Efficient Extended Finite Element Algorithms for Strongly and Weakly Discontinuous Entities with Complex Internal Geometries

by

Rui Yuan

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2015 by the
Graduate Supervisory Committee:

Jay Oswald, Chair
Nikhilesh Chawla
Yongming Liu
Kiran Solanki
Kangping Chen

ARIZONA STATE UNIVERSITY

December 2015

ABSTRACT

The objective of this research is to develop robust, accurate, and adaptive algorithms in the framework of the extended finite element method (XFEM) for fracture analysis of highly heterogeneous materials with complex internal geometries. A key contribution of this work is the creation of novel methods designed to automate the incorporation of high resolution data, e.g. from X-ray tomography, that can be used to better interpret the enormous volume of data generated in modern in-situ experimental testing. Thus new algorithms were developed for automating analysis of complex microstructures characterized by segmented tomographic images.

A centrality-based geometry segmentation algorithm was developed to accurately identify discrete inclusions and particles in composite materials where limitations in imaging resolution leads to spurious connections between particles in close contact. To allow for this algorithm to successfully segment geometry independently of particle size and shape, a relative centrality metric was defined to allow for a threshold centrality criterion for removal of voxels that spuriously connect distinct geometries.

To automate incorporation of microstructural information from high resolution images, two methods were developed that initialize signed distance fields on adaptively-refined finite element meshes. The first method utilizes a level set evolution equation that is directly solved on the finite element mesh through Galerkins method. The evolution equation is formulated to produce a signed distance field that matches geometry defined by a set of voxels segmented from tomographic images. The method achieves optimal convergence for the order of elements used. In a second approach, the fast marching method is employed to initialize a distance field on a uniform grid which is then projected by least squares onto a finite element mesh. This latter approach is shown to be superior in speed and accuracy.

Lastly, extended finite element method simulations are performed for the analysis of particle fracture in metal matrix composites with realistic particle geometries initialized from X-ray tomographic data. In the simulations, particles fracture probabilistically through a Weibull strength distribution. The model is verified through comparisons with the experimentally-measured stress-strain response of the material as well as analysis of the fracture. Further, simulations are then performed to analyze the effect of mesh sensitivity, the effect of fracture of particles on their neighbors, and the role of a particles shape on its fracture probability.

This dissertation is dedicated to my mom whose unyielding love, support, and encouragement have enriched my soul and inspired me to complete this research.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Professor Jay Oswald, for his mentoring, guidance, and encouragement over the past four and a half years. He gave me enough freedom to pursue my own ideas while he was always available for discussion and promptly answered my emails even after midnight. I have been incredibly fortunate to have been his student and his contagious enthusiasm, perseverance, and insightful vision for research will always be among the virtues I can learn and benefit in my lifetime.

I would also like to thank my collaborators, Professor Nikhilesh Chawla and Dr. Sudhanshu Shekhar Singh, for their contributions of experimental data used in this work. I have also vastly benefitted from discussions with them.

I would like to thank my thesis committee, Professor Yongming Liu, Professor Kangping Chen, Professor Kiran Solanki and Professor Nikhilesh Chawla, for their advice and participation in my defense.

I would like to thank the help and support from my colleagues at Arizona State University, Xiao Liao, Vipin Agrawal, Yiyang Li, Cheng Lv, Ruijin Cang, Kai Zhou, Tianwei Sun, Benyamin Gholami Bazehhour, Nitin Muthegowda, Scott Turnage, Dr. Mehul Bhatia, and Dr. Ilaksh Adlakha.

Most importantly, I would like to thank my family. Their unyielding love, support, and encouragement made everything easier to achieve.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

This chapter provides an introduction to methods applied in this work for understanding the subsequent discussion of an adaptive computational algorithm for automating simulation of experiments. Section 1.1 gives a review of the recent clustering algorithms for geometry segmentation. A brief introduction of betweenness centrality in graph networks is presented. Section 1.2 reviews different methods for geometry representation. The advantages of implicit geometry representation is discussed and presented. In Section 1.3, a review of methods used in fracture analysis are presented.

1.1 Geometry Segmentation

The study of material structure has been traditionally limited to two-dimensional (2D) analyses [1, 2]. This simplified approach is often inadequate in the analysis of cutting-edge problems such as microstructure characterization of the intermetallic size in Sn-rich solder [3], and the study of physical properties like porosity in soils as well the distribution of pore width [4]. Therefore, there is an increasing demand for improved three-dimensional (3D) visualization and measurement capabilities. The significant increase in data generated by 3D characterization techniques can lead to substantial challenges in dealing with large datasets, and thus robust and automated data processing algorithms must be developed alongside the advances in characterization in order to make such advances useful. This work focuses on an improved method for geometry segmentation applied for analyzing multi-phase tomographic data. The objective of this method is to accurately identify and separate discrete features, e.g. inclusions, from segmented tomography image data for cases where the

resolution of the tomography measurement is insufficient to resolve the space between nearly adjacent features.

Due to limits in the resolution of X-ray tomography, it can be difficult to clearly distinguish between two particles in close proximity, particularly for automated processing algorithms. As a consequence, neighboring but separate particles may erroneously appear to be connected when the resolution of the measurement is insufficient to resolve thin regions between the two adjacent particles. Such spurious connections will generally not affect predictions of average properties of a composite such as the elastic modulus, however, they can have great influence when stochastic behaviors are modeled such as fatigue. Even under a small load, fatigue cracks may nucleate at those thin regions due to fracture of these brittle ligaments. For such simulations, geometry segmentation is a necessary step that comes after image segmentation to achieve reliable numerical predictions. Moreover, since manual segmentation of different geometrical features becomes more and more cumbersome with the increase of resolution and the capability of larger samples, improved algorithms that can robustly distinguish between separate particles are needed. Geometry segmentation can also be meaningful in various fields such as bioengineering [5], solid-earth geophysics [6], and nuclear materials [7].

1.1.1 X-ray Synchrotron Tomography

Experimental techniques for 3D microstructural characterization include atom probe tomography (APT) [8], electron tomography [9], serial sectioning using focused ion beam (FIB) [10] and mechanical polishing followed by optical microscopy [3], and X-ray tomography [11]. Among these, 3D atom probe tomography and TEM tomography provide very high-resolution images (nearly atomic scale), but very small volume can be analyzed. In contrast, serial sectioning and X-ray tomography can be

used to study large volume of materials, resulting in statistically significant information along with the good resolution. While serial sectioning is a powerful technique for generating virtual 3D microstructures, it is very time-consuming and destructive in nature. X-ray tomography is a non-destructive technique with minimal sample preparation [12]. X-ray tomography has been successfully applied to characterize the microstructures in 3D for heterogeneous materials such as metal matrix composites [13], Al alloys [14], and Sn-rich alloys [15]. X-ray tomography can also be used to resolve time-dependent (4D) evolution of a variety of important phenomena such as fatigue [16], and stress corrosion cracking (SCC) [12]. To obtain the quantitative information about the microstructure, image segmentation is performed on X-ray tomography datasets. Post processing of the X-ray tomography dataset is a very time consuming process and therefore it is necessary to segment the data automatically and efficiently.

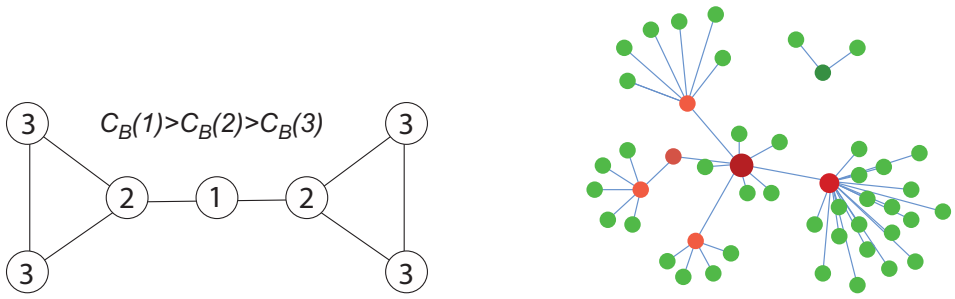
1.1.2 Review of Early Clustering Algorithms

Some partitioning algorithms have been formulated that can be applied in geometry segmentation. Commonly known algorithms include hierarchical clustering schemes [17], k-means clustering [18], spectral clustering [19], and density based clustering methods [20, 21]. Several recent contributions are listed as follows: Ketcham introduced a new computer program to separate touching objects in X-ray tomography data of geological specimens [22]; Peng applied hierarchical clustering to build clusters of genes with similar patterns of expression in the study of bone morphogenetic protein [23]. Stephenson performed the core-linkage clustering algorithm to identify solute-solute interactions in dilute alloys from experimental APT data [24]. There are some insufficiencies of these partitioning algorithms: Hierarchical clustering is limited due to its quadratic time complexity [25]; The number of clusters is a priori

in k-means clustering [26], which is typically an unknown parameter before clustering for realistic graphs; Spectral clustering methods sorts points based on eigendecomposition algorithms whose stability is well studied, but there still has disagreements on the use of eigenvectors in the context of segmentation [27]. Although density based clustering does not require the number of clusters as a priori, the algorithm is not entirely deterministic and data with large differences in densities cannot cluster with satisfied quality [28].

1.1.3 Betweenness Centrality

In this work, we develop a new efficient algorithm for geometry segmentation. The algorithm is based on the concept of betweenness centrality, which was first proposed by Freeman [29] to measure the importance of a node in the study of human communication. In a graph composed of a set of nodes and edges connecting the nodes, the betweenness centrality is the measure of the number of shortest paths between nodes that traverse a given node. The nodes in the graph are associated with segmented voxels and edge weights can be applied to define the dissimilarity between neighbor voxels. More recently, this concept has found a far more general application and has been applied to many problems, including ecological studies of pollination networks [30], network mapping of the human brain [31] and biological studies of protein networks [32].



(a) (b)

Figure 1.1: An illustration of betweenness centrality as a measure of the relative importance of nodes in a graph network.

Betweenness centrality is a commonly used metric that can capture the relative importance of a node in the overall network. In other words, betweenness centrality quantifies the number of times a node acts as a bridge in the communication between two other nodes. The betweenness centrality of vertex v in a graph $G := (V, E)$ with V vertices and E edges is defined as [29, 33]

$$\mathcal{C}_{\mathcal{B}}(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{1.1}$$

Where σ_{st} is the number of shortest paths between each vertex $s \in V$ and $t \in V$, and $\sigma_{st}(v)$ is the number of shortest paths passing through vertex v.

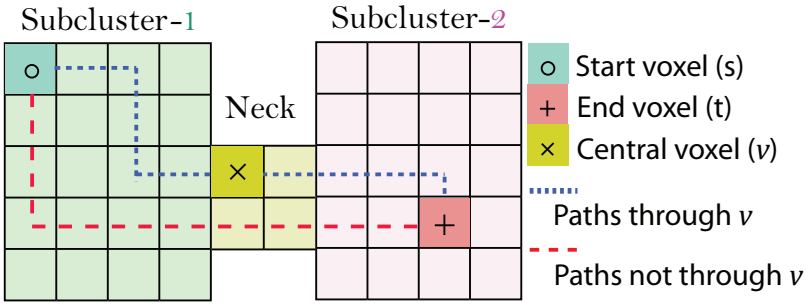


Figure 1.2: An illustration of the shortest path used in the calculation of betweenness centrality. Two typical paths are shown in dashed lines for a voxel pair (s, t).

The maximum possible absolute centrality for an undirected network with n vertices is [34]

$$\max(\mathcal{C}_{\mathcal{B}}) = \frac{n^2 - 3n + 2}{2} \tag{1.2}$$

The normalized betweenness centrality can be expressed as

$$\mathcal{C}_{\mathcal{B}}^{\mathcal{N}}(v) = \frac{2\mathcal{C}_{\mathcal{B}}(v)}{n^2 - 3n + 2} \tag{1.3}$$

In the context of geometry segmentation in material characterization, betweenness centrality is an effective tool for locating narrow connecting regions between microstructures in segmented image data. Due to the resolution limit of tomography techniques and imperfection of image segmentation, approximate but separate clusters may erroneously connect by some artificially introduced voxels. Those artificial voxels that bridge clusters have an elevated level of importance in binding the network of the combined cluster and, consequently, they contain unusual higher betweenness centrality values. Betweenness centrality can be an effective metric to eliminate spurious bridges between separate geometric entities.

Moreover, the centrality based geometry segmentation can be very efficient and feasible for realistic networks with millions of vertices. Calculating the betweenness centrality of all the vertices can be very time consuming if we directly compare all possible paths between all pairs of vertices on a graph. The direct calculation of between centrality involves calculating the shortest paths between all pairs of vertices on a graph, which requires $O(V^3)$ time with the Floyd-Warshall algorithm [35]. However, on sparse graphs, the calculation time decreases to $O(V^2 \log V + VE)$ with the help of Johnson's algorithm [36]. Fortunately, with the help of Brandes' algorithm [37], the centrality calculation requires $O(VE)$ time on unweighted graphs [38] compared with $O(V^3)$ time, originally. The V and E are the numbers of vertices and edges in the graph network, respectively.

1.2 Review of Methods for Geometry Representation

The assumption of material homogeneity in the stress analysis of engineered components and structures is necessitated by the vast separation of scales between the characteristic length of material microstructure and the dimensions of the part being analyzed. This assumption is appropriate when the constitutive response of the homogenized material is well-characterized and the characteristic length of the deformation is also far separated from the material microstructure scale [39]. This assumption is not appropriate when there is coupling between the microstructure and the macroscopic response. Examples of microstructure behavior that play an important role in macroscale material response are fiber bucking in reinforced polymer composites [40], shear localization in sand and soils [41], and fatigue crack initiation at precipitates in alloys [42]. In this work, we develop a level set technique for modeling materials with an arbitrary number of phases by representing their complex two- and three-dimensional microstructure implicitly with distance functions on adaptively refined meshes with hanging nodes.

This section provides some basic concepts of numerical methods for geometry representation. Section 1.2.1 reviews some previous work in explicit or parametric geometry representation. Section 1.2.2 provides some basic concepts of the level set method. In section 1.2.3, a review of the fast marching method is presented.

1.2.1 Explicit or Parametric Method

The most common method of representing material microstructure in finite element models is by an explicit or parametric approach. In this approach, an explicit representation of the material microstructure is fed into mesh generation software

so that each phase meshes by a separate set of elements and material interfaces are coincident with element faces.

There has been much progress in developing algorithms for generating high-quality, unstructured meshes in arbitrary geometry. Zhang et al. have developed an octree-based isocontouring method to construct adaptive tetrahedral or hexahedral meshes from volumetric data [43]. In their later work [44], material change edge was introduced to relocate material interfaces for automatic 3D mesh generation of multi-phase materials.

However, these approaches require sophisticated unstructured mesh generation, which complicates the imposition of periodic boundary conditions and ultimately couples the representation of the geometry to a specific mesh. Explicitly representing material microstructure through mesh topology poses the restriction that interfaces must not evolve with respect to the material coordinates, an occurrence that can be either physical, e.g. grain growth, or numerical, e.g. convection of interfaces in an arbitrary Lagrangian/Eulerian formulation for extreme deformations. Evolving geometry can also be modeled by remeshing as the interface advances. A recent advance in this area is the development of universal meshes, which robustly track complex moving interfaces in two dimensions by mapping from a fixed background mesh.

1.2.2 Level Set Method

Significant effort has been expended to develop level set methods (LSM) that address the drawbacks of explicit or parametric representations of interfaces. The LSM was created in 1988 by Osher and Sethian [45] as a numerical technique for tracking the propagation of an interface or front whose speed depends on the local curvature. Tracking the motion of an interface is meaningful in a wide range of physical phenom-

ena. Some applications are obvious, for example, the analysis of flame propagation [46], the computation of incompressible bubble dynamics [47], and tracking the motion of complex solid/liquid boundaries in crystal growth and dendritic solidification [48, 49]. Some applications are not so obvious, for example, path optimization in robotic navigation with constraints [50] and the manufacture of integrated circuits [51].

One attractive feature of the method is that it allows the topology of the interfaces to change (e.g. merging or splitting) automatically during evolution [52, 53]. The LSM allows one to follow the motion of an N-dimensional surface Γ in a N+1 dimensional space Ω [54, 55]. The interface Γ should subdivide the domain Ω into two or more separate regions, e.g. a sphere subdivides an infinite domain into two separate regions, one inside the sphere and the other one outside the sphere. One dimensional curves cannot divide three-dimensional space into separate sub-domains, and co-dimension two is not considered in this method. In the LSM, interfaces are represented by the zero value of a higher-dimensional function, e.g. $\phi(\mathbf{x}, t) = 0$ where $\mathbf{x} \in \Gamma$. The level set function can be any continuous function as long as it is zero on interfaces. Given an initial $\phi = 0$ at $t = 0$, it would be possible to know at any time t, the motion equation is

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = 0 \tag{1.4}$$

After applying chain rule, equation (1.4) becomes

$$\phi_t + \frac{\partial \phi}{\partial x} x_t = 0 \tag{1.5}$$

$$\phi_t + \nabla \phi \, x_t = 0 \tag{1.6}$$

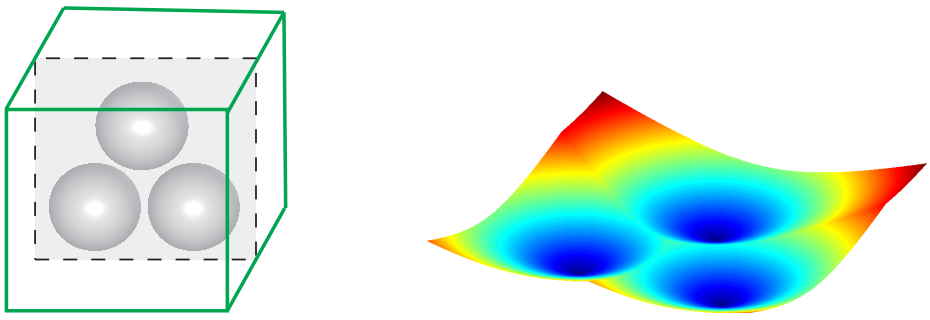
$$\phi_t + \nabla \phi \, F(x, t) n = 0 \tag{1.7}$$

$$\phi_t + \nabla \phi \, F(x, t) \frac{\nabla \phi}{|\nabla \phi|} = 0 \tag{1.8}$$

$$\phi_t + F(x, t) |\nabla \phi| = 0 \tag{1.9}$$

In the above equations, $F(x,t)$ is a velocity field function that defines the motion of the interface along its normal direction. The speed may be an arbitrary function of the curvature. The evolution of fronts or surfaces can be accomplished by solving the Hamilton-Jacobi equation [56, 57]. The velocity field can be chosen so that the evolution equation reaches a stationary point when the level set function is a signed distance function. A signed distance function is a special case of a level set function where the value of the field gives the nearest distance to an interface and the sign of the field determines whether or not a coordinate is interior or exterior to the interface. By definition, the magnitude of the gradient of a signed distance function is unity, so the velocity field is defined to enforce this condition. Figure (1.10) shows the distance field on a slice in a triple sphere model. The distance field is negative inside spheres and positive outside the spheres. The distance field equals to zero on the sphere surfaces. The level set field is enforced to satisfy the distance criterion if the velocity field is defined as

$$F = \text{sign}(\phi)\left(1 - \frac{1}{|\nabla\phi|}\right) \tag{1.10}$$



(a) Triple sphere model. (b) Distance field on the slice.

Figure 1.3: An illustration of the distance field in a triple sphere model.

1.2.3 Fast Marching Method

The fast marching method (FMM) was introduced by Sethian [58] and improved for higher order convergence by Chopp [59] as an efficient method to update the distance fields for monotonically advancing fronts and moving interfaces. This method has been successfully applied in various areas including digital inpainting for reconstruction of small damaged portions of an image [60], fatigue crack propagation [61, 62], and path planning for mobile robot navigation [63]. In FMM, an interface is represented implicitly as

$$t = \phi(\mathbf{x}) \tag{1.11}$$

Differentiating equation (1.11) with respect to t and recall that $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$, we have

$$1 = \nabla\phi \cdot (F(\mathbf{x})\mathbf{n}) = F(\mathbf{x})|\nabla\phi(\mathbf{x})| \tag{1.12}$$

where $F(\mathbf{x})$ is a monotonic speed function for a propagating interface or front. This method is vastly faster than any other numerical method by an order of magnitude. The interface evolution is computed in one pass over the mesh with computational complexity of $O(N \log N)$, where N is the total number of grid points in the domain [64]. Besides that, the FMM uses techniques incorporated from hyperbolic conservation laws to appropriately propagate interfaces or fronts with sharp corners [59], such that we do not need to worry about the stability problem in the evolution of the level set field.

In FMM, the gradient $\nabla\phi$ in equation (1.12) is approximated using upwind differences method. The upwind differences method requires a causality, in other words, information always flows from the initial contour $\phi(\mathbf{x}) = 0$ outward for increasing values of ϕ [65]. The causality means that the location of the interface at time t_0 must come from time t_1 for that $t_0 \geq t_1$. Hence, if equation (1.11) is solved in a monotonically increasing way, the upwind differences are assured to be valid and the

level set ϕ is eventually computed on all grid points. The causality of grid points is illustrated in Figure (1.4).

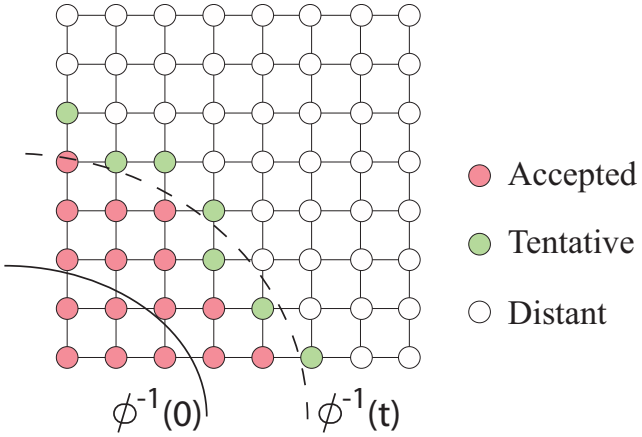


Figure 1.4: An illustration of causality for the fast marching method.

As shown in Figure (1.4), grid points are sorted into three separate sets: the set of *accepted* points A, the set of *tentative* points T and the set of *distant* points D. The set A contains all the grid points whose level set have been calculated. The grid points in set A are always closer to the initial interface than any of the grid points in set T or D. The grid points in T are all potential candidates to be next computed. We move a *tentative* point, $\mathbf{x}$, to the set A once its level set $\phi(\mathbf{x})$ is computed. The grid points in D are considered far away from the initial interface. In upwind differences method, the information always propagates from the initial contour outward to larger values of ϕ. Thus, for grid points $\mathbf{x}_a \in A$, $\mathbf{x}_t \in T$ and $\mathbf{x}_d \in D$, the evolved level set should satisfy $\phi(\mathbf{x}_a) < \phi(\mathbf{x}_t) < \phi(\mathbf{x}_d)$.

The fast marching method can be summarized as an algorithm [66]:

1. Initialize all grid points in a neighborhood of the initial interface with an initial value. These points are put in A.

2. Compute tentative approximations for all grid points that are adjacent to an *accepted* point but not already in A. These points are put in T. All remaining points

are placed in the set D. Grid points are considered as neighbors if their distance is only one space step away along one dimension. As shown in Figure (1.5), grid points in mutual diagonal positions are not considered as adjacent.

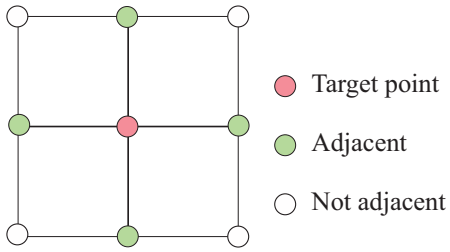


Figure 1.5: An illustration of adjacent grid points for a target point in the fast marching method.

3. Choose the point $\mathbf{x} \in T$ with the smallest tentative values. Move the point to A. For all grid points $\mathbf{y} \notin A$ that are adjacent to $\mathbf{x}$, compute tentative values for $\phi(\mathbf{y})$. If $\mathbf{y} \in D$, then move point $\mathbf{y}$ into T.

4. Go to step 3 until $T = \emptyset$.

Using upwind finite differences, equation (1.11) can be discretized in the direction controlled by which neighboring points are in $A \cup T$. One of the key components in fast marching method is to estimate the tentative values for point in T. Suppose, for example, a grid point $\mathbf{x}_{i,j} \in T$ is to have a tentative value computed and assume adjacent grid points $\mathbf{x}_{i+1,j}, \mathbf{x}_{i,j-1} \in A$. Then, the two-dimensional first order approximation [59] for equation (1.11) is

$$F_{i,j}\left(\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}\right)^2 + \left(\frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y}\right)^2\right)^{1/2} = 1 \tag{1.13}$$

In the above equation, the terms $F_{i,j}$, $\phi_{i+1,j}$, and $\phi_{i,j-1}$ are all known. Solving for $\phi_{i,j}$, we get a quadratic equation:

$$\left(\frac{\Delta x}{\Delta y} + \frac{\Delta y}{\Delta x}\right)\phi_{i,j}^2 - 2\left(\frac{\Delta y}{\Delta x}\phi_{i+1,j} + \frac{\Delta x}{\Delta y}\phi_{i,j-1}\right)\phi_{i,j} + \left(\frac{\Delta y}{\Delta x}\phi_{i+1,j}^2 + \frac{\Delta x}{\Delta y}\phi_{i,j-1}^2 - \frac{\Delta x \Delta y}{F_{i,j}^2}\right) = 0 \tag{1.14}$$

For this quadratic equation, its discriminant is

$$d = 4\left(\frac{\Delta x^2 + \Delta y^2}{F_{i,j}^2} - \left(\phi_{i+1,j} - \phi_{i,j-1}\right)^2\right) \tag{1.15}$$

Since $F_{i,j} > 0$, if discriminant $d < 0$, we have

$$\frac{\sqrt{\Delta x^2 + \Delta y^2}}{F_{i,j}} < |\phi_{i+1,j} - \phi_{i,j-1}| \tag{1.16}$$

Figure (1.6) is helpful to understand the above equation. The discriminant becomes negative when the travel time from point $\mathbf{x}_{i+1,j}$ to point $\mathbf{x}_{i,j-1}$ is smaller than the difference of their level sets, $|\phi_{i+1,j} - \phi_{i,j-1}|$. Then, when the discriminant is negative, the proper value for $\phi_{i,j}$ should only depend on the grid point whose information arrives at $\mathbf{x}_{i,j}$ well before the other one.

$$\phi_{i,j} = \min\left(\phi_{i+1,j} + \frac{\Delta x}{F_{i,j}},\ \phi_{i,j-1} + \frac{\Delta y}{F_{i,j}}\right) \tag{1.17}$$

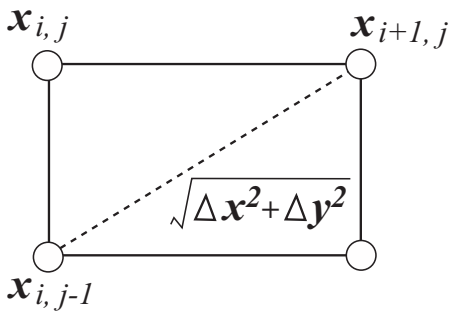


Figure 1.6: An illustration of propagation time between two grid points.

If the discriminant is not less than zero, we have two solutions:

$$\phi_{i,j} = \frac{2\left(\frac{\Delta y}{\Delta x}\phi_{i+1,j} + \frac{\Delta x}{\Delta y}\phi_{i,j-1}\right) \pm 2\sqrt{\frac{\Delta x^2+\Delta y^2}{F_{i,j}^2} - \left(\phi_{i+1,j} - \phi_{i,j-1}\right)^2}}{2\left(\frac{\Delta x}{\Delta y} + \frac{\Delta y}{\Delta x}\right)} \tag{1.18}$$

In order to satisfy $\phi_{i,j} > \phi_{i+1,j}$, the new estimate for $\phi_{i,j}$ is given by the largest of the two roots.

1.3 Fracture Analysis of Metal Matrix Composites

Metal matrix composites (MMCs) are widely used industrially due to their attractive combination of high strength, high stiffness and low density [67]. Efficient numerical analysis of the interrelations between microstructure and strength of these composites can serve as a basis in material design and optimization. Traditionally, the simulation of damage in MMCs has been limited to two dimensions with the reinforcing particles simplified as spheres or ellipsoids [68, 69, 70, 71]. These simplifications do not strongly influence predictions of average properties of a composite such as the elastic modulus, however, they introduce large errors in the prediction of phenomena dominated by extreme values, such as fracture and fatigue. The actual geometries of reinforcing particles may contain sharp corners and edges, which cause stress concentrations that act to initiate fracture. Furthermore, the spatial distribution of reinforcing particles can be a significant factor in the strength and toughness of the material. In this work, we present an extended finite element algorithm for the fracture analysis of MMCs with realistic internal geometries initialized from X-ray tomography data.

1.3.1 Fundamentals of Fracture Mechanics

Fracture mechanics is the field of mechanics that studies the initialization and propagation of cracks in materials [72]. Analytical and experimental solid mechanics are used to characterize the material resistance to fracture. There are three types of loading to enable a crack to propagate [73], as illustrated in Figure (1.8). In mode I loading, a principal load is applied normal to the crack plane to open the crack. Mode II refers to an in-plane shear loading and inclines to slide the crack surface. Mode III corresponds to an out-of-plane shear loading that acts parallel to the plane

of the crack and parallel to the crack front. A crack can be loaded in any one of the three modes, or a combination of two or three modes.

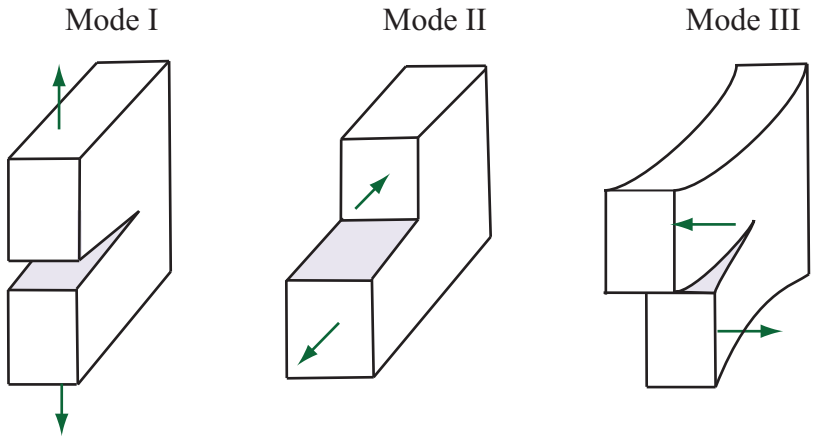


Figure 1.7: Three types of loading that can be applied to a crack.

There are two approaches to fracture analysis: the energy criterion [74, 75] and the stress-intensity approach [76, 77]. These two approaches are equivalent in certain circumstances. The energy approach states that the crack propagation happens when the energy available for crack extension is sufficient to overcome the resistance of the material [72]. The material resistance includes the surface energy of crack surfaces, plastic dissipation, or other types of energy dissipation associated with a growing crack. The energy criterion for fracture was firstly proposed by Griffith [78], to explain the failure of brittle materials. Griffith's theory has excellent agreement with experimental data for brittle materials, however, for ductile materials, the surface energy calculated by Griffith's model is usually unrealistic high.

Irwin and his coworkers [79, 80] found the important role of plasticity in the fracture of ductile materials. In ductile materials, there is a plastic zone around the crack tip of the crack. With increasing external load, the plastic zone increases in size until the crack propagates. Once crack propagates, the region behind the crack tip starts to unload. The cycles of plastic loading and unloading result in

energy dissipation as heat. Therefore, a dissipative energy term has to be added to the energy balance relation. In Irwin's strategy, the energy is partitioned into two parts: the stored elastic strain energy and the dissipated energy which includes plastic dissipation and the crack surface energy. The energy release rate G can be used to represent the rate of change in potential energy with the crack area. Finally, the energy release rate required for crack growth can be represented as

$$G = 2(\gamma_s + \gamma_p) \tag{1.19}$$

where γ_s is the surface energy of the material and γ_p is the plastic work per unit area and typically much larger than γ_s. For a crack of length $2a$ in an infinite plate subject to a remote tensile stress, the energy release rate is

$$G = \frac{\pi\sigma^2 a}{E} \tag{1.20}$$

where E is Young's modulus, σ is the remotely applied stress, and a is the half-crack length. Once the energy release rate equals to a critical value, the crack propagates. The critical energy release rate can be denoted as

$$G_c = \frac{\pi\sigma_f^2\, a}{E} \tag{1.21}$$

Combine equation (1.19), equation (1.20) and equation (1.21), we have

$$\sigma_f = \Big(\frac{2E(\gamma_s + \gamma_p)}{\pi a}\Big)^{1/2} \tag{1.22}$$

The stress intensity factor, K, can be used to predict the stress state near the tip of a crack subjected to external forces. It is originally developed for isotropic linear elastic materials. The magnitude of K usually depends on the sample geometry, the location of the crack tip, the area of the crack surface, and the magnitude of the external load. In a polar coordinate system (r, θ) with origin at the crack tip, the

stress field in any linear elastic crack body is given by [81, 82]

$$\sigma_{ij} = \left(\frac{k}{\sqrt{r}}\right) f_{ij}(\theta) + \sum_{m=0}^{\infty} A_m r^{\frac{m}{2}} g_{ij}^{(m)}(\theta) \tag{1.23}$$

where σ_{ij} is the stress tensor and $f_{ij}(\theta)$ is a dimensionless function that varies with the load and geometry. The second part in equation (1.23) is a higher order term. A_m is the amplitude and $g_{ij}^{(m)}$ is a dimensionless function of θ for the m^{th} term. This relation is no longer true close to the crack tip because as r goes to zero, the stress σ_{ij} goes to infinity. Plastic distortion usually occurs at the crack tip and the linear elastic solution is no longer applicable.

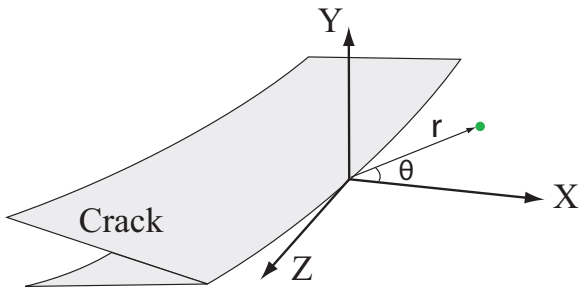


Figure 1.8: Polar coordinates at the crack tip.

The stress intensity factor for a through crack of length $2a$ in an infinite plate subjected to a remote tensile stress σ is

$$K_I = \sigma\sqrt{\pi a} \tag{1.24}$$

1.3.2 Weibull Distribution

The Weibull distribution is a continuous probability distribution that has been proved to be a well empirical statistical distribution for cleavage fracture of brittle materials [83, 84, 85, 86]. Eckschlager proposed a finite element based approach for modeling brittle cleavage of spherical particles on the basis of Weibull distribution [87]. Doremus compared normal, Weibull and Type I extreme value distributions

using failure strengths of glass [88]. Lu fitted fracture strength data to Weibull and normal distributions for three types of brittle materials [89], from which he pointed out that the difference between the two distributions was very small to be clearly distinguished in the case of SiC. Gao proposed a new way to calibrate Weibull stress parameters analytically which uses fracture toughness data under both low and high constraint conditions at the crack front [90]. In his later work [91], the new strategy was used to calibrate the Weibull stress model used to predict cleavage fracture in plates containing surface cracks.

Survival probability is used to denote the possibility of a test sample that does not break under an applied tensile stress. The survival probability can be denoted as [92]

$$P_s = \exp\left(-\left(\frac{\sigma - \sigma_u}{\sigma_f}\right)^m\right) \tag{1.25}$$

where σ_f is the nominal ultimate stress of the material and σ_u is a threshold stress parameter, which represents a minimum stress below which a test specimen does not break [93]. The threshold parameter, σ_u, is usually set to zero. A constant parameter m is used to represent how rapidly the strength decreases as increasing applied stress σ. Figure (1.9) shows the survival probability distribution with respect to the applied tensile stress. When applied tensile stress is zero, the survival probability is one, which means the sample will be sure to survive. The survival probability decreases to zero as the tensile stress increases to infinity. The survival probability is about 0.37 when the applied stress equals to the nominal ultimate stress, $\sigma = \sigma_f$.

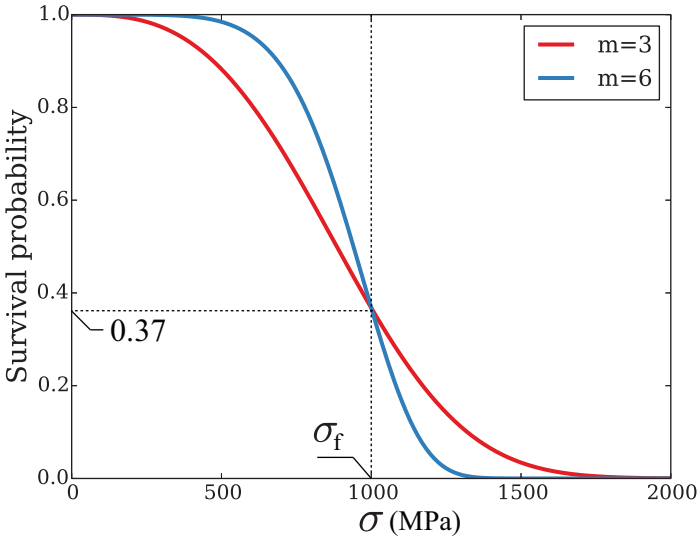


Figure 1.9: The distribution of Weibull survival probability.

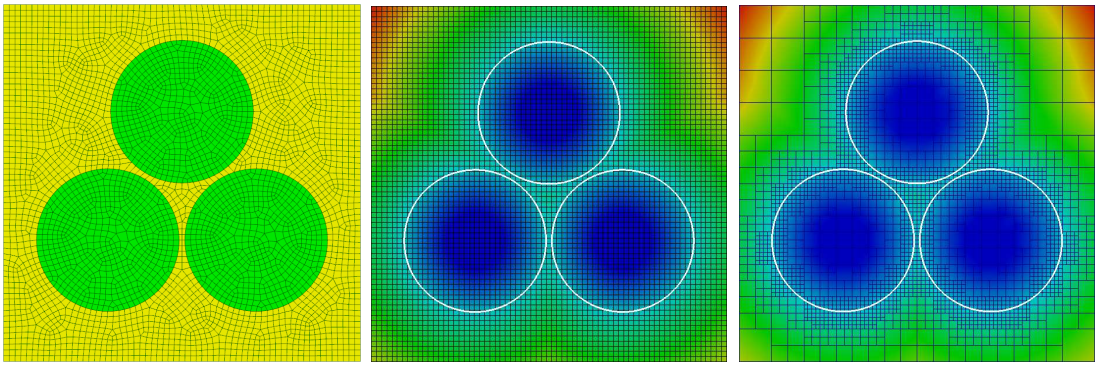
Then, for a uniformly stressed volume in a state of uniaxial tension, the Weibull fracture probability can be expressed as

$$P_f = 1 - \exp\left(-\left(\frac{\sigma - \sigma_u}{\sigma_f}\right)^m\right) \tag{1.26}$$

1.3.3 The Extended Finite Element Method

Implicit microstructure representation combined with the extended finite element method (XFEM) allows modelers to represent complex material microstructures with consistent mesh quality and accuracy. In this approach, material geometry can be implicitly described by a level set field. A signed distance function is a special case of level set field where the value of the field gives the nearest distance to an interface and the sign of the field determines whether or not a coordinate is interior or exterior to the interface [94]. We have presented two methods in our previous work [95] for the initialization of signed distance functions on adaptively-refined finite element meshes from segmented X-ray tomographic data.

The extended finite element method was developed by Belytschko et al. [96] in order to model crack propagation without needing to remesh the domain as the crack grows. The central idea of the XFEM is to use enrichments to the finite element basis that can locally represent discontinuities or other known features of the problem through a partition of unity [97]. Huynh and Belytschko presented further developments in the XFEM for fracture problems in composite materials [98]. Ye and Cheng integrated the XFEM in ABAQUS Subroutine to study the influence of reinforcing particles to the crack propagation behavior in a metal-matrix composite [99]. Wang and his coworkers investigated the interaction between a propagating crack and a single or multiple particles in a brittle matrix using the XFEM technique [100].



(a) Unstructured mesh. (b) Uniform mesh. (c) Oct-tree mesh.

Figure 1.10: A comparison of three types of mesh for a triple sphere model. (a). An unstructured mesh used in the standard finite element method. (b). Contour plot of the level set field on a uniform mesh. Interfaces are represented implicitly with zero level set. (c). Contour plot of the level set field on an oct-tree mesh. Elements close to interfaces are sorted and refined.

Chapter 2

GEOMETRY SEGMENTATION

In this work, we develop a robust method to automate geometry segmentation in segmented tomographic images of three-dimensional materials with discrete inclusions or particles. The objective of this method is to accurately identify and separate discrete features in composite materials where limitations in imaging resolution lead to spurious connections near close contacts. In the method, the betweenness centrality, a measure of the importance of a node with respect to the connectivity of a network, is used to identify voxels that create spurious bridges between distinct geometric features. To facilitate automation of the new algorithm, we develop a non-dimensional relative centrality metric to allow for the selection of a threshold criteria that is independent of inclusion shape or volume.

2.1 Voxel Data

In a heterogeneous material, each three-dimensional phase can approximated by a set of voxels contained within the phase.

$$v_{ijk}^{\alpha} = \begin{cases} 1 & \text{if} \quad \mathbf{x}_{ijk} \in \Omega_{\alpha} \\ 0 & \text{else} \end{cases}, \tag{2.1}$$

where Ω_{α} denotes the domain for phase α. The location of each voxel is described by a 3-tuple of integers such that

$$\mathbf{x}_{ijk} = \mathbf{x}_0 + i\,\boldsymbol{\Delta}_{\boldsymbol{x}} + j\,\boldsymbol{\Delta}_{\boldsymbol{y}} + k\,\boldsymbol{\Delta}_{\boldsymbol{z}}, \tag{2.2}$$

where $\mathbf{x}_0$ is the origin of the voxel data set. The voxel cell length in three directions is determined by the lengths of the voxel cell vectors, i.e. $\|\boldsymbol{\Delta}_{\boldsymbol{x}}\|$, $\|\boldsymbol{\Delta}_{\boldsymbol{y}}\|$ and $\|\boldsymbol{\Delta}_{\boldsymbol{z}}\|$.

2.2 Initial Clustering of Voxels into Clusters

Starting from a phase voxel data set, initial clustering is performed to rapidly split phase voxels into clusters based on their relative spatial connections. This initial clustering is performed by grouping phase voxels into non-contiguous groups of voxels. Voxels are considered contiguous if they share a face, but not if they share an edge or corner, i.e. each voxel has only six connected neighbors. As a result of limited imaging resolution or poor contrast between adjacent particles, these initial sets of voxels often contain more than a single particle, connected through spurious bridge connections. Geometry segmentation will be applied to detect and eliminate those spurious connections. The initial clustering algorithm can be summarized as follows: in a loop of all phase voxels, if a voxel is still not accepted by any cluster, initialize a new cluster and incorporate the voxel into that cluster. Then phase voxels adjacent to the voxel are then added into the same cluster. The voxel absorption continues until all phase voxels pertaining to the cluster have been found. Then the algorithm continues iterating over the set of phase voxels until another unaffilated voxel is found, and a new cluster is created. This procedure continues until all phase voxels are mapped to clusters. This procedure is detailed in Algorithm 1.

Algorithm 1 Fast clustering from phase voxels

Define $\mathcal{V} \rightarrow$ phase voxel set

Define $\mathcal{C} \rightarrow$ cluster set

$\tau \leftarrow$ empty array

for all $v \in \mathcal{V}$ **do**

  if $v \in \mathcal{C}$ **then**

    Continue

  $\mathcal{C}_0 \leftarrow$ empty cluster

  Insert v into τ

  while $\tau \neq \emptyset$ **do**

    $\kappa \leftarrow$ last element in τ

    $\tau \leftarrow$ delete last element

    if $\kappa \notin \mathcal{C}$ **then**

      $\mu \leftarrow$ phase neighbor voxels of κ

      $\tau \leftarrow$ insert μ

      Insert κ into $\mathcal{C}_0$

  Insert $\mathcal{C}_0$ into $\mathcal{C}$

2.3 Graph Construction

After initial clustering, a set of clusters are initialized and each cluster is composed of a set of phase voxels. We construct an undirected graph network composed of vertices and edges for each cluster. Each phase voxel is constructed as a vertex and edges are constructed between all adjacent phase voxels. The constructed graph network does not incorporate the physical coordinates of each voxel, and the edge weights in the graph do not correspond to the real distance between voxels. In this

work, an unweighted graph network is applied and the edges connecting vertices are considered as binary interactions.

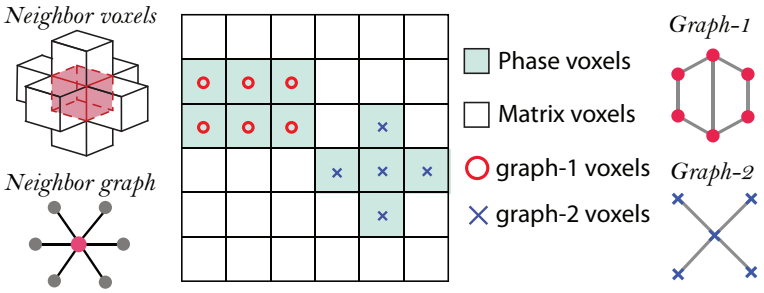


Figure 2.1: An illustration of graph initialization from phase voxels. Phase voxels cluster into two separate graphs since voxels in mutual diagonal position are not considered as neighbors.

2.4 Betweenness Centrality

In order to locate neck voxels, the betweenness centrality is calculated on each node in the constructed graph network. In a graph $G := (V, E)$ with V vertices and E edges, the betweenness centrality of vertex v is defined as [29, 33]

$$\mathcal{C}_{\mathcal{B}}(v) = \sum_{s \neq t \neq v \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2.3}$$

Where σ_{st} is the number of shortest paths between each vertex $s \in V$ and $t \in V$, and $\sigma_{st}(v)$ is the number of shortest paths passing through vertex v. The maximum possible absolute centrality for an undirected network with n vertices is [34]

$$\max(\mathcal{C}_{\mathcal{B}}) = \frac{n^2 - 3n + 2}{2} \tag{2.4}$$

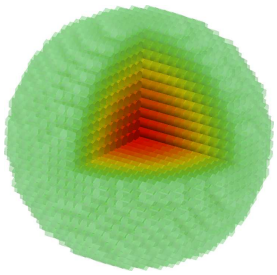
The normalized betweenness centrality can be expressed as

$$\mathcal{C}_{\mathcal{B}}^{\mathcal{N}}(v) = \frac{2\mathcal{C}_{\mathcal{B}}(v)}{n^2 - 3n + 2} \tag{2.5}$$

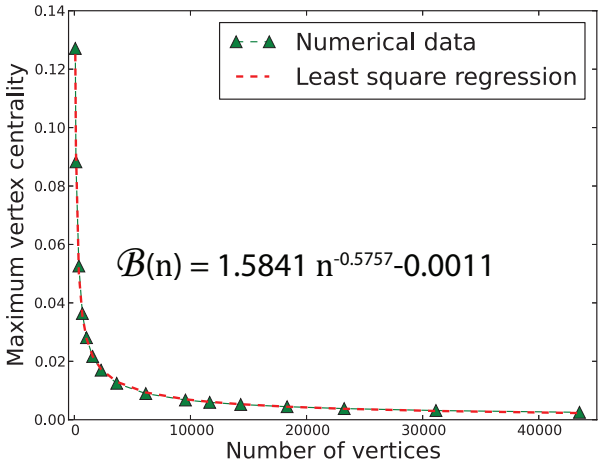
2.5 Automatic Neck Volume Localization

On the basis of constructing graphs with the same rule, graphs with more vertices are more likely to possess higher maximum centrality. Therefore, the absolute centrality can be influenced by two important aspects: the graph size and the corresponding network topology. For a cluster composed of two particles, we can try different threshold and remove vertices with absolute centrality higher than the threshold. The trial process is tedious but feasible for a single cluster. However, for realistic tomography data with hundreds of clusters, a more robust segmentation algorithm should be able to distinguish touching clusters and perform segregation automatically regardless of the cluster size. Compared with the absolute centrality, the normalized centrality is a better choice which can counteract the influence of graph size to a certain extent.

In an one-dimensional graph composed of aligned vertices, it is not difficult to understand that the center vertex possesses the maximum centrality and the maximum value is exactly 0.5 for graphs with odd number of vertices. Similarly in two- and three-dimensions, the maximum centrality also inclines to appear at the graph center unless the graph has special configuration such as neck volume. Figure 2.2a shows the betweenness centrality distribution in a sphere. Centrality decreases gradually from center to the outside surface. Figure 2.2b is a plot of the maximum normalized centrality against the number of vertices in spheres. The maximum normalized centrality is fitted by least square regression with respect to the number of vertices in the network.



(a) Centrality in a sphere



(b) Maximum normalized centrality

Figure 2.2: (a). Scaled betweenness centrality on vertices in a sphere colored by the blue to red rainbow. Inside vertices contain higher centrality values. (b). Least square regression of the maximum normalized centrality in spheres with the different number of vertices.

For an advanced segmentation algorithm, firstly it should decide whether a network possesses neck voxels and if it does, segmentation will be performed automatically. The maximum normalized centrality in a single sphere provides a reference for the establishment of critical centrality values in segmentation. For a graph with unknown topology, if its maximum normalized centrality is close to the centrality in a single sphere with an equal number of vertices, we can deduce that the graph does not contain any neck volume and no segmentation needs to be applied. Until now, we still need to develop a general rule to set the critical centrality. The following part studies the centrality in a couple sphere with two spheres connected by a small contact volume.

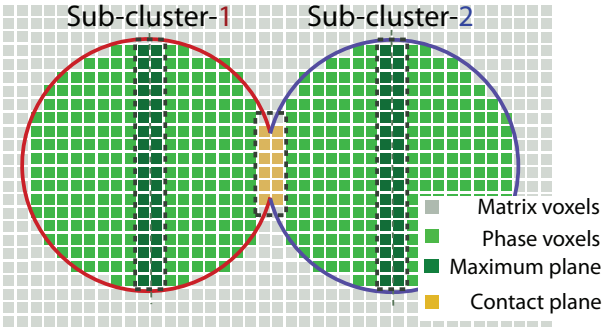


Figure 2.3: An illustration of couple spheres connected by a small neck. Contact volume can be increased by decreasing the distance between sphere centers. The maximum number of contact voxels equals to the voxel number on the maximum plane.

Contact volume ratio is obtained by dividing the number of voxels on the contact plane over the voxel number on the maximum plane. Here we introduce a relative centrality to represent the ratio of the normalized centrality in a network against the maximum normalized centrality in a single sphere with an equal number of vertices. The relative centrality of a graph can be expressed as

$$\widehat{\mathcal{C}}_{\mathcal{B}}(v) = \frac{\mathcal{C}_{\mathcal{B}}^{\mathcal{N}}(v)}{\mathcal{B}(n)} \tag{2.6}$$

where n is the number of phase voxels in the graph network.

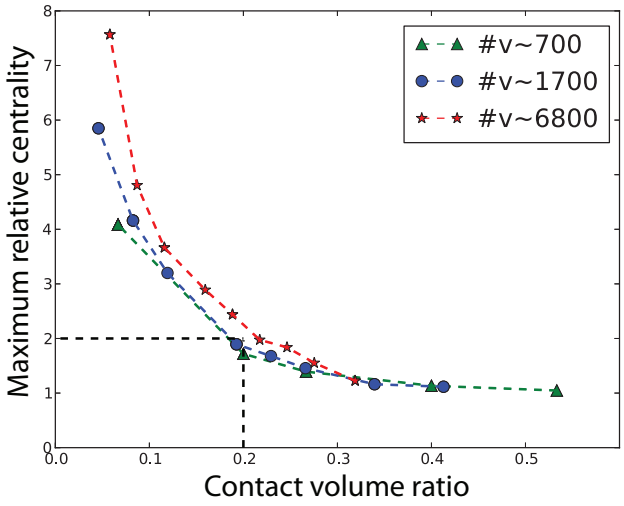


Figure 2.4: The plot of the maximum relative centrality in couple spheres with different contact ratios for three cases with a different number of voxels. Different contact ratios can be acquired by deleting contact voxel pairs. #v is the number of phase voxels in the graph network.

Figure 2.4 shows the influence of neck volume ratio on the maximum relative centrality. The non-dimensional relative centrality is an excellent metric that enables to localize neck voxels automatically. This metric reduces the influence of graph size to the maximum extent and at the same time, the graph topology can be completely reflected in the centrality magnitude. Besides that, the magnitude of relative centrality has a practical implication. The maximum relative centrality is a unit in single spheres and the maximum relative centrality increases exponentially with decreasing contact volume ratio. As shown in Figure 2.4, when contact volume ratio is less than 0.2, the maximum relative centrality is higher than 2, which is large enough for the localization of neck voxels.

2.6 Automatic Neck Volume Removal

Graph segmentation indeed can be separated into two processes: voxel localization and voxel removal. After locating neck voxels, the continued voxel removal is relatively straightforward. In this work, we use *coordination* to represent the number of phase voxels in its six neighbors. The maximum *coordination* is six in 3D. Voxel removal is performed based on the conjunction of high relative centrality and low *coordination*. Combined with *coordination*, only boundary voxels can be removed such that we can avoid introducing cavity inside the cluster. Finally, we loop over all phase voxels and remove voxels with relative centrality higher than 2, moreover, *coordination* less than 6. After each removal of voxels on the outermost layer, neck volume reduces and the volume decrement will in turn increase the maximum relative centrality on neck voxels. Hence, graph segmentation will continue until all neck voxels have been deleted. Detailed segmentation algorithm has been summarized in algorithm (2).

Algorithm 2 Graph segmentation for clusters

Define $\mathbb{C} \rightarrow$ initial clusters

for all $\mathcal{C} \in \mathbb{C}$ **do**

  while True **do**

    Initialize cut voxel number, $n \rightarrow 0$

    for all $v \in \mathcal{C}$ **do**

      if $\widehat{\mathcal{C}}(v) > 2$ && *coordination*$(v) < 6$ **then**

        Remove v from $\mathcal{C}$

        $n = n + 1$

    if $n = 0$ **then**

      Break

    $\mathcal{C} \rightarrow$ Redo cluster initialization

    $\mathcal{C} \rightarrow$ Recalculate relative centrality

Noted that some neck volumes with small contact ratio possibly contain significantly high relative centrality. As a side effect, the prominent high centrality is inclined to raise the centrality of surrounding voxels close to but not belong to the neck volume. Consequently, overshoot might occur in some graph networks and a few voxels could be deleted innocently under constant critical centrality. Certainly, we can define specific critical centrality for each graph network based on its maximum centrality, or be even more cautious, we can only remove the voxel with the highest centrality each time. These strategies can avoid deleting innocent voxels but more deleting times are required, which leads to the corresponding increment in computational effort. After each time of voxel removal, centrality has to be recalculated again and the centrality calculation is the most expensive part in geometry segmentation. In addition, the motivation to perform geometry segmentation is the desire to

counteract the resolution limit of X-ray tomography before implementing voxel data into the extended finite element (XFEM) analysis. Permissible overshoot is preferred in geometrical representation of microstructures since too close particles require very fine XFEM mesh to distinguish the particle boundaries [95]. After trade-off considerations, we set the critical relative centrality to 2, which works well for most practical problems.

2.7 Gap Expansion

As briefly mentioned, spatially closed particles require very fine finite element mesh to represent particle geometries accurately. Indeed, we have developed structured, oct-tree meshes with hanging nodes to represent material geometries [95]. With the help of oct-tree mesh, it is possible to differentiate close particles with acceptable computational effort. However, performing gap expansion on segmented voxels is a better way to avoid unnecessary expense. The basic idea of gap expansion is to delete voxels within certain distance to other clusters in order to broaden the shortest distance between clusters.

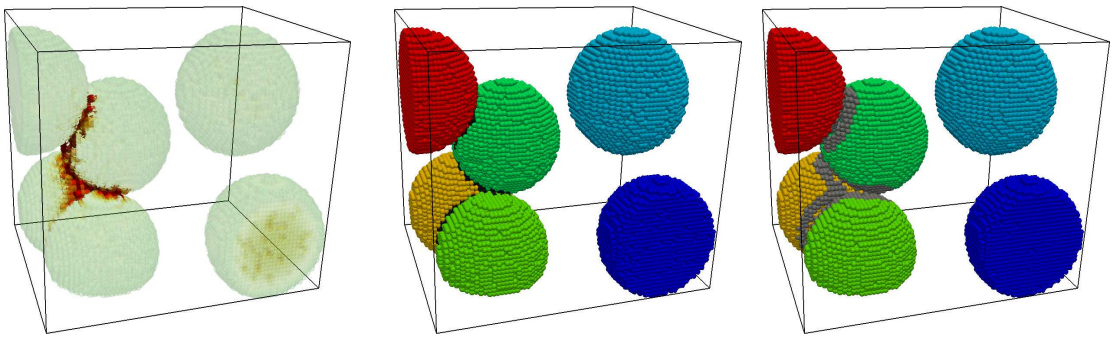
Gap expansion requires localizing voxels within a certain distance to other clusters before deleting voxel. Bin table is implemented to improve the efficiency in voxel localization. Bin table is a uniform grid whose size is several times larger than the corresponding voxel size. Bin table size can be set close to a critical distance whose magnitude should be chosen mainly according to the expected gap dimension, or to be further, the minimum element size in finite element mesh.

The gap expansion algorithm can be summarized as follows: Firstly, construct a bin table overlapped on the voxel domain. Phase voxels are sorted into corresponding bins based on their actual positions. In the loop of each cluster, we find boundary bins possessing any voxel with *coordination* less than six. Then for each boundary

bin, loop over its voxels and calculate the distance with respect to other voxels from the bin as well its neighbor bins. Both voxels are stored in a removal voxel list if their distance is smaller than the critical value, and moreover, the two voxels should belong to different clusters. Finally, it is straightforward to remove voxels from corresponding clusters.

2.8 Example Problem of Multiple Spheres

To conform the geometry segmentation algorithm, geometry segmentation was performed on a numerical example with multiple spheres. The numerical model is constituted by six spheres of the same radius randomly distributed inside a unit cube. The six spheres are initially divided into three clusters with one joined cluster composed of four spheres connected by some narrow volumes. The corresponding voxel discretization is $20\times20\times20$ and there are 1286 phase voxels. After geometry segmentation, the joined cluster is successfully segmented into four at the expense of 87 voxels deleted, which is only about 6.8% of the total number of phase voxels. All removed voxels are around narrow connections as expected. The critical distance in gap expansion is set to four times of voxel size but not absolutely necessary. The critical distance can be changed primarily based on the minimum finite element size expected to use.



(a) Initial centrality (b) Centrality removal (c) Gap expansion removal

Figure 2.5: Geometry segmentation for spheres. (a). Rescaled relative centrality on initial spheres with opacity proportional to centrality magnitude. High centrality is concentrated on narrow connections and the maximum relative centrality is about 12.3 before rescale. (b). Clusters after geometry segmentation with removed voxels highlighted in black. Remaining voxels are colored by corresponding cluster id. The maximum relative centrality is about 1.7 after segmentation. (c). Clusters after gap expansion with removed voxels highlighted in gray. Critical gap distance is four times of the voxel size.

2.9 Example Problem of SiC Particles

To demonstrate the applicability of our algorithm for practical microstructures with complex geometries, we applied the algorithm on X-ray tomography data of SiC particles. The material used was a 2080 aluminum alloy reinforced with 20 vol.% SiC particles (average particle size of 25 μm). The materials were processed by blending SiC and Al powders, compaction of the powder mixture, hot pressing, and hot extrusion (Alcoa Inc., Alcoa, PA). Details of the powder metallurgy process for fabrication of these composite materials can be found elsewhere [101]. X-ray synchrotron tomography was performed at the 2BM beamline of the Advanced Photon Source (APS) at Argonne National Laboratory. Details of the APS beamline 2-BM have

been described elsewhere [13, 102]. Using a standard detector, a 2048×2048 pixel CoolSnap K4 CCD camera coupled with a LuAG:Ce scintillator screen, a specimen pixel size of 1.47 μm was achieved. 2D projections were collected at angular increments of 0.125^o over a range of 180^o. These 2D projections were then reconstructed in 3D using a filtered back-projection algorithm. The tomography was completed in about 15 minutes.

A representative 2D slice of the reconstructed images is shown in Figure 2.6a. It can be clearly seen that the centers of the SiC particles have the similar gray values to that of the Al alloy matrix, however, the boundary between the Al matrix and SiC particles are clearly distinguishable. Due to the similar gray values of the center of the particles and the matrix, conventional segmentation techniques, such as thresholding, could not be used to separate the SiC particles from the matrix. Instead, a semi-automatic segmentation algorithm known as Livewire was used (Mimics, Materialise, Ann Arbor, MI), which is based on the sharp gradient in gray scale values observed along the boundaries of the SiC particles and the Al alloy matrix. The Livewire technique has been explained elsewhere in detail [103]. A selected volume 320 ×650×104 μm^3 was cropped from the 3D reconstructed volume for the segmentation. The representative 2D slice after segmentation is shown in Figure 2.6b. The obtained voxel data was imported into our algorithm to perform geometry segmentation.

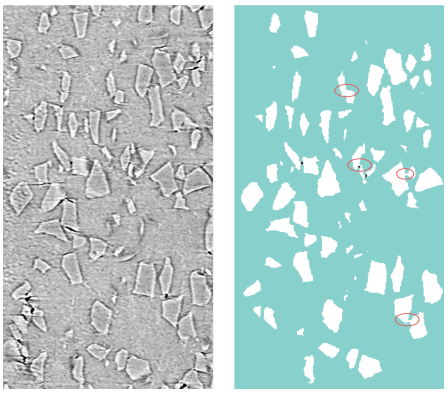
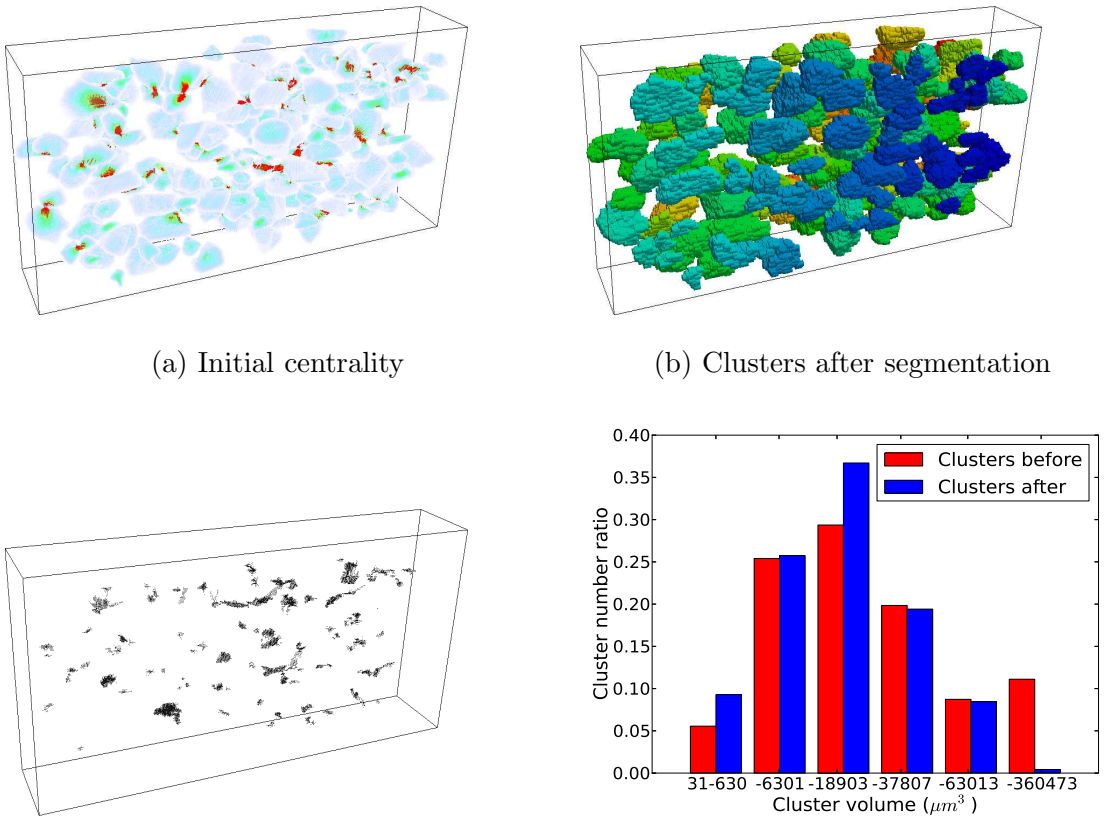


Figure 2.6: (a). Reconstructed 2D tomographic slice of SiC particle reinforced Al alloy matrix composite. (b). SiC phase voxels after image segmentation. The red lines circle some spurious connections.

After image segmentation, we acquired a voxel data with 1,121,582 SiC phase voxels. After applied the initial clustering, phase voxels initially cluster into 126 clusters including about 1% of gigantic clusters composed of more than ten thousands voxels. Three tiny clusters with less than five voxels are abandoned. In parallel computing, since geometry segmentation is performed independently on each individual cluster, the segmentation of one cluster can be completely isolated from others, therefore, the initialized clusters can be evenly distributed into processors in parallel. This feature is very attractive in parallel computing for practical problems with hundreds of clusters. Figure 2.7a shows the centrality distribution on initial phase voxels. After performing geometry segmentation, initial clusters are separated into 237 clusters at the expense of 11577 voxels removed, which accounts for about 1% of the initial phase voxels. Figure 2.7b and 2.7c exhibit segmented clusters and removed voxels, respectively. The maximum cluster contains 24,013 voxels after geometry segmentation, compared with 114,412 voxels initially. Cluster volume distribution is shown in Figure 2.7d.



(a) Initial centrality

(b) Clusters after segmentation

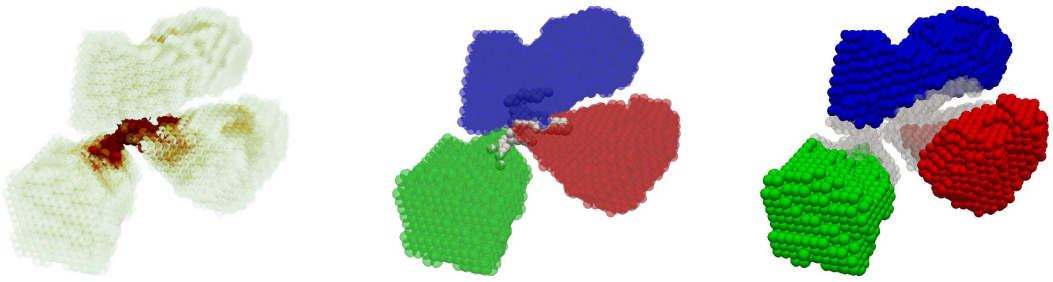
(c) Removed voxels

(d) Cluster distribution

Figure 2.7: Geometry segmentation on SiC particles. (a). Rescaled relative centrality on initial clusters with opacity proportional to centrality magnitude. Initial maximum centrality is about 483. (b). Clusters after geometry segmentation. There are 237 separated clusters (126 clusters, initially). (c). All removed voxels in geometry segmentation. (d). Cluster volume distribution before and after geometry segmentation.

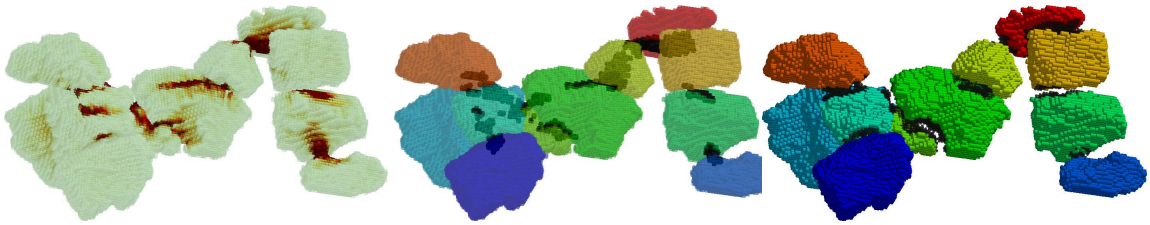
Figure 2.8 and 2.9 exhibit geometry segmentation in more details for two specific initialized clusters, respectively. From these two examples, it is distinct to distinguish the locations of neck volumes, centrality concentration, and removed voxels. As can be seen, neck volumes tend to possess higher relative centrality and all deleted voxels are located around neck volumes. The geometry segmentation algorithm is capable of

segmenting joined clusters with a minimal number of voxels removed, moreover, the geometrical characteristics of original clusters are maintained at the maximum extent. Critical relative centrality is 2 for all examples used in this work. Therefore, the segmentation can be performed automatically for different clusters without tweaking the critical value. Gap expansion is an optional process after geometry segmentation. Voxels within a certain distance to other clusters are deleted to enlarge the distance among clusters. The critical distance was set to four times of the voxel size in both examples.



(a) Cluster-a (b) Centrality cut (c) Gap expansion cut

Figure 2.8: Geometry segmentation on an example cluster. (a). Rescaled relative centrality on one cluster with opacity proportional to centrality magnitude. The initial maximum centrality is about 59. (b). Separated clusters after geometry segmentation with removed voxels highlighted in gray. (c). Separated clusters after gap expansion with removed voxels highlighted in gray.



(a) Cluster-b (b) Centrality cut (c) Gap expansion cut

Figure 2.9: Geometry segmentation on an example cluster. (a). Rescaled relative centrality on one cluster with opacity proportional to centrality magnitude. The initial maximum centrality is about 260. (b). Separated clusters after geometry segmentation with removed voxels highlighted in black. (c). Separated clusters after gap expansion with removed voxels highlighted in black.

Chapter 3

EFFICIENT METHODS FOR IMPLICIT GEOMETRY REPRESENTATION

In this section, the evolution equation for initializing distance fields is developed and transformed into a weak form for solution by Galerkin's method. A second approach for computing distance fields is to initialize the fields with the fast marching method (FMM) on a uniform grid and project the solution to a finite element mesh by least-squares.

3.1 Distance Field Initialization

In this section, the evolution equation for initializing distance fields from segmented voxel data is developed and transformed into a weak form for solution by Galerkin's method.

3.1.1 Level Set Equations

A level set function represents fronts or material interfaces, denoted by C, as the zero level set $C(t) = \{\mathbf{x}|\phi(t,\mathbf{x}) = 0\}$ of a continuous level set function $\phi(\mathbf{x},t)$. The evolution equation for a level set function $\phi(\mathbf{x},t)$ can be written in the general form

$$\frac{\partial\phi}{\partial t} + F|\nabla\phi| = 0, \tag{3.1}$$

where F is a velocity field function that defines the motion of the interface along its normal direction. In this work, the velocity field is chosen so that the evolution equation reaches a stationary point when the level set function is a signed distance function. A signed distance function is a special case of a level set function where the value of the field gives the nearest distance to an interface and the sign of the

field determines whether or not a coordinate is interior or exterior to the interface. By definition, the magnitude of the gradient of a signed distance function is unity, so the velocity field is defined to enforce this condition.

$$F = \text{sign}(\phi)\left(1 - \frac{1}{|\nabla\phi|}\right) \tag{3.2}$$

The second criterion for the evolution equation is to ensure that the distance field is negative at any voxel location within the microphase and positive exterior to the microphase. To enforce this criterion, a penalty term is introduced that decreases the level set function at any voxel that is within the microphase if the level set field is positive. To reduce spurious oscillations in the distance field caused by shocks that arise as a result of the hyperbolic nature of the evolution equation, a shock capturing viscosity is added to penalize the curvature of the distance field. Lastly, a fill ratio corrective term is added that uniformly shrinks the microphase domains if the total volume fill ratio, γ, exceeds the fill ratio computed from the voxel data, γ_V. The resulting evolution equation is

$$\begin{aligned}\frac{\partial\phi}{\partial t} + \underbrace{\text{sign}(\phi)\left(|\nabla\phi| - 1\right)}_{\text{Distance}} + \underbrace{\alpha\sum_{V}\delta(\mathbf{x}-\mathbf{x}_v)H(\phi)h_e^3}_{\text{Voxel penalty}} \\ = \underbrace{\nu\nabla^2\phi}_{\text{Viscosity}} + \underbrace{\beta(\gamma-\gamma_V)H(\gamma-\gamma_V)}_{\text{Fill ratio corrector}}\end{aligned} \tag{3.3}$$

where ν is the viscosity, $H(\cdot)$ and $\delta(\cdot)$ are the Heaviside and Dirac delta functions respectively, α and β are user-specified weighting parameters, and h_e is the element size. The voxel penalty term is a sum over the set V of segmented microphase voxels, each located at $\mathbf{x}_v$. The solution of (3.3) will develop shocks at points equidistant from two interface points, and, therefore, the numerical solution requires regularization by a viscosity-like term. Hansbo proposed a non-linear artificial viscosity [104] which was later applied by Chessa in a stabilized Galerkin least-squares method to solve

a solidification problem [105]. For meshes with large variation in element size such as the adaptively refined meshes used in this work, a scalar viscosity field defined as $\nu = \eta \min(h_e^2)$ is effective for damping out shocks. Here, η is a user-specified viscosity factor, and $\min(h_e)$ is the minimum element size.

3.1.2 Weak Form and Discretization

In order to solve the evolution equation on a finite element mesh, equation (3.3) is transformed into a weak form by multiplying by a test function ω and integrating over the domain.

$$\int_\Omega \omega \left[\frac{\partial \phi}{\partial t} + \text{sign}(\phi)(|\nabla \phi| - 1) \right] d\Omega + \alpha \int_\Omega \omega \sum_V \delta(\mathbf{x} - \mathbf{x}_v) H(\phi) h_e^3 \, d\Omega$$
$$= \nu \int_\Omega \omega \nabla^2 \phi \, d\Omega + \beta \int_\Omega \omega(\gamma - \gamma_V) H(\gamma - \gamma_V) \, d\Omega \tag{3.4}$$

After integration by parts and rearranging terms, equation (3.4) becomes

$$\int_\Omega \omega \frac{\partial \phi}{\partial t} d\Omega = - \int_\Omega \omega \, \text{sign}(\phi)(|\nabla \phi| - 1) d\Omega - \alpha \sum_V \omega H(\phi) h_e^3$$
$$+ \nu \int_\Gamma \omega \nabla \phi \cdot \mathbf{n} \, d\Gamma - \nu \int_\Omega \nabla \omega \nabla \phi \, d\Omega + \beta \int_\Omega \omega(\gamma - \gamma_V) H(\gamma - \gamma_V) \, d\Omega \tag{3.5}$$

where ω and ϕ are C^0 continuous test and trial functions, respectively. The Neumann boundary conditions must be integrated over the boundary Γ with normal $\mathbf{n}$.

The resulting semi-discretized equations for the level set update are

$$\mathbf{M}_\phi \dot{\phi} = \mathbf{f}_{BC} + \mathbf{f}_f - \mathbf{f}_\phi - \mathbf{f}_{SC} - \mathbf{f}_v \tag{3.6}$$

where

$$\mathbf{M}_\phi = \int_\Omega \mathbf{N}^T \mathbf{N}\, d\Omega \tag{3.7}$$

$$\mathbf{f}_\phi = \int_\Omega \mathbf{N}^T \text{sign}(\phi)\, (\,|\nabla\phi| - 1)\; d\Omega \tag{3.8}$$

$$\mathbf{f}_{SC} = \int_\Omega \nu \nabla \mathbf{N}^T \cdot \nabla\phi\, d\Omega \tag{3.9}$$

$$\mathbf{f}_{BC} = \int_\Gamma \nu \mathbf{N}^T \nabla\phi\, \mathbf{n}\, d\Gamma \tag{3.10}$$

$$\mathbf{f}_v = \alpha \sum_V \mathbf{N}^T H(\phi) h_e^3 \tag{3.11}$$

$$\mathbf{f}_f = \beta \int_\Omega \mathbf{N}^T (\gamma - \gamma_V) H(\gamma - \gamma_V)\, d\Omega, \tag{3.12}$$

and $\mathbf{N}$ are the nodal shape functions. The global matrices and vectors are denoted analogously to their equivalents for explicit finite element simulation of elastodynamics, i.e. $\mathbf{M}_\phi$ is analogous to a mass matrix, and is similarly lumped by row summation into a diagonal matrix to avoid solving a large linear system. The force terms of the semi-discrete equations, are defined in (3.8-3.12). The derivatives in time are then discretized by a forward-Euler integration

$$\mathbf{\Phi}^{n+1} = \mathbf{\Phi}^n - \frac{\Delta t}{\mathbf{M}_\phi}(\mathbf{f}_\phi^n + \mathbf{f}_{SC}^n - \mathbf{f}_{BC}^n + \mathbf{f}_v^n - \mathbf{f}_f^n) \tag{3.13}$$

where the superscript indicates the time step, i.e. $\mathbf{\Phi}^n = \mathbf{\Phi}(n\Delta t)$.

3.1.3 Critical Time Step

While the evolution equation (3.3) can be characterized as a hyperbolic partial differential equation, the viscous term adds a parabolic component to the system. The critical time step, computed from the CFL condition, for the hyperbolic component of equation (3.3) is

$$\Delta t^h = \min_{\mathbf{e}} \frac{h(e)}{F(e)} \tag{3.14}$$

where $h(e)$ is the element size for element e and $F(e)$ is the corresponding velocity.

As for the parabolic part, some simplifications can be made [106] based on the assumption that element size h is sufficiently small leading to

$$\Delta t^p = \min_{\mathbf{e}} \frac{h^2(e)}{2\nu} = \frac{1}{2\eta} \qquad \text{for } \nu(\phi, t) > 0 \tag{3.15}$$

The combination of the critical time steps of the hyperbolic and parabolic parts of the evolution equation lead to a transition between a constant critical time step for large elements, and a linear relationship between the element size and velocity for small elements.

3.2 Fast Marching Method

In the FMM, upwind finite differences are taken discretizing (1.12) to solve the hyperbolic partial differential equation without regularization. For initializing a distance field, the upwind direction is always oriented toward the nearest interface, or toward the direction of smaller absolute value of the field. A key aspect of the method is the order in which the unknown grid points are computed to maintain the validity in the upwind differences. It is straightforward to develop the discretization of (1.12) for three dimensions. Suppose for the computation of point $\mathbf{x}_{ijk}$, the correct upwind directions are $(-x, -y, -z)$. The first-order discretization of (1.12) reduces to the quadratic equation

$$\begin{aligned}(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2})\phi_{ijk}^2 - 2(\frac{\phi_{i-1jk}}{\Delta x^2} + \frac{\phi_{ij-1k}}{\Delta y^2} + \frac{\phi_{ijk-1}}{\Delta z^2})\phi_{ijk} \\ + \frac{\phi_{i-1jk}^2}{\Delta x^2} + \frac{\phi_{ij-1k}^2}{\Delta y^2} + \frac{\phi_{ijk-1}^2}{\Delta z^2} - 1 = 0\end{aligned} \tag{3.16}$$

where Δx, Δy and Δz are the grid discretization sizes in each dimension.

In the FMM, a set of grid points must first be computed directly in order for subsequent points to have valid upwind differences. These initial points are found

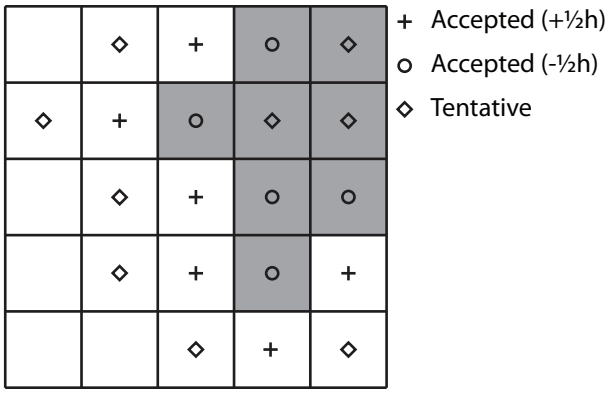


Figure 3.1: Initialization of boundary points in the FMM. Grid points adjacent to the voxel boundary are initialized and accepted. All neighboring points to these newly accepted points have tentative values computed.

by iterating over the set of voxels belonging to the microphase and searching for neighboring voxels that do not belong to the phase. The interface is assumed to pass halfway between these points so that each can be initialized as one-half the voxel spacing distance. The sign of these newly initialized points is chosen to reflect whether or not they belong to a microphase's voxels. The initialization process is shown in Figure 3.1. The newly initialized values are then placed in a set of accepted grid points and all neighboring points that are not yet accepted are placed in a tentative set and their values are computed by the discretization of (1.12).

After directly initializing the band of grid points around the interface, the remaining grid points are computed in the order of smallest absolute value. In order for the method to scale efficiently, the tentative set of grid points in this method is stored in a binary tree. The FMM is summarized in Algorithms 3 and 4, where h represents the appropriate grid size, Δx, Δy or Δz. Generally, the grid size should be equal to the size of the discretized voxel set, however, to reduce the computational cost of the distance field initialization, a coarser grid could be applied.

Algorithm 3 Initialize level set field by FMM

Define min(**T**) $\rightarrow$ the voxel with the minimum tentative value

Define V $\rightarrow$ filled voxel set

for all $v \in$ **V do**

  for all $\mu \in$ neighbors of v such that $\mu \notin$ **V do**

    $\phi(v) \leftarrow -h/2$

    $\phi(\mu) \leftarrow +h/2$

    Insert v and μ into **A**

for all $v \in$ **A do**

  for all $n \in$ neighbors of v **do**

    $\phi(n) \leftarrow$ TENTATIVE_VALUE(n)

    T $\leftarrow$ **T** $\cup\, n$

while T $\neq \emptyset$ **do**

  $v \leftarrow \min(\mathbf{T})$

  Move v from **T** to **A**

  for all $\mu \in$ neighbors of v such that $\mu \notin$ **A do**

    $\phi(n) \leftarrow$ TENTATIVE_VALUE(μ)

    Move v from **T** to **A**

To reduce the steep gradients caused by the stepwise nature of the segmented voxel data, the distance function is smoothed by a 3×3×3 weighted kernel. After the distance field is computed and smoothed on the uniform grid, the finite element mesh is refined by interpolating the distance function at the center of each element from the uniform grid and refining the element if the absolute value of the distance function is below a threshold, i.e. elements near the material interfaces are refined. The threshold distance can be set to limit refinement to a single band of elements or

Algorithm 4 Compute tentative value at grid point

function TENTATIVE_VALUE(v)

Set up discretization equation: $a\phi^2 + b\phi + c = 0$ by equation (3.16)

$\Delta \leftarrow b^2 - 4ac$

if $\Delta \geq 0$ **then**

return $\frac{\max(|-b-\sqrt{\Delta}|, |-b+\sqrt{\Delta}|)}{2a}$

else

$\tau \leftarrow$ empty array

for all $\mu \in$ neighbors of v such that $\mu \in \mathbf{A}$ **do**

$\tau \leftarrow$insert$\{\phi(\mu) + \text{sign}(\phi(\mu))h\}$

return $\min(\tau)$

to a region of specified thickness centered at the interfaces. The nodal distance values are then computed on the refined mesh by least-squares projection. The projection operator is

$$\phi_I = \left(\sum_{ijk} N_I(\mathbf{x}_{ijk}) N_J(\mathbf{x}_{ijk}) \right)^{-1} \sum_{ijk} N_J(\mathbf{x}_{ijk}) \phi_{ijk}, \tag{3.17}$$

where ϕ_I are the nodal values of the distance function and N are the nodal shape functions evaluated at each voxel grid point. The summations are over all points in the uniform grid.

3.3 Mesh Refinement with Hanging Node

Approximation of the displacement and stress fields at equilibrium in complex microstructured materials can pose significant computational challenges when the microstructure of the material is geometrically complex. It is often the case that the optimal discretization is nonuniform throughout the domain. For example, the density of finite elements required to resolve stress concentrations near irregularly-

shaped inclusions may be significantly higher than in a bulk matrix material. In this case, a uniformly structured mesh is far from optimal. Instead, we construct the implicit representation of the material geometry on structured, oct-tree meshes with hanging nodes [107].

In this approach, the starting point is a coarse, uniform structured mesh, which can only roughly approximate the distance field. For level set evolution, a stationary point is found for the distance field at this coarse scale and the error in each element is estimated by the number of voxel points within the element that have an incorrectly signed distance. Elements are selected and refined by either a threshold percentile or by a maximum allowable element error. These elements are refined and a new stationary point is found on the refined mesh. This procedure is then repeated either until the error is minimized to a prescribed tolerance or until a maximum number of mesh elements is reached.

For the FMM construction of the signed distance function, elements are selected for refinement based on their distance to the zero level set of the distance functions. The initialization adaptivity is driven by the nearest distance to material interfaces.

To simplify the enforcement of compatibility and continuity of the distance fields, the refined meshes are restricted to 1-irregular which only allow for hanging nodes to be located at the center of an element face or edge. After each element is subdivided into eight child elements, the neighboring elements of the original parent element are checked to ensure that the 1-irregular property is maintained. If necessary, neighboring elements are also refined in order to maintain this restriction.

To enforce compatibility and continuity of the distance fields, two operations are defined in order to restrict the value at a hanging node to be the average of its neighbors. Hanging nodes at a face are restricted to the average of the four corner nodes on the face. Hanging nodes on an element edge are the average of the two

nodes on the edge. The first operation is a scatter operation used to distribute nodal values accumulated by integration of element domains, i.e. this operation is used to scatter the terms that contribute to the time derivative of the distance function.

$$\phi_i \leftarrow \phi_i + \frac{1}{n}\,\phi_h \quad \text{for } i \in \mathcal{H}, \tag{3.18}$$

where $\mathcal{H}$ is the set, of length n, of shared nodes that are coupled to the hanging node, h.

The other operation interpolates the value of the hanging nodes from their surrounding shared nodes.

$$\phi_h = \frac{1}{n} \sum_{i \in \mathcal{H}} \phi_i \tag{3.19}$$

3.4 Verification Problems

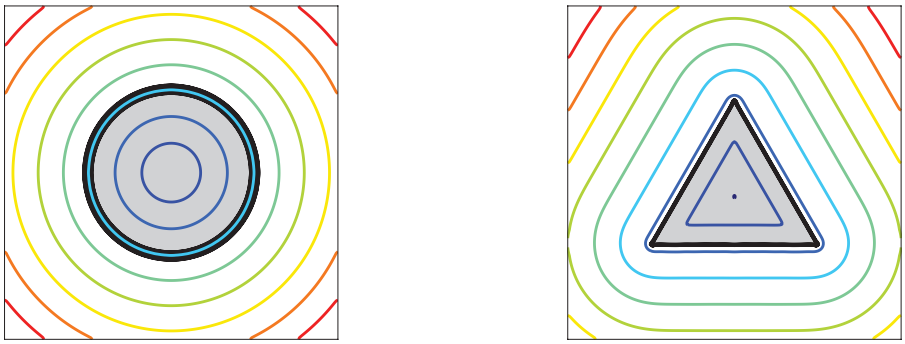
To verify the distance field evolution equation developed in Section 2, numerical experiments were conducted for simple two- and three-dimensional microstructures. Linear elements, four-node quadrilaterals in 2D and eight-node hexahedral elements in 3D were used. The error in the distance function was assessed by two metrics. The first is the ratio of voxels whose interpolated value has the incorrect sign with respect to the segmented data set. This metric is a function only of the distance field and converges quadratically with respect to element size for the linear elements used. The second metric measures the deviation of the solution from a distance field by computing the ℓ^2 error norm of the distance field

$$e_{\ell^2} = \sqrt{\frac{\int_\Omega \left(\|\nabla\phi\| - 1\right)^2 d\Omega}{\int_\Omega d\Omega}} \tag{3.20}$$

This distance error is a function of the gradient of the solution, and analogous to the energy error in elastic problems, the rate of convergence of the distance error is linear with respect to the element size.

3.4.1 *Circular Inclusion*

Figure 3.2a shows the initialized level set contours for a circular inclusion. The simulation domain is a bi-unit square with a center circular inclusion. The normalized distance error (3.20), is less than 2% at the stationary value with mesh dimension 50×50.



(a) Circular inclusion

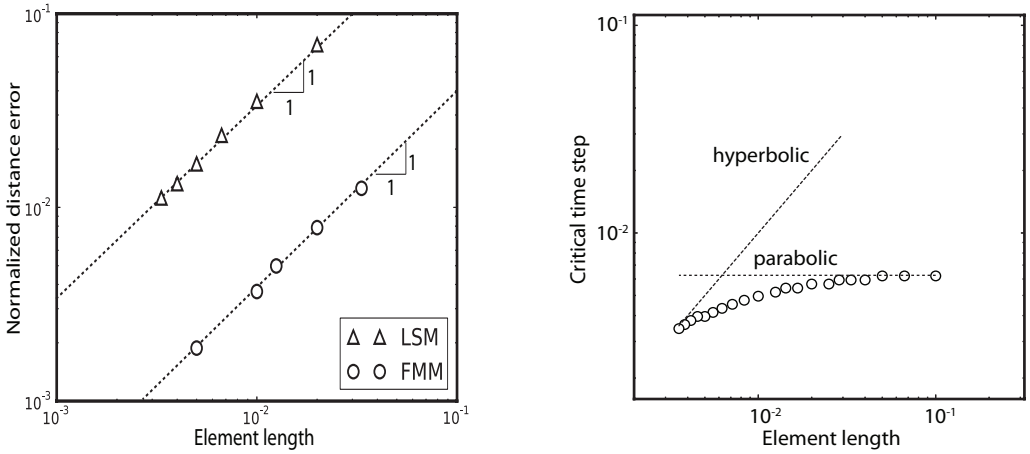
(b) Triangular inclusion

Figure 3.2: Level set contours for two-dimensional circular and triangular inclusions.

3.4.2 *Triangular Inclusion*

The evolution equation was tested with a triangular inclusion to determine whether the shock capturing operator successfully eliminates the oscillations caused by sharp corners. In choosing the viscosity parameter, a trade-off is made between reducing oscillations and avoiding excessive rounding of the corners in the data. Figure 3.2b shows the stationary contours for a center triangle model containing sharp corners. The mesh dimension used here is 60×60 and the viscosity factor η, voxel penalty factor α and fill force factor β were set to 5, 0.01, and 0.001, respectively. Figure 3.3a shows the convergence of the distance error for both the level set and the FMM.

Both methods have optimal convergence rates, however the FMM is superior in accuracy. Figure 3.3b shows the numerically determined critical time closely follows the analytical bounds determined earlier for the triangular inclusion model.



(a) Convergence (b) Critical time step

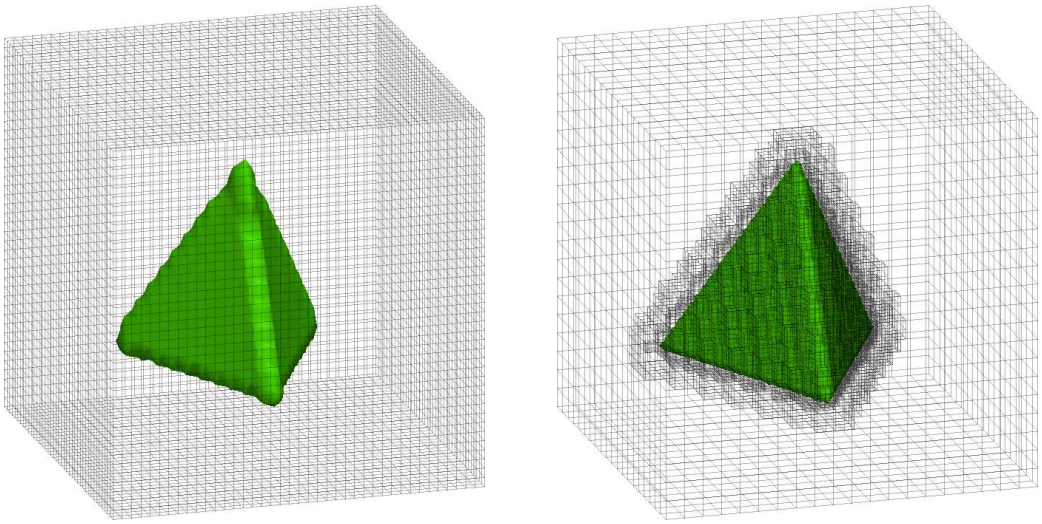
Figure 3.3: Observed convergence and critical time step for the triangular inclusion model.

3.4.3 Three-dimensional Tetrahedral Inclusion

The evolution equation was tested in three dimensions with a tetrahedral inclusion. The sharp edges of the tetrahedron present a challenging case due to the high curvature in the distance field. A dataset of segmented voxels was created within a cube of unit length such that the voxel resolution was $300 \times 300 \times 300$ and the final distance field was computed for both uniform and refined meshes. The uniform mesh contained 64,000 elements, and the refined oct-tree mesh was initialized with 8,000 elements and refined in two iterations, resulting in a total of 25,175 elements.

The viscosity factor η, voxel penalty factor α, and fill force factor β were set to 30, 0.03, and 0.001, respectively for both meshes. Figure 3.4 shows that oct-tree mesh can represent sharp corners better than the uniform mesh despite having less than

half the total number of elements. The voxel errors were 0.29% and 0.24% for the uniform and refined tree meshes, respectively.



(a) Uniform mesh (b) Refined tree mesh

Figure 3.4: Final zero level set of the distance field on uniform and refined meshes.

3.5 Distance Field Initialization for a Wrought Al7075 Alloy Model

To demonstrate the applicability of implicit geometrical representation for complex material microstructures, we applied both methods on X-ray tomography data of an aluminum alloy. The material used in this study is a commercially available, wrought 7075-T651 aluminum alloy. The specimen was machined by electro-discharge machining for microstructural characterization with X-ray synchrotron tomography. X-ray tomography was performed at the 2-BM beamline at the Advanced Photon Source (APS) in Argonne National Laboratory. The tomography system at APS has been described in detail elsewhere [108]. The beam was focused on the specimen and a $CdWO_4$ scintillator screen was used to convert the transmitted X-rays to visible light. This was coupled with an objective lens and a CoolSnap K4 CCD camera to achieve pixel sizes of 1.8 μm. 2D projections were collected at an angular

increment of 0.12°. The 3D reconstruction of 2D projections was performed using a filtered-back-projection algorithm.

The domain dimensions of the alloy model are 369×703.8×864 μm and the corresponding voxel discretization is 205×391×480. A fine uniform mesh can pose significant computational inefficiency due to the wide range of inclusion sizes. The largest cluster in this dataset contains over 9000 voxels while the smallest are composed of several voxels. Figure 3.5 shows the overall distribution of the cluster volume for both Fe-rich and Si-rich phases.

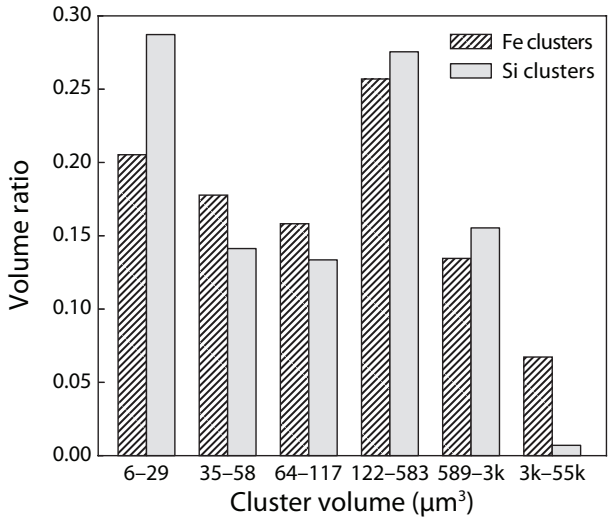


Figure 3.5: Distribution of filled cluster volume.

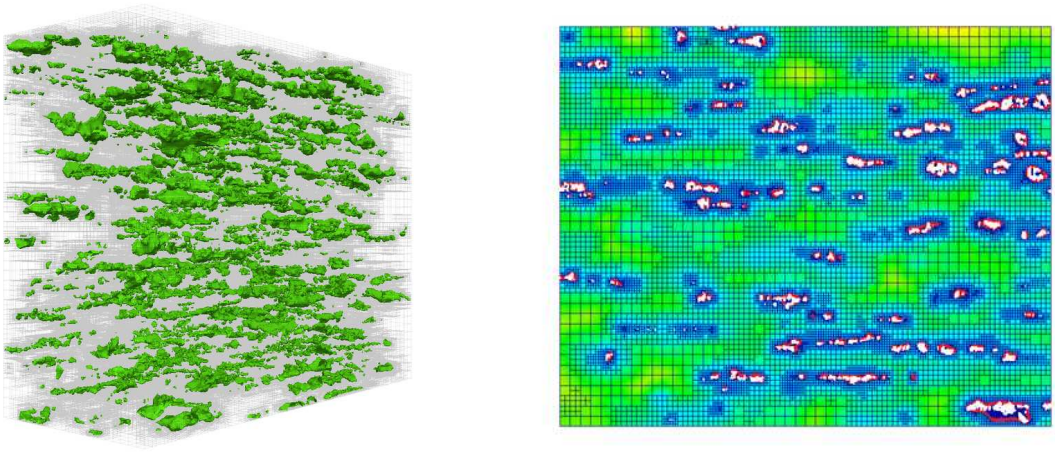
3.5.1 Evolution Equation Initialization

In the solution of the distance field evolution equation, the optimization of viscosity and penalty parameters was performed on a small subset of the model. Due to the wide range of inclusion sizes, optimized parameters were updated after each mesh refinement to capture the smaller inclusions more precisely. The viscosity parameter η was initially chosen as 5.0 and was increased by a factor of two for each successive refinement. The optimized voxel penalty factor α varied between 2.0 and 10.0 for

the Si-rich and Fe-rich phases. The weighting of the fill ratio corrector was kept at a constant value of 50.0.

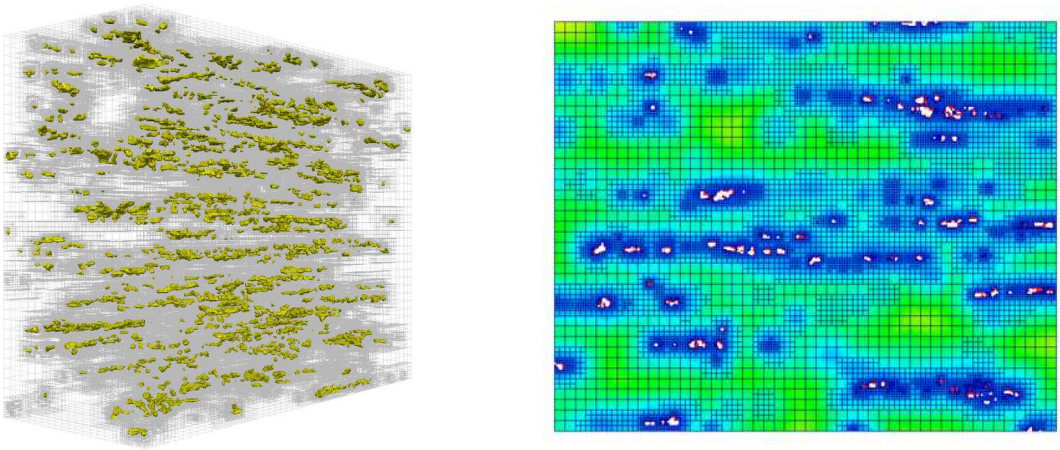
An initial uniform mesh of 20×39×48 elements was created for the distance field evolution and subsequently refined based on the previously described refinement algorithms. Figure 3.6 shows the initialized Fe-rich inclusion geometries on the refined mesh wireframe. After three iterations of mesh refinement, the mesh contained nearly 1.6 million elements. A uniform mesh with an element size equal to the finest elements would contain nearly 19 million elements. The resulting voxel error reached was 1.71% with a fill ratio of 1.89%, compared to the 1.91% ratio of segmented voxels. On a 16-core server, the level set initialization of the Fe-rich phase took 18 hours in total.

Figure 3.7 shows the final Si-rich inclusion geometries on the refined mesh. The number of elements is about 0.7 million after three times of mesh refinement. Finally the voxel error is about 0.52% and the fill ratio is about 0.41% (0.41% actual). On a 16-core server, the time required to initialize the distance field was 9 hours.



(a) Full domain. (b) 2D slice of the domain.

Figure 3.6: Implicit geometry on refined mesh wireframe showing Fe-rich inclusions. In the 2D slice view, segmented voxels for each inclusion are drawn in white and the zero level sets are highlighted with thick lines.



(a) Full domain. (b) 2D slice of the domain.

Figure 3.7: Implicit geometry on refined mesh wireframe showing Si-rich inclusions. In the 2D slice view, segmented voxels for each inclusion are drawn in white and the zero level sets are highlighted with thick lines.

3.5.2 Fast Marching Method Initialization

The FMM was applied for the same aluminum alloy dataset. The distance fields were initialized on a fine uniform grid with spacing equal to the voxel dataset. Trilinear smoothing was performed as previously described to smooth the high gradients resulting from the discretized voxel geometry. An initially uniform, $15\times30\times36$ mesh was refined in three iterations such that elements bisected by interfaces were chosen for refinement, resulting in a mesh containing 1.8 million elements. The smoothed distance fields for each phase were then projected by least-squares to the refined finite element mesh.

Figure 3.8 shows the zero contours of the final distance fields for both phases. The resulted voxel errors are about 2.8% and 0.49% for Fe-rich and Si-rich inclusions respectively. The FMM was not implemented in parallel however the projection to the finite element mesh was parallelized. On a 16-core server, the time required to initialize the distance field for both phases and project the solution to the FEM mesh was five minutes.

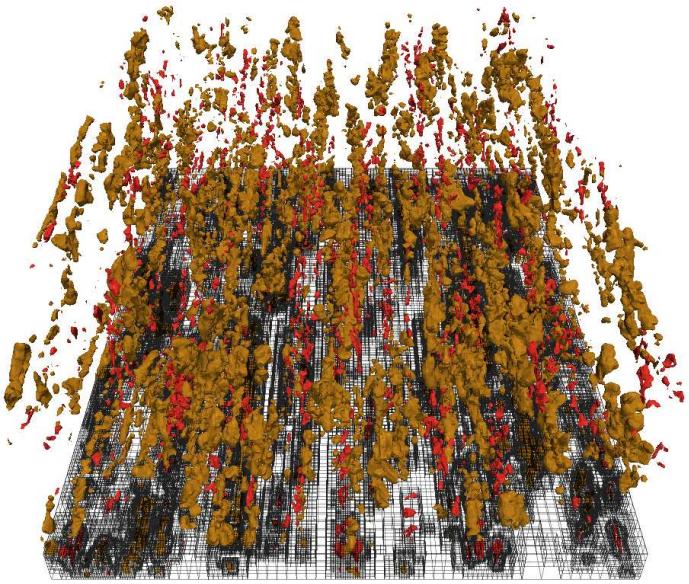


Figure 3.8: Implicit geometry showing Fe-rich inclusions (yellow) and Si-rich inclusions (red). For clarity, the mesh element edges are shown only within a small layer at the bottom of the domain.

Chapter 4

FRACTURE ANALYSIS OF METAL MATRIX COMPOSITES

MMCs exhibit three basic mechanisms for the initiation and growth of damage: Cleavage fracture of the brittle particles, debonding at the interface between matrix and reinforced particles, and after that, cavities and voids nucleate and grow in the matrix that leads to the ductile failure of the matrix ligaments between particles [87, 109]. In this work, only the first stage of fracture is considered. Brittle particles are considered as isotropic before cleavage fracture. Matrix deforms plastically according to a nonlinear J_2 plastic law fitted from a stress-strain curve of 2080-T6 aluminum [110]. Reinforced particles and matrix are perfectly bonded all the time during fracture. Failure strengths of brittle materials vary unpredictably in a wide range from specimen to specimen even though samples are manufactured in the same way and tested under the same condition [89, 88]. Therefore, fracture statistics has to be applied to the failure strengths of brittle reinforced particles.

4.1 Level Set Geometry Representation

In this work, level sets are employed to represent microstructures implicitly. Level set methods provide a concise way to describe complex microstructures with consistent mesh quality and level set fields can be conveniently applied in enrichments. Level sets have been introduced for material interface modeling [111, 55] and crack modeling [112, 113]. The phase interfaces are represented by the zero level set of a continuous level set function

$$f_\alpha^{\text{int}}(\mathbf{x}) = 0, \quad \alpha = 1 \dots n^{\text{int}} \tag{4.1}$$

where n^{int} is the number of reinforcing phases. For each phase, one level set can represent all interfaces of that phase. If point $\mathbf{x}$ is inside phase α, we set $f_\alpha^{\text{int}}(\mathbf{x})$ to be negative, otherwise, $f_\alpha^{\text{int}}(\mathbf{x})$ is set to be positive. We have presented two methods in previous work [95] for level set initialization of complex material interfaces. In the first method, a level set evolution equation is formulated and solved by the Galerkin method. In the second approach, we initialize the distance field by the fast marching method [58, 59] on a uniform grid, and then project the solution onto the finite element mesh by least squares. The second approach is superior in speed and accuracy, which is applied to initialize the matrix/reinforcement interfaces.

To represent crack surface on a fractured particle, a similar level set function is introduced for each particle.

$$f_\beta^{\text{cr}}(\mathbf{x}) = 0, \quad \beta = 1 \dots n^{\text{par}} \tag{4.2}$$

where β is a particle id, n^{par} is the total number of particles, and the zero level set gives the crack surface of the particle. The above level set field defines a surface gliding through the whole simulation model. In order to restrict crack surfaces within the particle, one more level set function is used to define the location of the crack tip.

$$g_\beta^{\text{cr}}(\mathbf{x}) < 0, \qquad \text{if } \mathbf{x} \text{ is inside of particle } \beta \tag{4.3}$$

$$g_\beta^{\text{cr}}(\mathbf{x}) > 0, \qquad \text{if } \mathbf{x} \text{ is outside of particle } \beta \tag{4.4}$$

$$g_\beta^{\text{cr}}(\mathbf{x}) = 0, \qquad \text{if } \mathbf{x} \text{ is on the interface of particle } \beta \tag{4.5}$$

Finally, the crack surface on particle β can be represented by the combination of two level set fields, $f_\beta^{\text{cr}}(\mathbf{x}) = 0$ and at the same time, $g_\beta^{\text{cr}}(\mathbf{x}) \leq 0$.

4.2 Formulation of XFEM for Discontinuities

In traditional finite element models, discontinuities must reside along element faces such that the discontinuities can be represented explicitly in finite element models.

In the XFEM, interior discontinuities are represented implicitly by level set fields and the displacement field approximates by a discontinuous displacement enrichment [96] based on a local partition of unity [97]. Given a finite element model $\Omega \in \mathbb{R}^3$, partitioned into finite elements, let $\mathcal{S}$ be the set of all finite element nodes, $\mathcal{S}^{\text{cr}}$ be the set of nodes of elements whose edges are intersected by a crack surface Γ^{cr}, and $\mathcal{S}^{\text{int}}$ be the set of nodes of elements intersected by a material interface Γ^{int}. The XFEM displacement field can be expressed by

$$\mathbf{u}^h(\mathbf{x},t) = \sum_{I\in\mathcal{S}} N_I(\mathbf{x})\mathbf{u}_I + \sum_{J\in\mathcal{S}^{\text{cr}}} \phi_J(\mathbf{x})\mathbf{q}_J + \sum_{K\in\mathcal{S}^{\text{int}}} N_K(\mathbf{x})\psi(\mathbf{x})\mathbf{q}_K \tag{4.6}$$

where $\mathbf{u}_I$ are the nodal displacements and $N_I(\mathbf{x})$ are finite element shape functions. Additional enriched degrees of freedom $\mathbf{q}_J$ and $\mathbf{q}_K$ are for crack and material interfaces, respectively. The function $\phi_J(\mathbf{x})$ represents jump enrichment which introduces a discontinuity in the displacement at the crack surface. For a strongly-bonded material interface, the displacement field remains continuous, but strain can be discontinuous on the interface. The function $\psi(\mathbf{x})$ represents kink enrichment which introduces a discontinuity in the gradient of displacement at a material interface. The jump enrichment function for a crack is [114]

$$\phi_J(\mathbf{x}) = N_J(\mathbf{x})\big[H\big(f^{\text{cr}}(\mathbf{x})\big) - H\big(f^{\text{cr}}(\mathbf{x}_J)\big)\big]H\big(-g^{\text{cr}}_\beta(\mathbf{x})\big) \tag{4.7}$$

where $H(\cdot)$ is the Heaviside step function given by

$$H(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

The simplest kink enrichment function is an absolute value function [115, 111, 55]. Modeling interface with the absolute value function is troublesome because it does not vanish at the edges of the elements intersected by the interfaces. Moës proposed

a modified kink enrichment function [116] to preserve the ridge at interfaces, but also vanish at the edges of enriched elements. The enrichment function is given as

$$\psi(\mathbf{x}) = \sum_I \left|f^{\text{int}}(\mathbf{x})\right| N_I(\mathbf{x}) - \left|\sum_I f^{\text{int}}(\mathbf{x}) N_I(\mathbf{x})\right| \tag{4.9}$$

This enrichment function eliminates the need of blending elements such that only elements intersected by interfaces are enriched.

4.3 Implementation of Weibull Distribution Model

Due to the high brittleness of the embedded particle, once fracture happens, the crack will immediately expand through the whole particle, leading to the total splitting. While the particle and matrix are always perfectly bonded, the cracks are assumed to be arrested at the particle-matrix interface. The Weibull distribution model [117] has been widely used to predict the fracture probability of brittle particles in ductile matrix [118, 119]. The Weibull distribution function gives a simple but appropriate mathematical expression that can automatically account for the size effect on failure in particles. The Weibull model is based on the weakest link statistics in which the interaction between flaws can be neglected. A particle can be analogous to a chain consisting of several links and each link is analogous to a flaw in the particle. The chain fails as soon as a single link fails such that the failure probability of the chain is primarily dominated by the weakest link. If we want to calculate the probability of failure P_n of a chain consisting of n links, and assume the links are identical with the probability of failure P, the probability of not failure of the chain, $1 - P_n$, is equal to the probability of the simultaneous not failure of all the links, $1 - P_n = (1 - P)^n$. The central idea of Weibull distribution is to define a distribution function in an exponential form [117]

$$P(X \leq x) = 1 - e^{-f(x)} \tag{4.10}$$

where $f(x)$ can be any positive, nondecreasing function which should vanish at a value. The exponential form of the distribution function has intrinsic merits to account for the size effect since the cumulative failure probability can be easily denoted as

$$P_n(X \leq x) = 1 - e^{-nf(x)} \tag{4.11}$$

The cumulative Weibull probability can automatically account for the particle size effect. The critical flaw size will simply decrease and the particle strength becomes larger as the particle volume shrinks [120]. Therefore, the probability of fracture is larger when the particle size increases or the stress acting on the particle becomes larger [121]. When the critical flaw size is smaller than most of particle sizes, a Weibull distribution is suitable to describe the strength distribution of particles. The cumulative Weibull fracture probabilities were evaluated for each particle i at each load increment by using the expression [87, 120]

$$P^i = 1 - \exp\Bigg(- \frac{1}{V_0} \int_{\Omega_{\mathbb{P}_i}} \left(\frac{\sigma_1(\mathbf{x})}{\sigma_f} \right)^m d\Omega_{\mathbb{P}_i} \Bigg) \tag{4.12}$$

where V_0 is a reference volume, σ_1 is the maximum principal stress at $\mathbf{x}$, σ_f and m are the characteristic strength and Weibull modulus, respectively. The Weibull modulus is a measure of the degree of strength dispersion. Large Weibull modulus will narrow down the probability distribution [89]. To create a fracture plane in particle $\mathbb{P}_i$, an on-plane point at coordinate $\mathbf{x}_f$ and the plane normal direction $\mathbf{n}_f$ are needed. The cleavage point and the normal direction of the fracture surface can be denoted as

$$\begin{aligned} \mathbf{x}_f &= \frac{\int_{\Omega_{\mathbb{P}_i}} \mathbf{x} \cdot \sigma_1(\mathbf{x})\, d\Omega_{\mathbb{P}_i}}{\int_{\Omega_{\mathbb{P}_i}} \sigma_1(\mathbf{x})\, d\Omega_{\mathbb{P}_i}} \\ \mathbf{n}_f &= \frac{\int_{\Omega_{\mathbb{P}_i}} \mathbf{n}_1 \cdot \sigma_1(\mathbf{x})\, d\Omega_{\mathbb{P}_i}}{\| \int_{\Omega_{\mathbb{P}_i}} \mathbf{n}_1 \cdot \sigma_1(\mathbf{x})\, d\Omega_{\mathbb{P}_i} \|} \end{aligned} \tag{4.13}$$

The overall algorithm can be summarized as

Algorithm 5 Particle fracture approximation

Identify all $\mathbb{P}_i \subset \mathbb{R}^3$ as intact particles

Assign a random probability $r_i \in [0, 1]$ to each particle $\mathbb{P}_i$

for Quasi-static load step i = 1 to n **do**

Calculate stress field σ from XFEM solution

if No intact particle left in the domain **then**

Continue to next step i = i+1

for $\mathbb{P}_i \subset \mathbb{R}^3$ **do**

Compute Weibull fracture probability P_f^i

if $\max(P_f^i - r_i) > 0$ **then**

Break the particle $\mathbb{P}_i$ with the $\max(P_f^i - r_i)$

Create a strong discontinuous plane at $\mathbf{x}_f$ with normal direction $\mathbf{n}_f$

Mark $\mathbb{P}_i$ as failed particle, $\mathbb{P}_i \not\subset \mathbb{R}^3$

Roll back and resolve load step i

else

Continue to next step i+1

4.4 Example Problem of SiC Particle Reinforced Aluminium

To demonstrate the applicability of our XFEM algorithm for the fracture analysis of particle reinforced MMCs, we applied our method on X-ray tomography data of 2080 aluminum alloy (3.6% Cu, 1.9% Mg, 0.25% Zr) reinforced with 20 vol.% SiC particles (average particle size of 25 μm) was used in this study. The materials were processed by blending SiC and Al powders, compaction of the powder mixture, hot pressing, and hot extrusion (Alcoa Inc., Alcoa, PA). Details of the powder metallurgy process for fabrication of these composite materials can be found elsewhere [101].

Electro discharge machining (EDM) was used to obtain dog-bone specimens with a gage length of 2.5 mm and a 0.75 mm square cross-section. Specimens were machined parallel to the extrusion axis. In- situ uniaxial tensile tests were carried out on these specimens in the synchrotron using the loading stage shown in Figure 4.1. The load cell had a 500 N capacity, and the stepper motor had a captive linear actuator capable of 8 μm per step and 12 mm stroke. Actuator was moved to apply load on the sample. Scans were taken at different displacements until failure. The details of the loading stage has been provided elsewhere [12].

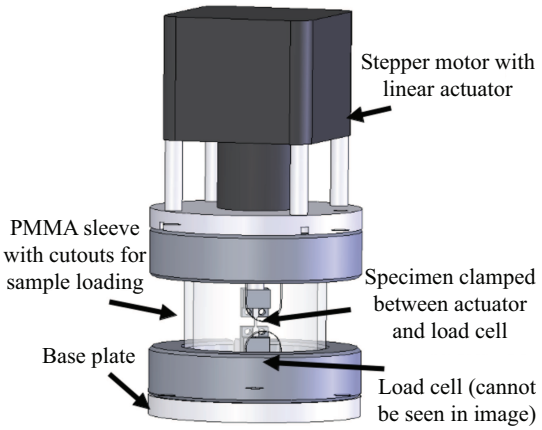


Figure 4.1: In situ loading stage to perform tensile testing on MMCs.

4.4.1 Numerical Modeling

The domain dimensions of the specimen gage section are 750×750×2500 μm. As shown in Figure 4.2, the fractured plane in the experiment is closed to one end of the gage section in the loading direction [122]. X-ray tomography is performed on a selected volume in the gage section and the relative position of the selected volume is shown in Figure 4.2. The selected volume is used as our simulation volume and the domain dimensions are 190×370×100 μm.

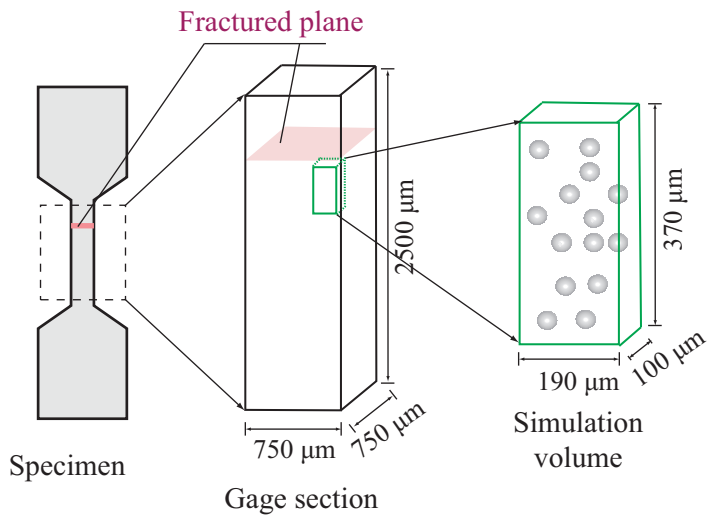


Figure 4.2: An illustration of domain dimensions and the relative location of the simulation volume in the gage section of specimen.

Brittle SiC particles are considered as linear isotropic before cleavage fracture. The Young's modulus and Poison's ratio of SiC particles are 410 GPa and 0.19, respectively [122]. Weibull distribution model is used to estimate the fracture probability of SiC particles by equation (4.12). The reference volume V_0, characteristic strength σ_f and Weibull modulus m are three material probabilities used to characterize the strength of brittle particles [118]. The different sets of the three parameters should obey $V_0\,\sigma_f^m = \overline{V_0}\,\overline{\sigma}_f^{\overline{m}}$. Llorca has estimated the three parameters for SiC particles in Al matrix [123]. In his work, the estimated Weibull modulus m is 6, characteristic strength σ_f is 1323 MPa, and the reference volume is set to be average particle volume, which is estimated as 7.5^3 μm^3. In this work, the average particle volume is 16800 μm^3, the Weibull modulus is 6, and the characteristic strength is calculated to be 715 MPa based on the relationship of the three parameters. The Young's modulus and Poison's ratio of the aluminum matrix are 74 GPa and 0.33, respectively. The aluminum matrix deforms plastically according to a nonlinear J_2 plastic law fitted

from an experimental stress-train curve of 2080-T6 aluminum [110]. The stress-strain relationship is fitted as $\sigma_y(\gamma) = C(1 - e^{-b\gamma}) + \sigma_{y_0}$, where σ_y is the yield stress, γ is the plastic strain, C =185.1 MPa, b =23.9 and the initial yield stress σ_{y_0} is 370 MPa. Figure 4.3 compares the fitted stress-strain curve with the experimental data.

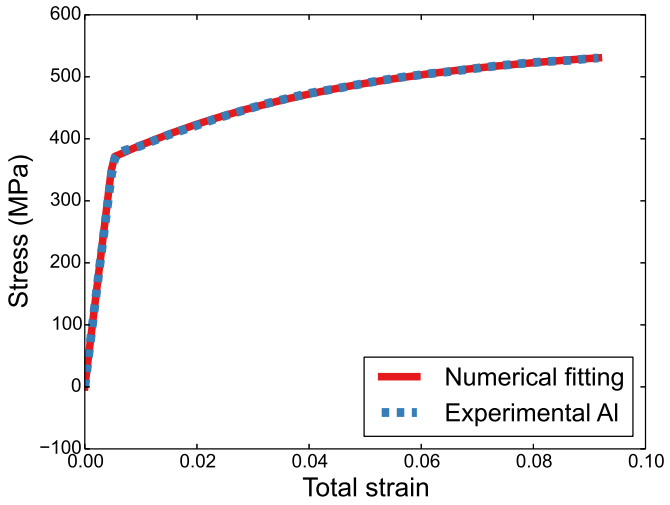


Figure 4.3: Fitted stress-strain curve of 2080-T6 aluminum.

4.4.2 Convergence Study of Weibull Probability

Weibull fracture probability of a particle is calculated by an integration of the stress field over all elements inside or partially inside the particle. The stress field is solved on a finite element mesh and the stress magnitude at a point is different under different mesh sizes, but the magnitude will converge to a value with more and more elements if there is not crack in the particle. We found that the fracture probability of a particle decreases with finer mesh used. However, it is hard to calculate the convergence rate of the fracture probability directly since the exact probability is an unknown value. It is reasonable to assume that the numerical Weibull fracture probability P is a summation of an exact probability P^{exact} and a probability error

$e(h)$. This relationship can be expressed as

$$P(h) = P^{\text{exact}} + e(h) \tag{4.14}$$

The exact Weibull probability is constant for a particle while the numerical Weibull probability and the error is a function of element size h. Taking derivative of equation (4.14) with respect to element size, we have

$$\frac{dP(h)}{dh} = \frac{de(h)}{dh} \tag{4.15}$$

In a log-log scale, if we can express the derivative of Weibull probability as a linear function with respect to element size, we have

$$\log_{10}\left(\frac{dP(h)}{dh}\right) = \beta + \alpha \log_{10}(h) = \log_{10}\left(10^{\beta} h^{\alpha}\right) \tag{4.16}$$

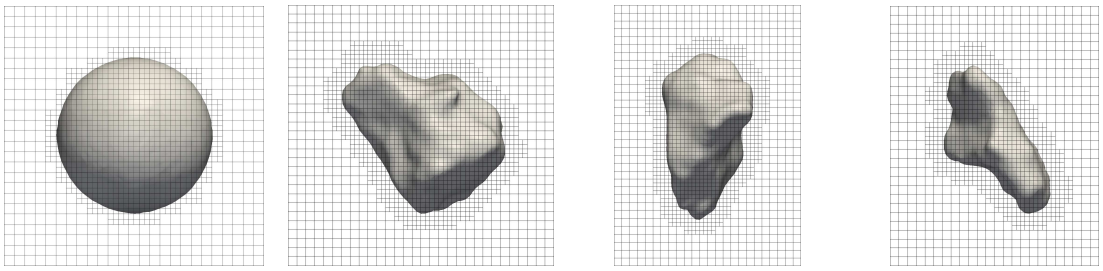
where α is the slope of the fitted line and β is the y-axis intercept. Combine equation (4.15) and equation (4.16), the probability error can be estimated as

$$\frac{de(h)}{dh} = 10^{\beta} h^{\alpha} \tag{4.17}$$

$$e(h) = \frac{10^{\beta}}{\alpha + 1} h^{\alpha+1} \tag{4.18}$$

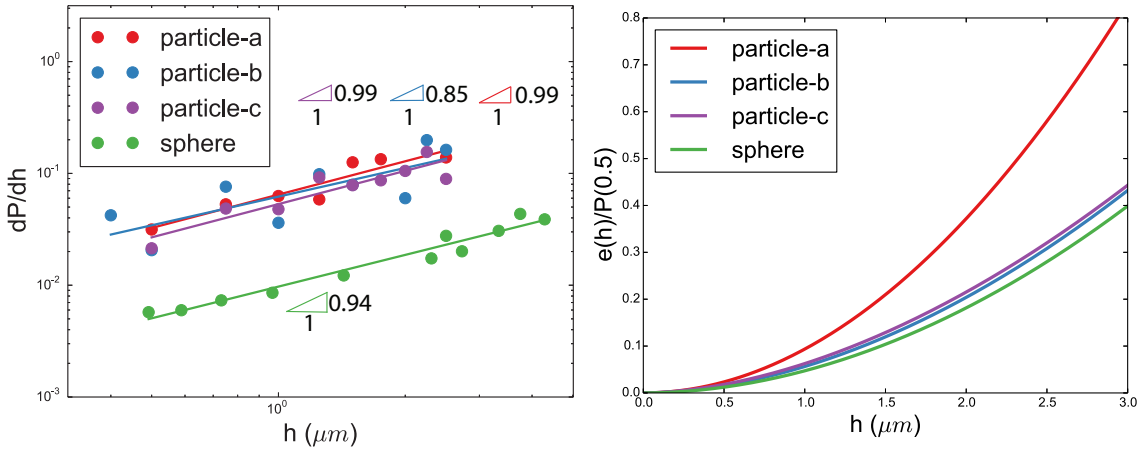
Therefore, the convergence rate of the fracture probability is $\alpha + 1$, where α is the convergence rate of the derivative of fracture probability. In the following, we studied the convergence of dP/dh for particles with different geometries. Figure 4.4 shows four particle geometries used in the convergence study. Figure 4.4a shows a sphere particle located at the center of a cube with length of 60 μm. The ratio of the sphere radius over the cube length is 0.3 and the sphere volume is 24429 μm^3. Figure 4.4b, 4.4c and 4.4d are three particles with realistic geometries. The three particles are shifted to the center and the domain lengths are chosen to maintain the same padding distance of matrix in three directions. The volume of the three particles are

27143, 23570 and 16424 μm^3, respectively. For all cases, only rigid body motions are fixed and traction is applied in Y direction. The applied traction is increased linearly from 0 to 400 MPa in four load steps. Oct-tree mesh is used and elements close to a material interface or inside a particle are sorted and refined with one time. Figure 4.5a shows the fitted convergence rates for the four particles. The convergence rates of dP/dh are close to 1. Therefore, the convergence rates of Weibull probability are approximate to 2, which equals to the convergence rate of L_2 error in linear elements. The fitted lines are $y = 0.99x - 1.19$, $y = 0.85x - 1.21$, $y = 0.99x - 1.27$, and $y = 0.94x - 2.01$, respectively. Figure 4.5b shows a relative probability error with respect to the element size used for particles. The relative error is calculated by the ratio of $e(p)$ over the fracture probability when element size is 0.5 μm. Particles with realistic geometry have higher probability error than the sphere under the same element size. When the element size is close to 1 μm, the probability error is less than 10 %.



(a) Sphere (b) Particle-a (c) Particle-b (d) Particle-c

Figure 4.4: XY view of particles in mesh wireframe that used in the convergence study.



(a) Convergence in particles

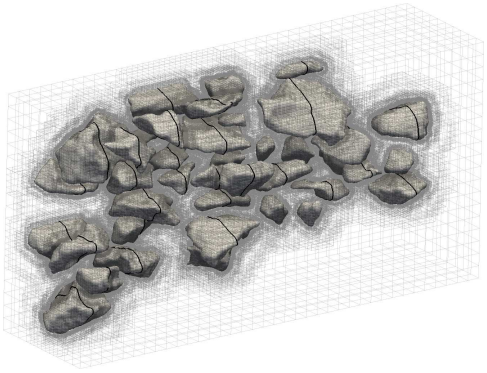
(b) Estimated probability error

Figure 4.5: Convergence of the derivative of Weibull fracture probability in particles with different geometries and the corresponding estimated probability error.

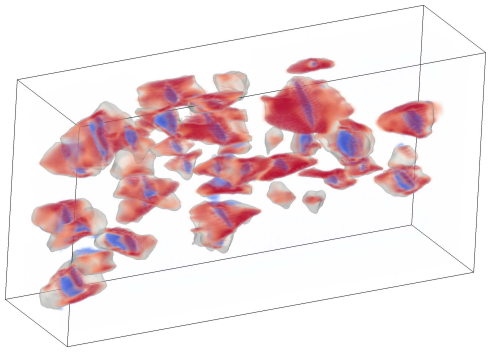
4.4.3 Result and Discussion

Figure 4.2 shows the domain size of the simulation volume and its relative position in the experimental specimen. We apply symmetric boundary conditions in the fracture analysis and the traction is applied on the top surface of the volume as shown in Figure 4.2. Quasistatic analysis is performed and the traction is increased linearly from 0 to 405 MPa in 18 steps. Oct-tree mesh is implemented and the initial uniform discretization is 19×10×37. Elements close to material interfaces or inside particles are sorted and refined. After three times of mesh refinement, the simulation volume is discretized with 560,226 elements and the element size used for particles is 1.25 μm. There are 41 particles in total and finally there are 31 fractured particles in the end. Figure 4.6a shows the 41 particle geometries in the mesh wireframe along with 31 crack surfaces denoted in black. Figure 4.6b shows axial stress distribution on particles. Blue stands for zero stress and the maximum axial stress is scaled to 1 GPa. The normal direction of fracture surfaces are computed by equation (4.13) and

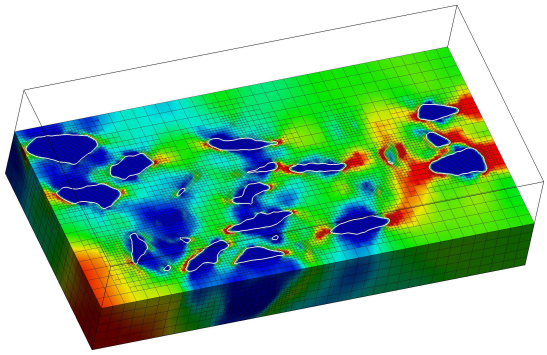
the numerical normal directions are close to the loading direction. Figure 4.6c and 4.6d are contours of effective strain and axial stress on a slice at the last load step, respectively.



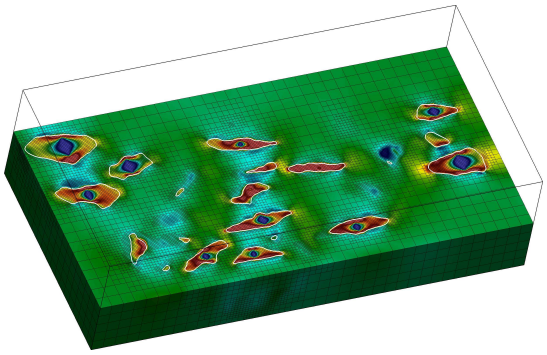
(a) Particle geometries.



(b) Axial stress distribution.



(c) Effective strain.



(d) Axial stress.

Figure 4.6: (a) Particle geometries in the oct-tree mesh wireframe. Break surfaces are denoted in black. (b) Axial stress distribution on particles. (c) Effective strain on a slice parallel to the loading direction. The traction is applied on left surface and the maximum effective strain is scaled to 0.008. (d) Axial stress on a slice parallel to the loading direction. The maximum axial stress is scaled to 1 GPa. The stress releases to zero on crack surfaces inside fractured particles.

Then, we compared our numerical stress-strain curve with the experimental result. In the experiment, the onset of damage in the specimen appears to begin close to 440

MPa [122]. In the numerical model, we still use the same simulation volume and the same mesh scheme, but increasing the traction from 0 to 450 MPa in 20 steps. Figure 4.7 shows the stress-strain comparison with the experiment. The numerical curve is a little above the experimental curve after yielding. In this work, only the first stage of fracture is modeled. Particle and matrix are considered as perfectly bonded and the matrix is idealized without fracture. However, in the experiment, there are localized void growth and the failure of matrix can increase the strain under the same traction [13]. Since there are some other fracture mechanisms that play a role in the experiment but not considered in the numerical model, the numerical stress-strain curve is a little higher in the yielding region.

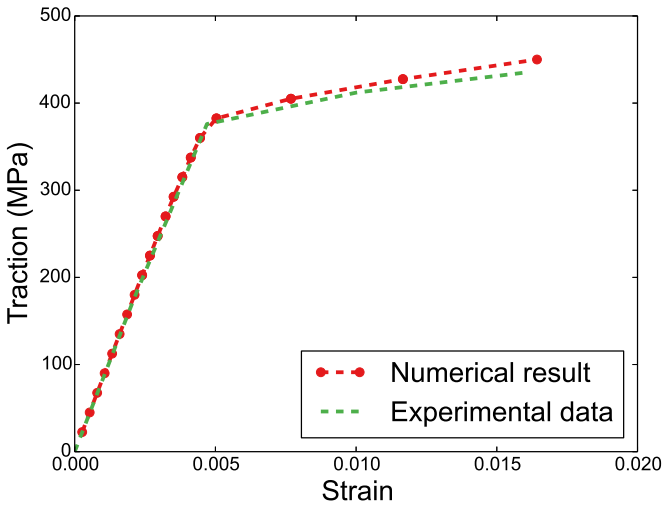


Figure 4.7: Comparison of numerical and experimental strain-stress curves.

There are 41 particles in the simulation model and the particle volume ranges from 1420.54 μm^3 to 36350.3 μm^3. The Weibull fracture model can account for particle size effect automatically. Large particles contain more defects and their Weibull fracture probabilities are correspondingly higher than small particles. Therefore, large particles tend to fracture before small particles. Figure 4.8 shows the average

volume of fractured particles under different applied loads. Particles start to break when the applied traction is 247.5 MPa. There are more small particles break under higher applied tractions and the average volume of fractured particles decreases along with traction.

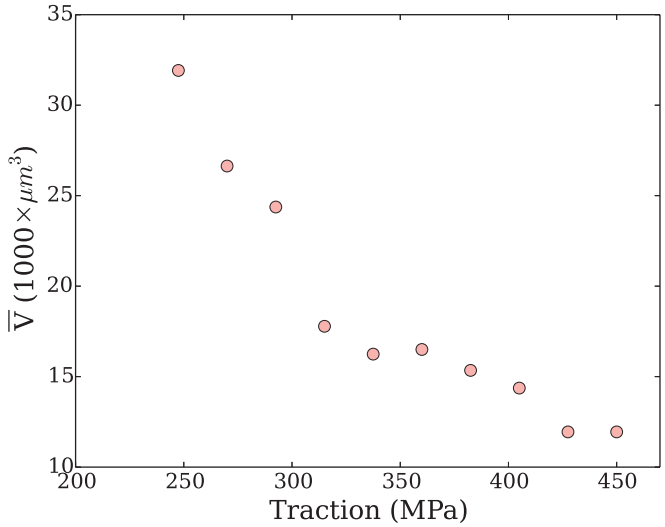
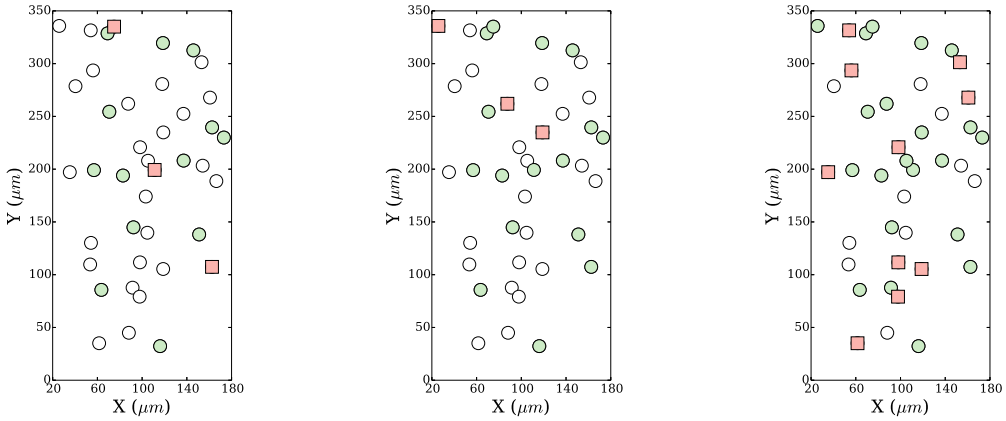


Figure 4.8: The average volume of fractured particles in each load step.

Figure 4.6a shows all break surfaces on 3D particles, but it is not straightforward to see the relative position of break particles since particles can interrupt the view of others. In order to distinctly visualize the relative positions of break particles, we projected the centroid positions of all particles to an XY plane, in which the Y direction is the loading direction. Figure 4.9 shows the particle centroid positions on a projected XY plane under three load steps. Each marker stands for one particle centroid and particles are split into three types: not cracked, cracked in previous load steps and cracked in current load step. Due to the randomness of critical fracture probability assigned on each particle, the sequence of cracked particles in a load step does not follow a straightforward order. But there are still some patterns we can find here: Particles tend to break in groups and particles adjacent to previously cracked

particles are more inclined to break. After one particle break, the stress field in an adjacent region surrounding the cracked particle will be redistributed and some not cracked particles in that region are inclined to support extra strength released by the broken particle. As a consequence, the stress on some adjacent particles will increase after one particle break, and the fracture probability on those particles become larger, correspondingly.



(a) Traction is 337.5 MPa. (b) Traction is 360 MPa. (c) Traction is 405 MPa.

Figure 4.9: Graphical representation of the relative position of SiC particle centroids on a projected XY plane that parallels to the loading direction. Solid circles colored in white represent particle centroids that are not cracked. Green circles and red squares are particle centroids that are "previously" cracked and "newly" cracked, respectively, relative to the previous load step.

In order to study the influence of one cracked particle on others, we calculated the change of fracture probability on not cracked particles before and after one particle break. Figure 4.10 shows particle centroid positions projected on an XY plane and the centroids are colored by ΔP, where ΔP is the change of fracture probability before and after one particle break. Figure 4.10a-4.10d show the distribution of ΔP on not cracked particle centroids after four broken particles. The fractured

particle can exert influences on particles not far away and the imposed influences can either increase or decrease the fracture probability on other particles. The fracture probability tends to increase on a particle that closes to the newly cracked particle and in the meantime, they are on the same XZ plane perpendicular to the loading direction. This tendency is straightforward to understand because the force on each cross section should be constant and balance the external applied load. After one particle break, the remaining particles on the same XZ plane have to support extra strength to keep the balance. While the fracture probability is inclined to decrease on a particle that is not far away to the newly cracked particle but their relative position is parallel to the loading direction. This tendency is caused by the change of stress flow before and after fracture. Before fracture, the top and bottom parts of the particle in the loading direction tend to have higher stress than the middle part. After the fracture, the stress flows to the crack tip and particles usually break around a cross section in the middle of the loading direction. Since the stress unloading of the top and bottom parts, the adjacent particles close to the top and bottom parts tend to withstand more strength.

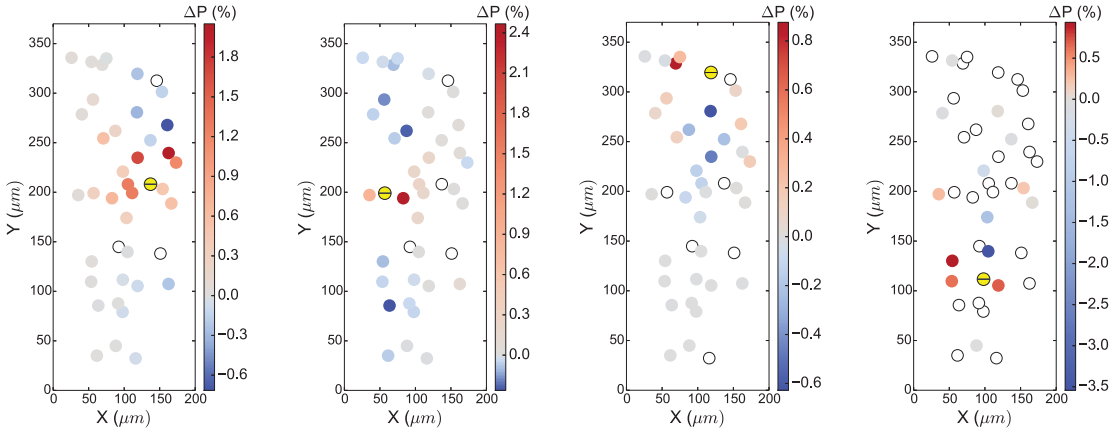
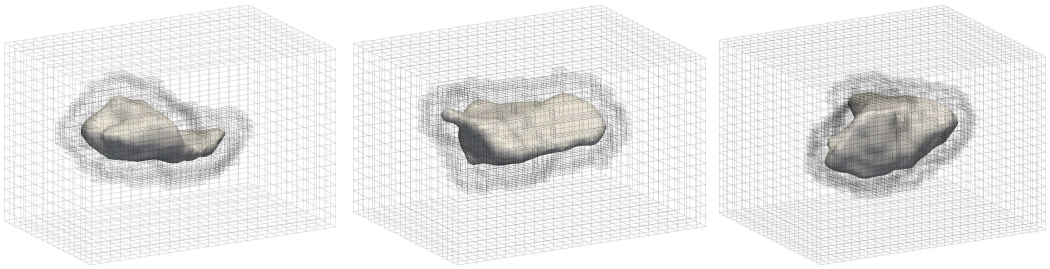


Figure 4.10: The influence of one newly cracked particle on the Weibull fracture probability of not cracked particles. Previously cracked particles are denoted in solid white circles. The newly cracked particle is denoted in a solid yellow circle with a black break surface. Not cracked particles are colored by the change of Weibull fracture probability before and after fracture.

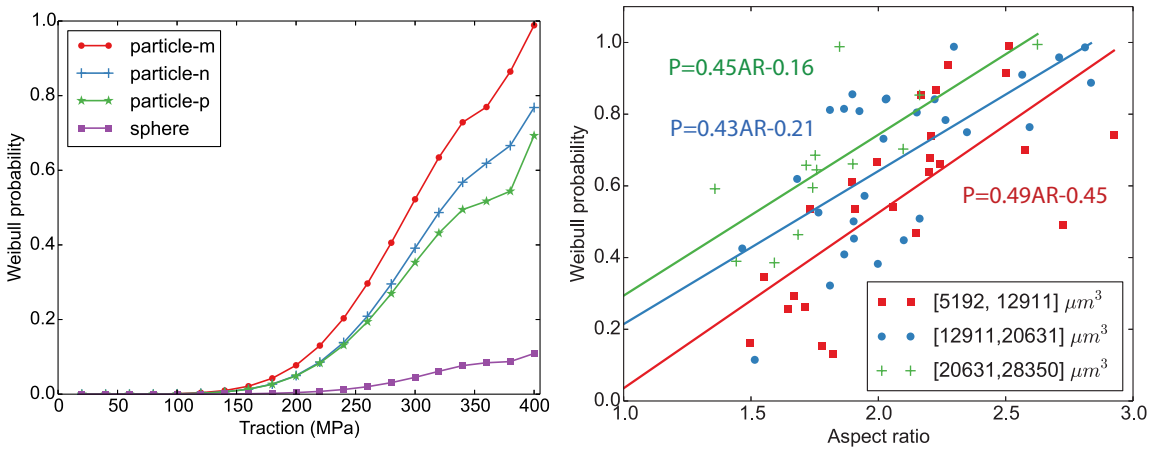
In order to characterize the influence of particle geometry on particle fracture probability, we studied Weibull fracture probability versus applied traction for several particle geometries separately. As shown in Figure 4.12a, Weibull probability increases much more rapidly along with traction for some particles. The three realistic particles, as shown in Figure 4.11, have much higher Weibull probability compared with sphere when traction is higher than 200 MPa. Particle-m has a narrow volume and its fracture probability is higher than others. Particle-n is flatter than particle-p and the corresponding fracture probability is higher under the same applied traction. In order to characterize the influence of particle geometry on the fracture probability, we studied the Weibull probability with respect to aspect ratio for about one hundred particles when applied traction is 400 MPa. The particle volumes range from 5192 μm^3 to 28350 μm^3. The particle aspect ratio is calculated by the ratio of the longest

ferret distance over the shortest ferret distance. As shown in Figure 4.12b, particles are separated into three groups with equal volume interval. Large particles generally have higher fracture probability and the fitted slopes are close for the three volume intervals. The coefficient of determination, denoted as R^2, is a number from 0 to 1, which can be used to indicate how well the regression line fits the data [124]. An R^2 of 1 indicates that the regression line perfectly fits the data. The R^2 of the three fitted lines are 0.71, 0.67 and 0.76, respectively. Each particle is shifted to the center of the simulation volume and the domain sizes of the volume are set to maintain a constant fill ratio for all cases.



(a) Particle-m (b) Particle-n (c) Particle-p

Figure 4.11: Particle geometries in mesh wireframe that used in history plot of fracture probability. Rigid body motions are fixed and traction is applied on both left and right surfaces.



(a) History plot of fracture probability. (b) Fracture probability versus aspect ratio.

Figure 4.12: Characteristic study of the influence of particle geometry on the magnitude of Weibull fracture probability.

Chapter 5

CONCLUSIONS

5.1 Summary

In summary, we have developed robust, accurate and adaptive algorithms in the framework of extended finite element method for the fracture analysis of highly heterogeneous materials with complex internal geometries. The algorithms are developed to automate the simulation of experiments and numerical solutions are compared and validated with experimental results.

In Chapter 2, we have developed an adaptive and parallelled geometry segmentation algorithm to robustly identify and separate discrete features from segmented tomography image data. The concept of betweenness centrality is introduced as an effective metric to identify narrow connecting regions between inclusions. The non-dimensional relative centrality is developed as a tool to identify bridge voxels, which reduces the influence of particle volume on centrality to the maximum extent and at the same time, graph topology can be completely reflected. With the help of relative centrality, our algorithm is capable to automatically detect narrow bridges for any particle regardless its geometry and volume. Besides that, the segmentation is performed at the cost of the minimum number of removed voxels and the initial geometrical characteristics of particles are maintained at the maximum extent. In the consideration of the XFEM modelling, a gap expansion algorithm is also developed and presented. The gap expansion is an optional process and the expected gap distance can be easily reset on the demand of users. The algorithm is demonstrated in 3D examples, but it can also segment 2D problems straightforwardly. More impor-

tantly, the algorithm is not restricted for one specific type of materials, but for any microstructure with spurious connections required to be segmented.

In Chapter 3, we have presented two level set techniques to construct distance fields for multiphase materials containing multiple, geometrically-complex phases. Both methods are implemented for oct-tree meshes with hanging nodes to resolve the fine details of the inclusion geometry efficiently. For the problems that were considered, the FMM is optimal in speed and accuracy for initializing the distance fields. Utilizing upwind finite differences eliminates the need for artificial viscosity which tends to round edges and artificially shrink small inclusions if the user-specified parameter is set too large.

In Chapter 4, we have implemented the Weibull distribution model in the LSM/XFEM algorithm for the modeling of fracture in brittle reinforced particles. The convergence rate of Weibull probability has been studied based on a single sphere and several particles with realistic geometry. The simulation volume used in fracture analysis is created with internal realistic particle geometries segmented in the gage section of the experimental specimen. Our numerical stress-strain curve can generally match with experimental result. We studied the histogram of fractured particles under different applied load and the influence of one fractured particle on its adjacent particles are discussed and presented. Finally, we applied characteristic study of particle geometry on the fracture probability.

5.2 Future Work

For the LSM developed in Chapter 3, the general approach of formulating an evolution equation based on the LSM affords greater flexibility. For instance, additional terms could easily be added to eliminate possible overlap between inclusions. Furthermore, other post processing, such as eliminating spurious bridges between adjacent

inclusions or particles resulting from errors in the image segmentation process can be devised for the LSM.

In the fracture model developed in Chapter 4, only the first stage of fracture is modeled. Particle and matrix are considered as perfectly bonded and the matrix is idealized without fracture. However, in the experiment, there are localized void growth and the failure of matrix can significantly increase the strain under the same traction. New algorithms need to be incorporated to account for the material interface debonding and the fracture of plastic matrix.

REFERENCES

[1] Z. P. Bazant, M. R. Tabbara, M. T. Kazemi, G. Pijaudier-Cabot, Random particle model for fracture of aggregate or fiber composites, Journal of Engineering Mechanics 116 (8) (1990) 1686–1705.

[2] P. H. Geubelle, J. S. Baylor, Impact-induced delamination of composites: a 2D simulation, Composites Part B: Engineering 29 (5) (1998) 589–602.

[3] R. Sidhu, N. Chawla, Three-dimensional microstructure characterization of Ag_3Sn intermetallics in Sn-rich solder by serial sectioning, Materials Characterization 52 (3) (2004) 225–230.

[4] E. Moreau, B. Velde, F. Terribile, Comparison of 2D and 3D images of fractures in a vertisol, Geoderma 92 (1) (1999) 55–72.

[5] C. R. Monks, B. A. Freiberg, H. Kupfer, N. Sciaky, A. Kupfer, Three-dimensional segregation of supramolecular activation clusters in T cells, Nature 395 (6697) (1998) 82–86.

[6] A. L. Bull, A. K. McNamara, J. Ritsema, Synthetic tomography of plume clusters and thermochemical piles, Earth and Planetary Science Letters 278 (3) (2009) 152–162.

[7] E. Meslin, B. Radiguet, P. Pareige, A. Barbu, Kinetic of solute clustering in neutron irradiated ferritic model alloys and a french pressure vessel steel investigated by atom probe tomography, Journal of Nuclear Materials 399 (2) (2010) 137–145.

[8] M. Miller, E. Kenik, Atom probe tomography: A technique for nanoscale characterization, Microscopy and Microanalysis 10 (03) (2004) 336–341.

[9] P. A. Midgley, R. E. Dunin-Borkowski, Electron tomography and holography in materials science, Nature materials 8 (4) (2009) 271–280.

[10] K. Yazzie, J. Williams, N. Phillips, F. De Carlo, N. Chawla, Multiscale microstructural characterization of Sn-rich alloys by three dimensional (3D) X-ray synchrotron tomography and focused ion beam (FIB) tomography, Materials Characterization 70 (2012) 33–41.

[11] S. Stock, X-ray microtomography of materials, International Materials Reviews 44 (4) (1999) 141–164.

[12] S. S. Singh, J. J. Williams, P. Hruby, X. Xiao, F. De Carlo, N. Chawla, In situ experimental techniques to study the mechanical behavior of materials using X-ray synchrotron tomography, Integrating Materials and Manufacturing Innovation 3 (1) (2014) 9.

[13] J. Williams, Z. Flom, A. Amell, N. Chawla, X. Xiao, F. De Carlo, Damage evolution in SiC particle reinforced Al alloy matrix composites by X-ray synchrotron tomography, Acta Materialia 58 (18) (2010) 6194–6205.

[14] S. S. Singh, C. Schwartzstein, J. J. Williams, X. Xiao, F. De Carlo, N. Chawla, 3D microstructural characterization and mechanical properties of constituent particles in Al 7075 alloys using X-ray synchrotron tomography and nanoindentation, Journal of Alloys and Compounds 602 (2014) 163–174.

[15] E. Padilla, V. Jakkali, L. Jiang, N. Chawla, Quantifying the effect of porosity on the evolution of deformation and damage in Sn-based solder joints by X-ray microtomography and microstructure-based finite element modeling, Acta Materialia 60 (9) (2012) 4017–4026.

[16] S. S. Singh, J. J. Williams, X. Xiao, F. De Carlo, N. Chawla, In situ three dimensional (3D) X-ray synchrotron tomography of corrosion fatigue in Al7075 alloy, John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012.

[17] S. C. Johnson, Hierarchical clustering schemes, Psychometrika 32 (3) (1967) 241–254.

[18] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, California, USA, 1967, pp. 281–297.

[19] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., On spectral clustering: Analysis and an algorithm, Advances in neural information processing systems 2 (2002) 849–856.

[20] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise., in: Kdd, Vol. 96, 1996, pp. 226–231.

[21] J. Sander, M. Ester, H.-P. Kriegel, X. Xu, Density-based clustering in spatial databases: The algorithm gdbscan and its applications, Data mining and knowledge discovery 2 (2) (1998) 169–194.

[22] R. A. Ketcham, Computational methods for quantitative analysis of three-dimensional features in geological specimens, Geosphere 1 (1) (2005) 32–41.

[23] Y. Peng, Q. Kang, H. Cheng, X. Li, M. H. Sun, W. Jiang, H. H. Luu, J. Y. Park, R. C. Haydon, T.-C. He, Transcriptional characterization of bone morphogenetic proteins (BMPs)-mediated osteogenic signaling, Journal of cellular biochemistry 90 (6) (2003) 1149–1165.

[24] L. T. Stephenson, M. P. Moody, P. V. Liddicoat, S. P. Ringer, S. Byers, A. Rafferty, A. Cerezo, L. Davin, P. Clark, F. Evans, et al., New techniques for the analysis of fine-scaled clustering phenomena within atom probe tomography (APT) data, Microscopy and Microanalysis 13 (6) (2007) 448–463.

[25] M. Steinbach, G. Karypis, V. Kumar, et al., A comparison of document clustering techniques, in: KDD workshop on text mining, Vol. 400, Boston, 2000, pp. 525–526.

[26] S. Ray, R. H. Turi, Determination of number of clusters in K-means clustering and application in colour image segmentation, in: Proceedings of the 4th international conference on advances in pattern recognition and digital techniques, 1999, pp. 137–143.

[27] Y. Weiss, Segmentation using eigenvectors: A unifying view, in: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, Vol. 2, IEEE, 1999, pp. 975–982.

[28] R. J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: Advances in Knowledge Discovery and Data Mining, Springer, 2013, pp. 160–172.

[29] L. C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977) 35–41.

[30] A. M. Martín González, B. Dalsgaard, J. M. Olesen, Centrality measures and the importance of generalist species in pollination networks, Ecological Complexity 7 (1) (2010) 36–43.

[31] O. Sporns, Network attributes for segregation and integration in the human brain, Current opinion in neurobiology 23 (2) (2013) 162–171.

[32] H. Jeong, S. P. Mason, A.-L. Barabási, Z. N. Oltvai, Lethality and centrality in protein networks, Nature 411 (6833) (2001) 41–42.

[33] J. M. Anthonisse, The rush in a graph, Amsterdam: University of Amsterdam Mathematical Centre.

[34] L. C. Freeman, S. P. Borgatti, D. R. White, Centrality in valued graphs: A measure of betweenness based on network flow, Social networks 13 (2) (1991) 141–154.

[35] T. H. Cormen, Introduction to algorithms, MIT press, 2009.

[36] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, Journal of the ACM (JACM) 24 (1) (1977) 1–13.

[37] U. Brandes, A faster algorithm for betweenness centrality, Journal of Mathematical Sociology 25 (2) (2001) 163–177.

[38] G. Csardi, T. Nepusz, The igraph software package for complex network research, InterJournal, Complex Systems 1695 (5).

[39] V. Kouznetsova, M. Geers, W. Brekelmans, Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme, International Journal for Numerical Methods in Engineering 54 (2002) 1235–1260.

[40] S. Pimenta, R. Gutkin, S. Pinho, P. Robinson, A micromechanical model for kink-band formation: Part I–Experimental study and numerical modelling, Composites Science and Technology 69 (7) (2009) 948–955.

[41] J. Desrues, G. Viggiani, Strain localization in sand: An overview of the experimental results obtained in Grenoble using stereophotogrammetry, International Journal for Numerical and Analytical Methods in Geomechanics 28 (4) (2004) 279–321.

[42] C. Kung, M. Fine, Fatigue crack initiation and microcrack growth in 2024-T4 and 2124-T4 aluminum alloys, Metallurgical and Materials Transactions A 10 (5) (1979) 603–610.

[43] Y. Zhang, C. Bajaj, Adaptive and quality quadrilateral/hexahedral meshing from volumetric data, Computer methods in applied mechanics and engineering 195 (9) (2006) 942–960.

[44] Y. Zhang, T. J. Hughes, C. L. Bajaj, An automatic 3D mesh generation method for domains with multiple materials, Computer methods in applied mechanics and engineering 199 (5) (2010) 405–415.

[45] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of computational physics 79 (1) (1988) 12–49.

[46] J. A. Sethian, Analysis of flame propagation, Tech. rep., Lawrence Berkeley Lab., CA (USA) (1982).

[47] S. Nagrath, K. E. Jansen, R. T. Lahey, Computation of incompressible bubble dynamics with a stabilized finite element level set method, Computer Methods in Applied Mechanics and Engineering 194 (42) (2005) 4565–4587.

[48] J. A. Sethian, J. Straint, Crystal growth and dendritic solidification, Journal of Computational Physics 98 (2) (1992) 231–253.

[49] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, Journal of Computational Physics 135 (1) (1997) 8–29.

[50] J. A. Sethian, Tracking interfaces with level sets: An "act of violence" helps solve evolving interface problems in geometry, fluid mechanics, robotic navigation and materials sciences, American Scientist (1997) 254–263.

[51] D. Adalsteinsson, J. Sethian, A level set approach to a unified model for etching, deposition, and lithography, Journal of computational physics 138 (1) (1997) 193–223.

[52] X. Han, C. Xu, J. L. Prince, A topology preserving level set method for geometric deformable models, Pattern Analysis and Machine Intelligence, IEEE Transactions on 25 (6) (2003) 755–768.

[53] D. Adalsteinsson, J. A. Sethian, Transport and diffusion of material quantities on propagating interfaces via level set methods, Journal of Computational Physics 185 (1) (2003) 271–288.

[54] S. Osher, R. P. Fedkiw, Level set methods: An overview and some recent results, Journal of Computational physics 169 (2) (2001) 463–502.

[55] N. Sukumar, D. L. Chopp, N. Moës, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, Computer methods in applied mechanics and engineering 190 (46) (2001) 6183–6200.

[56] B. Merriman, J. K. Bence, S. J. Osher, Motion of multiple junctions: A level set approach, Journal of Computational Physics 112 (2) (1994) 334–363.

[57] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1) (1994) 146–159.

[58] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, Proceedings of the National Academy of Sciences 93 (4) (1996) 1591–1595.

[59] D. L. Chopp, Some improvements of the fast marching method, SIAM Journal on Scientific Computing 23 (1) (2001) 230–244.

[60] A. Telea, An image inpainting technique based on the fast marching method, Journal of graphics tools 9 (1) (2004) 23–34.

[61] N. Sukumar, D. Chopp, B. Moran, Extended finite element method and fast marching method for three-dimensional fatigue crack propagation, Engineering Fracture Mechanics 70 (1) (2003) 29–48.

[62] N. Sukumar, D. Chopp, E. Béchet, N. Moës, Three-dimensional non-planar crack growth by a coupled extended finite element and fast marching method, International Journal for Numerical Methods in Engineering 76 (5) (2008) 727.

[63] S. Garrido, L. Moreno, M. Abderrahim, F. Martin, Path planning for mobile robot navigation using voronoi diagram and fast marching, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, IEEE, 2006, pp. 2376–2381.

[64] J. A. Sethian, Fast marching methods, SIAM review 41 (2) (1999) 199–235.

[65] J. A. Sethian, Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, Vol. 3, Cambridge university press, 1999.

[66] J. A. Sethian, A. M. Popovici, 3-D traveltime computation using the fast marching method, Geophysics 64 (2) (1999) 516–523.

[67] K. K. Chawla, Metal matrix composites, Wiley Online Library, 2006.

[68] A. Ayyar, G. Crawford, J. Williams, N. Chawla, Numerical simulation of the effect of particle spatial distribution and strength on tensile behavior of particle reinforced composites, Computational Materials Science 44 (2) (2008) 496–506.

[69] Z. P. Bažant, M. R. Tabbara, M. T. Kazemi, G. Pijaudier-Cabot, Random particle model for fracture of aggregate or fiber composites, Journal of engineering mechanics.

[70] L. Mishnaevsky, K. Derrien, D. Baptiste, Effect of microstructure of particle reinforced composites on the damage evolution: Probabilistic and numerical analysis, Composites Science and Technology 64 (12) (2004) 1805–1818.

[71] D. Bray, S. Gilmour, F. Guild, A. Taylor, The effects of particle morphology on the analysis of discrete particle dispersion using Delaunay tessellation, Composites Part A: Applied Science and Manufacturing 54 (2013) 37–45.

[72] T. L. Anderson, T. Anderson, Fracture mechanics: Fundamentals and applications, CRC press, 2005.

[73] S. T. Rolfe, J. M. Barsom, Fracture and fatigue control in structures: Applications of fracture mechanics, ASTM International, 1977.

[74] J. Berry, Some kinetic considerations of the Griffith criterion for fracture-I: Equations of motion at constant force, Journal of the Mechanics and Physics of Solids 8 (3) (1960) 194–206.

[75] R. Nuismer, An energy release rate criterion for mixed mode fracture, International journal of fracture 11 (2) (1975) 245–250.

[76] J. Newman, I. Raju, An empirical stress-intensity factor equation for the surface crack, Engineering Fracture Mechanics 15 (1) (1981) 185–192.

[77] E. F. Rybicki, M. Kanninen, A finite element calculation of stress intensity factors by a modified crack closure integral, Engineering Fracture Mechanics 9 (4) (1977) 931–938.

[78] A. A. Griffith, The phenomena of rupture and flow in solids, Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character (1921) 163–198.

[79] G. IRWIN, Onset of fast crack propagation in high strength steel and aluminum, in: Proc. Sec. Sagamore Conf, Vol. 2, p. 289.

[80] G. R. Irwin, Analysis of stresses and strains near the end of a crack traversing a plate, SPIE MILESTONE SERIES MS 137 (1997) 167–170.

[81] H. Westergaard, Bearing pressures and cracks, Journal of applied mechanics.

[82] I. Sneddon, The distribution of stress in the neighbourhood of a crack in an elastic solid, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, Vol. 187, The Royal Society, 1946, pp. 229–260.

[83] F. Mudry, A local approach to cleavage fracture, Nuclear Engineering and design 105 (1) (1987) 65–76.

[84] C. Ruggieri, X. Gao, R. H. Dodds Jr, Transferability of elastic–plastic fracture toughness using the Weibull stress approach: Significance of parameter calibration, Engineering fracture mechanics 67 (2) (2000) 101–117.

[85] R. C. Hidalgo, Y. Moreno, F. Kun, H. J. Herrmann, Fracture model with variable range of interaction, Physical review E 65 (4) (2002) 046148.

[86] L. Xia, C. F. Shih, Ductile crack growth-III. Transition to cleavage fracture incorporating statistics, Journal of the Mechanics and Physics of Solids 44 (4) (1996) 603–639.

[87] A. Eckschlager, W. Han, H. J. Böhm, A unit cell model for brittle fracture of particles embedded in a ductile matrix, Computational materials science 25 (1) (2002) 85–91.

[88] R. Doremus, Fracture statistics: A comparison of the normal, Weibull, and type I extreme value distributions, Journal of applied physics 54 (1) (1983) 193–198.

[89] C. Lu, R. Danzer, F. D. Fischer, Fracture statistics of brittle materials: Weibull or normal distribution, Physical Review E 65 (6) (2002) 067102.

[90] X. Gao, C. Ruggieri, R. H. Dodds Jr, Calibration of Weibull stress parameters using fracture toughness data, International Journal of Fracture 92 (2) (1998) 175–200.

[91] X. Gao, R. Dodds, R. Tregoning, J. Joyce, R. Link, A Weibull stress model to predict cleavage fracture in plates containing surface cracks, Fatigue and Fracture of Engineering Materials and Structures 22 (1999) 481–494.

[92] W. Weibull, A statistical theory of the strength of materials, no. 151, Generalstabens litografiska anstalts förlag, 1939.

[93] J. B. Quinn, G. D. Quinn, A practical and systematic review of Weibull statistics for reporting strengths of dental materials, dental materials 26 (2) (2010) 135–147.

[94] D. Adalsteinsson, J. A. Sethian, A fast level set method for propagating interfaces, Journal of computational physics 118 (2) (1995) 269–277.

[95] R. Yuan, S. S. Singh, N. Chawla, J. Oswald, Efficient methods for implicit geometrical representation of complex material microstructures, International Journal for Numerical Methods in Engineering 98 (2) (2014) 79–91.

[96] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, International journal for numerical methods in engineering 45 (5) (1999) 601–620.

[97] J. M. Melenk, I. Babuška, The partition of unity finite element method: Basic theory and applications, Computer methods in applied mechanics and engineering 139 (1) (1996) 289–314.

[98] D. Huynh, T. Belytschko, The extended finite element method for fracture in composite materials, International Journal for Numerical Methods in Engineering 77 (2) (2009) 214–239.

[99] C. Ye, J. Shi, G. J. Cheng, An extended finite element method (XFEM) study on the effect of reinforcing particles on the crack propagation behavior in a metal–matrix composite, International Journal of Fatigue 44 (2012) 151–156.

[100] Z. Wang, L. Ma, L. Wu, H. Yu, Numerical simulation of crack growth in brittle matrix of particle reinforced composites using the XFEM technique, Acta Mechanica Solida Sinica 25 (1) (2012) 9–21.

[101] N. Chawla, J. Jones, C. Andres, J. Allison, Effect of SiC volume fraction and particle size on the fatigue resistance of a 2080 Al/SiC_p composite, Metallurgical and Materials Transactions A 29 (11) (1998) 2843–2854.

[102] F. De Carlo, B. Tieman, High-throughput X-ray microtomography system at the Advanced Photon Source beamline 2-BM, in: Optical Science and Technology, the SPIE 49th Annual Meeting, International Society for Optics and Photonics, 2004, pp. 644–651.

[103] W. A. Barrett, E. N. Mortensen, Interactive live-wire boundary extraction, Medical Image Analysis 1 (4) (1997) 331–341.

[104] P. Hansbo, Explicit streamline diffusion finite element methods for the compressible Euler equations in conservation variables, Journal of Computational Physics 109 (2) (1993) 274–288.

[105] J. Chessa, P. Smolinski, T. Belytschko, The extended finite element method(XFEM) for solidification problems, International Journal for Numerical Methods in Engineering 53 (2002) 1959–1977.

[106] H. Antia, Numerical Methods for Scientists and Engineers, Birkhäuser Basel, 2002.

[107] T.-P. Fries, A. Byfut, A. Alizada, K. W. Cheng, A. Schröder, Hanging nodes and XFEM, International Journal for Numerical Methods in Engineering 86 (4-5) (2011) 404–430.

[108] J. Williams, Z. Flom, A. Amell, N. Chawla, X. Xiao, F. De Carlo, Damage evolution in SiC particle reinforced Al alloy matrix composites by X-ray synchrotro tomography, Acta Materialia 58 (18) (2010) 6194–6205.

[109] L. L. Mishnaevsky, Automatic voxel-based generation of 3D microstructural FE models and its application to the damage analysis of composites, Materials Science and Engineering: A 407 (1) (2005) 11–23.

[110] N. Chawla, U. Habel, Y.-L. Shen, C. Andres, J. Jones, J. Allison, The effect of matrix microstructure on the tensile and fatigue behavior of SiC particle-reinforced 2080 Al matrix composites, Metallurgical and Materials Transactions A 31 (2) (2000) 531–540.

[111] T. Belytschko, C. Parimi, N. Moës, N. Sukumar, S. Usui, Structured extended finite element methods for solids defined by implicit surfaces, International journal for numerical methods in engineering 56 (4) (2003) 609–635.

[112] M. Stolarska, D. Chopp, N. Moës, T. Beltyschko, Modelling crack growth by level sets, in: ECF14, Cracow 2002, 2013.

[113] A. Gravouil, N. Moës, T. Belytschko, Non-planar 3D crack growth by the extended finite element and level sets-Part II: Level set update, International Journal for Numerical Methods in Engineering 53 (11) (2002) 2569–2586.

[114] T. Belytschko, R. Gracie, G. Ventura, A review of extended/generalized finite element methods for material modeling, Modelling and Simulation in Materials Science and Engineering 17 (4) (2009) 043001.

[115] Y. Krongauz, T. Belytschko, EFG approximation with discontinuous derivatives, International Journal for Numerical Methods in Engineering 41 (7) (1998) 1215–1233.

[116] N. Moës, M. Cloirec, P. Cartraud, J.-F. Remacle, A computational approach to handle complex microstructure geometries, Computer methods in applied mechanics and engineering 192 (28) (2003) 3163–3177.

[117] W. Weibull, A statistical distribution function of wide applicability, Journal of applied mechanics 18 (1951) 292–297.

[118] C. Lewis, P. Withers, Weibull modelling of particle cracking in metal matrix composites, Acta metallurgica et materialia 43 (10) (1995) 3685–3699.

[119] M. Li, S. Ghosh, O. Richmond, H. Weiland, T. Rouns, Three dimensional characterization and modeling of particle reinforced metal matrix composites part II: Damage characterization, Materials Science and Engineering: A 266 (1) (1999) 221–240.

[120] T. Antretter, F. Fischer, Particle cleavage and ductile crack growth in a two-phase composite on a microscale, Computational materials science 13 (1) (1998) 1–7.

[121] K. Wallin, T. Saario, K. Törrönen, Fracture of brittle particles in a ductile matrix, International journal of fracture 32 (3) (1986) 201–209.

[122] J. Williams, N. Chapman, V. Jakkali, V. Tanna, N. Chawla, X. Xiao, F. De Carlo, Characterization of damage evolution in SiC particle reinforced Al alloy matrix composites by in-situ X-ray synchrotron tomography, Metallurgical and Materials Transactions A 42 (10) (2011) 2999–3005.

[123] C. Gonzalez, J. Llorca, Prediction of the tensile stress-strain curve and ductility in al/sic composites, Scripta materialia 35 (1) (1996) 91–97.

[124] N. R. Draper, H. Smith, E. Pownell, Applied regression analysis, Vol. 3, Wiley New York, 1966.