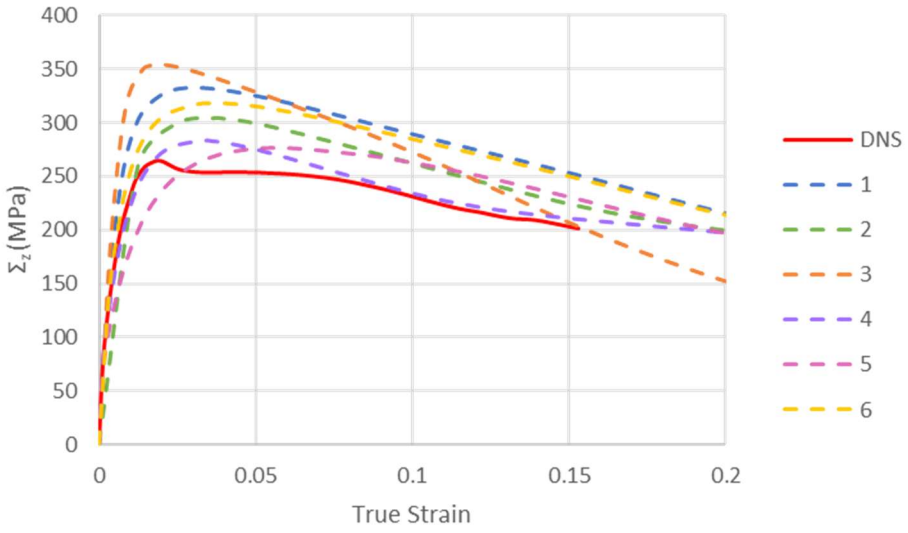

Figure 59: Z-direction stress strain curve for void distribution 3

While the unit cell results are not exact matches for the results of the DNS of the void distributions, they do provide a close approximation of the peak stresses. Unit cells 5 and 6 most closely approximate the results of the realistic void distribution for all 3 void geometries. As shown in Figure 60, the error between the peak stress (Σ_z^{max}) of the unit cells and the void distribution is lowest for these unit cells. Unit cell 5 has an average error for all three void geometries of 6.2%, and unit cell 6 has an average error of 8.3%.

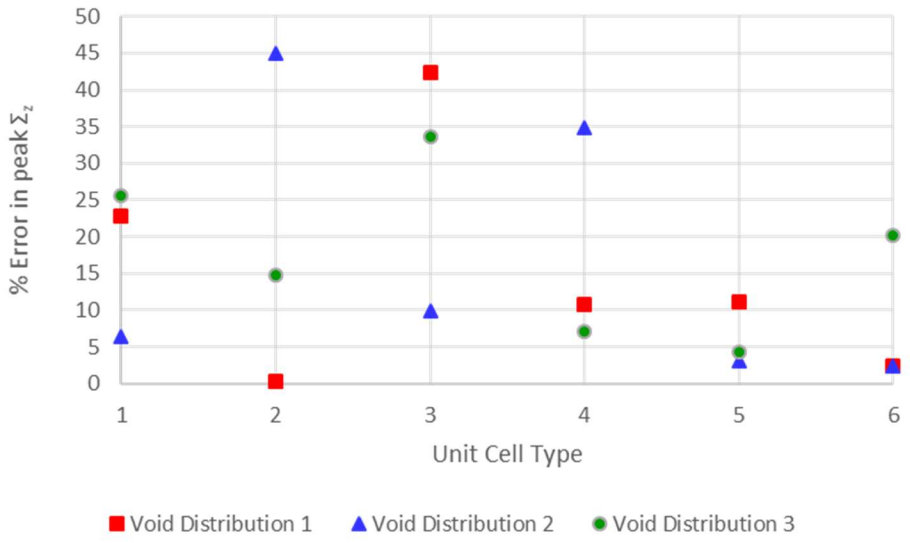

Figure 60: Difference in peak stress between unit cell and realistic void distribution

The error in the stress at higher strains is generally greater than the error in peak stress as illustrated in Figure 61. At 5% strain, unit cell 4 has the lowest average error for all three void geometries of 19.7%; however, unit cell 5 provides the second closest approximation at 29.5% average error.

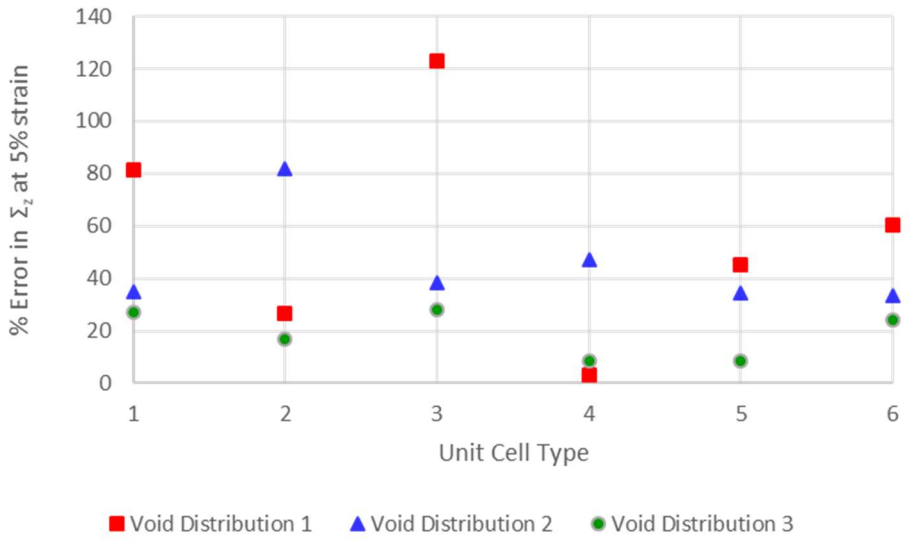

Figure 61: Difference in stress at 5% strain between unit cell and realistic void distribution

This method of using the Voronoi tessellation to generate the representative unit cell dimensions for a distribution of void locations produces unit cells which reasonably match the peak stress of the original void distribution. This method also avoids the problems encountered by the simple nearest-neighbor approach when voids are arranged in closely spaced lines.

CHAPTER 6

SUMMARY AND CONCLUSIONS

This work provides a method of accounting for the effects of precipitate orientation on the stress-strain response of a material by modeling second phase particles as pre-existing voids. Chapter 2 attempts to capture the effects of particle geometry by calibrating a set of GTN parameters to stress strain curves generated by explicitly modelling the particles for each orientation. While the stress-strain response of the calibrated models is very accurate, this method requires complex DNS to generate the stress-strain curves for calibration, which is time and computationally intensive. Chapter 3 presents a method of converting a real microstructure to a distribution of ellipsoids, and using that distribution to generate and average representative unit cell geometry. That unit geometry is then rotated to obtain a set of unit cells for each loading orientation. In Chapter 4 the stress strain response of the unit cells generated using the method in Chapter 3 for both the microstructure of rolled AZ31B and for an aluminum alloy are compared to the stress-strain response of the DNS of AZ31B. While the exact strains to failure are not the same, the trends in strain to failure of the AZ31B unit cells and the DNS are similar. In Chapter 5 an improved method of estimating the representative spacing using Voronoi unit cells is presented. This method avoids problems inherent in the simpler nearest-neighbor approach used in the previous section. The Voronoi unit cell approach also accurately approximates the peak stress under plane strain loading compared to DNS of a distribution of spherical voids used to generate the

unit cells. To accurately describe a three-dimensional distribution of particles, the Voronoi spacing algorithm would need to be extended to 3D in the future. Using the presented methods of generating representative stress-strain curves directly from a given microstructure allows the effects of second phase particle geometry to be estimated without having to explicitly model the precipitates. This enables stress-strain curves to be generated directly from any real microstructure, with any primary loading direction, which can then be used for calibration of a homogenized model for ballistics calculations such as the GTN model.

This process may also be useful for future microstructural design applications, since it can be used to generate a wide range of potential representative unit cells. These unit cell calculations can be completed quickly to determine a possible "ideal" particle size and spacing, which could then be used as a target for material scientists aiming to manufacture the best microstructure for a particular application.

REFERENCES

[1] J. Besson, "Continuum Models of Ductile Fracture: A Review," vol. 19, no. 1, pp. 3-52, 2010.

[2] A. A. Benzerga, J.-B. Leblond, A. Needleman, and V. Tvergaard, "Ductile failure modeling," *International Journal of Fracture,* vol. 201, no. 1, pp. 29-80, 2016/09/01 2016.

[3] W. M. Garrison and N. R. Moody, "Ductile fracture," *Journal of Physics and Chemistry of Solids,* vol. 48, no. 11, pp. 1035-1074, 1987/01/01/ 1987.

[4] V. Tvergaard, "Material Failure by Void Growth to Coalescence," in *Advances in Applied Mechanics*, vol. 27, J. W. Hutchinson and T. Y. Wu, Eds.: Elsevier, 1989, pp. 83-151.

[5] A. A. Benzerga and J.-B. Leblond, "Ductile Fracture by Void Growth to Coalescence," in *Advances in Applied Mechanics*, vol. 44, H. Aref and E. v. d. Giessen, Eds.: Elsevier, 2010, pp. 169-305.

[6] M. Gologanu, J.-B. Leblond, G. Perrin, and J. Devaux, "Recent Extensions of Gurson's Model for Porous Ductile Metals," in *Continuum Micromechanics*, P. Suquet, Ed. Vienna: Springer Vienna, 1997, pp. 61-130.

[7] T. Pardoen and J. W. Hutchinson, "Micromechanics-based model for trends in toughness of ductile metals," *Acta Materialia,* vol. 51, no. 1, pp. 133-148, 2003/01/08/ 2003.

[8] V. Tvergaard and A. Needleman, "Analysis of the cup-cone fracture in a round tensile bar," *Acta Metallurgica,* vol. 32, no. 1, pp. 157-169, 1984/01/01/ 1984.

[9] J. T. Lloyd, A. J. Matejunas, R. Becker, T. R. Walter, M. W. Priddy, and J. Kimberley, "Dynamic tensile failure of rolled magnesium: Simulations and experiments quantifying the role of texture and second-phase particles," *International Journal of Plasticity,* vol. 114, pp. 174-195, 2019/03/01/ 2019.

[10] D. Mallick, C. Williams, and J. Wilkerson, "A Brief Review of Spall Failure in Pure and Alloyed Magnesium," *Journal of Dynamic Behavior of Materials,* 2020.

[11] J. W. Christian and S. Mahajan, "Deformation twinning," *Progress in Materials Science,* vol. 39, no. 1, pp. 1-157, 1995/01/01/ 1995.

[12] M. T. Pérez-Prado, J. A. del Valle, J. M. Contreras, and O. A. Ruano, "Microstructural evolution during large strain hot rolling of an AM60 Mg alloy," *Scripta Materialia,* vol. 50, no. 5, pp. 661-665, 2004/03/01/ 2004.

[13] A. L. Gurson, "Continuum Theory of Ductile Rupture by Void Nucleation and Growth: Part I—Yield Criteria and Flow Rules for Porous Ductile Media," *Journal of Engineering Materials and Technology,* vol. 99, no. 1, pp. 2-15, 1977.

[14] T. Pardoen and J. W. Hutchinson, "An extended model for void growth and coalescence," *Journal of the Mechanics and Physics of Solids,* vol. 48, no. 12, pp. 2467-2512, 2000/12/01/ 2000.

[15] M. Gologanu, J.-B. Leblond, and J. Devaux, "Approximate models for ductile metals containing non-spherical voids—Case of axisymmetric prolate ellipsoidal cavities," *Journal of the Mechanics and Physics of Solids,* vol. 41, no. 11, pp. 1723-1754, 1993/11/01/ 1993.

[16] M. Gologanu, J.-B. Leblond, and J. Devaux, "Approximate Models for Ductile Metals Containing Nonspherical Voids—Case of Axisymmetric Oblate Ellipsoidal Cavities," *Journal of Engineering Materials and Technology,* vol. 116, no. 3, pp. 290-297, 1994.

[17] M. Gologanu, J.-B. Leblond, G. Perrin, and J. Devaux, "Theoretical models for void coalescence in porous ductile solids. I. Coalescence “in layers”," *International Journal of Solids and Structures,* vol. 38, no. 32, pp. 5581-5594, 2001/08/01/ 2001.

[18] M. Gologanu, J.-B. Leblond, and J. Devaux, "Theoretical models for void coalescence in porous ductile solids. II. Coalescence “in columns”," *International Journal of Solids and Structures,* vol. 38, no. 32, pp. 5595-5604, 2001/08/01/ 2001.

[19] G. Perrin and J.-B. Leblond, "Accelerated void growth in porous ductile solids containing two populations of cavities," *International Journal of Plasticity,* vol. 16, no. 1, pp. 91-120, 2000/01/01/ 2000.

[20] M. Gărăjeu, J. C. Michel, and P. Suquet, "A micromechanical approach of damage in viscoplastic materials by evolution in size, shape and distribution of voids," *Computer Methods in Applied Mechanics and Engineering,* vol. 183, no. 3, pp. 223-246, 2000/03/17/ 2000.

[21] K. Madou and J.-B. Leblond, "A Gurson-type criterion for porous ductile solids containing arbitrary ellipsoidal voids—I: Limit-analysis of some representative cell," *Journal of the Mechanics and Physics of Solids,* vol. 60, no. 5, pp. 1020-1036, 2012/05/01/ 2012.

[22] K. Madou and J.-B. Leblond, "A Gurson-type criterion for porous ductile solids containing arbitrary ellipsoidal voids—II: Determination of yield criterion parameters," *Journal of the Mechanics and Physics of Solids,* vol. 60, no. 5, pp. 1037-1058, 2012/05/01/ 2012.

[23] K. Madou and J.-B. Leblond, "Numerical studies of porous ductile materials containing arbitrary ellipsoidal voids – I: Yield surfaces of representative cells," *European Journal of Mechanics - A/Solids,* vol. 42, pp. 480-489, 2013/11/01/ 2013.

[24] K. Madou, J.-B. Leblond, and L. Morin, "Numerical studies of porous ductile materials containing arbitrary ellipsoidal voids – II: Evolution of the length and orientation of the void axes," *European Journal of Mechanics - A/Solids,* vol. 42, pp. 490-507, 2013/11/01/ 2013.

[25] P. P. Castañeda and M. Zaidman, "Constitutive models for porous materials with evolving microstructure," *Journal of the Mechanics and Physics of Solids,* vol. 42, no. 9, pp. 1459-1497, 1994/09/01/ 1994.

[26] M. Pinz, G. Weber, and S. Ghosh, "Generating 3D virtual microstructures and statistically equivalent RVEs for subgranular gamma-gamma' microstructures of nickel-based superalloys," *Computational Materials Science,* vol. 167, pp. 198-214, 2019/09/01/ 2019.

[27] Y. Jiao, F. H. Stillinger, and S. Torquato, "Modeling heterogeneous materials via two-point correlation functions: Basic principles," *Physical Review E,* vol. 76, no. 3, p. 031110, 09/11/ 2007.

[28] S. R. Niezgoda, D. M. Turner, D. T. Fullwood, and S. R. Kalidindi, "Optimized structure based representative volume element sets reflecting the ensemble-averaged 2-point statistics," *Acta Materialia,* vol. 58, no. 13, pp. 4432-4445, 2010/08/01/ 2010.

[29] A. Hasanabadi, M. Baniassadi, K. Abrinia, M. Safdari, and H. Garmestani, "3D microstructural reconstruction of heterogeneous materials from 2D cross sections: A modified phase-recovery algorithm," *Computational Materials Science,* vol. 111, pp. 107-115, 2016/01/01/ 2016.

[30] V. Sundararaghavan and N. Zabaras, "Classification and reconstruction of three-dimensional microstructures using support vector machines," *Computational Materials Science,* vol. 32, no. 2, pp. 223-239, 2005/02/01/ 2005.

[31] A. Kumar, L. Nguyen, M. DaGraef, and V. Sundararaghavan, "A Markov random field approach for microstructure synthesis," *Modelling and Simulation in Materials Science and Engineering,* vol. 24, 2016.

[32] M. A. Groeber, M. D. Uchic, and D. M. Dimiduk, "A framework for automated 3D microstructure analysis & representation," *Journal of Computer-Aided Materials Design,* vol. 14, pp. 63-74, 2007.

[33] S. Ghosh and D. V. Kubair, "Exterior statistics based boundary conditions for representative volume elements of elastic composites," *Journal of the Mechanics and Physics of Solids,* vol. 95, pp. 1-24, 2016/10/01/ 2016.

[34] S. Ghosh, J. Bai, and D. Paquet, "Homogenization-based continuum plasticity-damage model for ductile failure of materials containing heterogeneities," *Journal of the Mechanics and Physics of Solids,* vol. 57, no. 7, pp. 1017-1044, 2009/07/01/ 2009.

[35] Mathworks. (2019). *regionprops3*. Available: https://www.mathworks.com/help/images/ref/regionprops3.html

[36] E. W. Weisstein. *Central Moment*. Available: http://mathworld.wolfram.com/CentralMoment.html

[37] E. W. Weisstein. *Covariance*. Available: http://mathworld.wolfram.com/Covariance.html

[38] MathWorks. (2019). *Voronoi Diagrams*. Available: https://www.mathworks.com/help/matlab/math/voronoi-diagrams.html

APPENDIX A

MATLAB CODES

Fit of ellipsoid distribution to 2nd phase particle CT scans:

run.m

```
file_names = 'file_names.xlsx'; %file containing list of
image names
size = [471,472,479]; %[[size of binary images] number of
images]


bw = assemble_slice_data(size,file_names);


%fit ellipses
conn = 6; % connectivity paramter: 6 = faces only, 18 =
edges, 26 = corners
CC = bwconncomp(bw2,conn);
[s,angles] = fit_ellipses(CC);
[avg_dist,dist_xyz] = spacing_stats(s);
```

assemble_slice_data.m

```
function [bw] = assemble_slice_data(size,file_names)
%Read slice data and assemble into 3D binary data
%   size: size of image to be assembled (1x3 array)
%   file_names: string name of excel file containing base
image name in column 1 and
%               image numbers in column 2 (including .xlsx
extension)

bw = zeros(size(1),size(2),size(3));

%start with matrix of file names to be read in
names=importdata(file_names);
n = length(names);

count=0;
```

```
for i=1:n
    file = names{i};
    bw1 = imread(file);
    bw2 = logical(floor(double(bw1)./255));
        dp = 1;
        count=count + dp;
        bw(:,:,count)=bw2;
end
end
```

fit_ellipses.m

```
function [s,angles] = fit_ellipses(CC)
%Fit ellipses to binary 3D data
%   bw: 3D binary data
%   s: ellipse statistics
%   angles: rotation angles of ellipses in the order:
%           gamma (about z), beta (about y), alpha (about
x)

s=regionprops3(CC,"Centroid","PrincipalAxisLength","EigenVa
lues","EigenVectors");

n=height(s); %number of voids identified

angles = zeros(n,3);
for i=1:n
    %Update this to get angles for all regions
    eigs = s.EigenVectors{i,1};
    Rnew = zeros(3,3);
    Rnew(1,:)=eigs(2,:);
    Rnew(2,:)=eigs(1,:);
    Rnew(3,:)=eigs(3,:);
    Rnew =Rnew';
    %Calculate rotation angles
    angles(i,:) = rad2deg(rotm2eul(Rnew,'ZYX'));
end

end
```

spacing_stats.m

```
function [avg_dist,dist_xyz] = spacing_stats(s)
%Gets Spacing Statistics of Voids in each direction x y z
```

```
%   For each void centroid - find distance to each other
void centroid
%       dist_xyx: array of distances in each coordinate to
nearest neighbor void
%       min_dist_e: average distance in each coordinate to
nearest neighbor void

n = height(s);

dist_xyz = zeros(n,3);
for i =1:n
    x1 = s.Centroid(i,1);
    y1 = s.Centroid(i,2);
    z1 = s.Centroid(i,3);
    %reinitialize min. distances
    min_dist = inf;
    for j=1:n
        if i==j
            continue
        else
            x2 = s.Centroid(j,1);
            y2 = s.Centroid(j,2);
            z2 = s.Centroid(j,3);
            dist_x = abs(x2-x1);
            dist_y = abs(y2-y1);
            dist_z = abs(z2-z1);
            dist = sqrt(dist_x^2+dist_y^2+dist_z^2);
            if dist < min_dist
                %finds closest void by euclidian dist,
                %then get the x, y, and z spacings.
                min_dist = dist;
                dist_xyz(i,:) = [dist_x dist_y dist_z];
%distance in x,y,z coords to nearest void
            end
        end
    end
end

avg_dist = mean(dist_xyz);


end
```

Voronoi spacing algorithm:

Voronoi_UC.m

```
name = 'coords.xlsx'; %file containing coordinates of void
centers
xy = importdata(name);

ux = 100;
uy= 100;
n_void = 8;

x = xy(:,1)';
y = xy(:,2)';

figure
[VX,VY] = voronoi(x,y);
h = plot(VX,VY,'-b',x,y,'.r');
xlim([-50,ux+50])
ylim([-50,uy+50])

% Assign labels to the points X.
nump = size(x);
plabels = arrayfun(@(n) {sprintf('X%d', n)}, (1:nump)');
hold on

% Compute the Voronoi diagram.
dt = delaunayTriangulation(x',y');
[V,R] = voronoiDiagram(dt);
hold on
triplot(dt,'--r');
hold off


% Assign labels to the Voronoi vertices V.
% By convention the first vertex is at infinity.
numv = size(V,1);
vlabels = arrayfun(@(n) {sprintf('V%d', n)}, (2:numv)');
hold on
axis square
xline(0,'m'); xline(ux,'m');
yline(0,'m'); yline(uy,'m');
hold off
```

```
%Smallest rectangle encompassing voronoi polygon
max_min = zeros(n_void,4);
for i = 1:n_void
    vertices = R{i};
    coords = V(vertices,:);
    max_min(i,1:4) = [max(coords(:,1))  min(coords(:,1))
max(coords(:,2))  min(coords(:,2))];
        % max x   min x   max y   min y
end
cell_max = [max_min(:,1)-max_min(:,2) max_min(:,3)-
max_min(:,4)]./2;

%Rectangle encompassing midpoints of polygon edges
mdpts_extrema = zeros(n_void,4);
for i=1:n_void
    vertices = R{i};
    coords = V(R{i},:);
    n=length(coords);
    mdpts=zeros(n,2); % midpoints x y coords
    for j = 1:n
        if j < n
            mdpts(j,1) = (coords(j,1)+coords(j+1,1))/2;
            mdpts(j,2) = (coords(j,2)+coords(j+1,2))/2;
        else %j=n
            mdpts(j,1) = (coords(j,1)+coords(1,1))/2;
            mdpts(j,2) = (coords(j,2)+coords(1,2))/2;
        end
    end
    mdpts_extrema(i,1:4) = [max(mdpts(:,1))
min(mdpts(:,1)) max(mdpts(:,2))  min(mdpts(:,2))];
        % max x   min x   max y   min y
    if i == 7
        hold on
        plot(mdpts(:,1),mdpts(:,2),'g*')
        hold off
    end
end
cell_mid = [mdpts_extrema(:,1)-mdpts_extrema(:,2)
mdpts_extrema(:,3)-mdpts_extrema(:,4)]./2;


% Unit Cell Dimensions

%Using Max Rectangle
```

```
mean_y = mean(cell_max(:,2));
std_y = std(cell_max(:,2));
mean_x = mean(cell_max(:,1));
std_x = std(cell_max(:,1));
max_unit_cells = [min(cell_max(:,1)) min(cell_max(:,2));
...
                  mean_x-std_x mean_y-std_y; ...
                  mean_x mean_y; ...
                  mean_x+std_x mean_y+std_y; ...
                  max(cell_max(:,1)) max(cell_max(:,2)) ];
max_lambdas =max_unit_cells(:,2)./max_unit_cells(:,1);
%Using Midpoint Rectangle
mean_x = mean(cell_mid(:,1));
std_x = std(cell_mid(:,1));
mean_y = mean(cell_mid(:,2));
std_y = std(cell_mid(:,2));
mid_unit_cells = [min(cell_mid(:,1)) min(cell_mid(:,2));
...
                  mean_x-std_x mean_y-std_y; ...
                  mean_x mean_y; ...
                  mean_x+std_x mean_y+std_y; ...
                  max(cell_mid(:,1)) max(cell_mid(:,2)) ];
mid_lambdas =mid_unit_cells(:,2)./mid_unit_cells(:,1);
```