

# **Statistical Analysis and Description of Microstructures**

**Dual Degree Project**

By

**Iyer Adithya Jairam**

Roll No. 16D110011

Under the guidance of

**Prof. M P Gururajan and Prof. Hina Gokhale**



Metallurgical Process Engineering

Department of Metallurgical Engineering and Materials Science

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

2021

## Dissertation Approval Sheet

The dissertation entitled " **Statistical Analysis and Description of Microstructures**" submitted by **Iyer Adithya Jairam**(Roll No:16D110011), is approved for the degree of Master of Technology in Metallurgical Process Engineering from Indian Institute of Technology, Bombay.

Date: 8 July 2021

Digital Signature  
Gururajan Pandurangan  
Mogadalai (i09074)  
08-Jul-21 01:40:36 PM

---

Prof. M P Gururajan

(Project Guide)

Digital Signature  
Nagamani Balila (i16280)  
08-Jul-21 01:43:40 PM

---

Prof. Nagamani Jaya Balila

(Chairman and Examiner)



---

Dr R Sankarasubramanian

(Examiner)

## Declaration

I declare that this written submission represents my ideas in my own words, and where other's ideas or words have been included, I have adequately cited and referenced the original source. I also declare that I have adhered to all academic honesty and integrity principles and have not misrepresented or fabricated, or falsified any idea/data/fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can evoke penal action from the sources which have thus not been appropriately cited or from whom the proper submission has not been taken when needed.

Digital Signature IYER ADITHYA JAIRAM (16d110011) 08-Jul-21 01:33:38 PM
--

Iyer Adithya Jairam

(16D110011)

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Literature Survey</b>	<b>3</b>
2.1	Phase Field Modeling . . . . .	4
2.2	Homogenisation Theory . . . . .	5
2.3	Spatial Features . . . . .	6
2.3.1	N-point correlation function . . . . .	7
2.3.2	Linear Path Function . . . . .	7
2.3.3	Geometric and Further statistical Parameters . . . . .	8
2.4	Reconstruction Methods . . . . .	9
2.4.1	The Yeong and Torquato (YT) reconstruction method . . . . .	9
2.5	Building structure-property relations . . . . .	10
2.6	Microstructure Evolution . . . . .	11
<b>3</b>	<b>Methodology and Formulation</b>	<b>13</b>
3.1	Image Data Description and Pre-processing . . . . .	13
3.1.1	Data Description . . . . .	13
3.1.2	Data Pre-processing . . . . .	13
3.2	Calculating homogenized strains . . . . .	14
3.3	Calculation of elastic constants . . . . .	16
3.4	2 Point Statistics . . . . .	18
3.4.1	Formulation . . . . .	18
3.4.2	Algorithm . . . . .	19
3.4.3	Visualization . . . . .	20
3.5	Level Set Methods and Velocity Formulation . . . . .	21
3.5.1	Level Set Methods: Formulation . . . . .	22
3.5.2	Algorithm . . . . .	23
3.5.3	Visualization . . . . .	24
3.6	Hoshen-Kopelman Algorithm and Application . . . . .	24

3.6.1	Formulation: General Hoshen-Kopelman . . . . .	24
3.6.2	Formulation : Hoshen Kopleman for PBC . . . . .	25
3.6.3	Visualization . . . . .	27
3.7	Shape/size quantification and Precipitate Tracking . . . . .	27
3.7.1	Shape/size quantification . . . . .	27
3.7.2	Precipitate tracking . . . . .	29
3.7.3	Visualization . . . . .	32
3.8	Convexity measures . . . . .	32
3.8.1	Definition . . . . .	32
3.8.2	Monte Carlo method . . . . .	34
3.8.3	Testing convexity on precipitate growth . . . . .	34
3.8.4	Convexity for complete microstructural images . . . . .	35
3.8.5	Calculating Short Range Averaged Convexity using different box sizes . . . . .	37
<b>4</b>	<b>Results and Discussion</b>	<b>38</b>
4.1	Characterisation examples and use cases . . . . .	39
4.1.1	Datasets used for comparison . . . . .	39
4.1.2	2-Point Correlations . . . . .	39
4.1.3	Using Clustering algorithms for precipitate counting and size estimation . . . . .	44
4.2	Comparison between Isotropic, Hexagonal and Cubic Microstructures	46
4.2.1	Auto-Correlation Heat-maps . . . . .	46
4.2.2	Radial Correlations . . . . .	49
4.2.3	Precipitate Size Statistics . . . . .	52
4.3	Precipitate tracking . . . . .	54
4.3.1	Directional growth . . . . .	55
4.3.2	Identifying Precipitate Coalescence . . . . .	56
4.3.3	Precipitate inclination . . . . .	58
4.4	Elastic constants . . . . .	60
4.5	Microstructure Property Correlation . . . . .	65
<b>5</b>	<b>Summary</b>	<b>68</b>

# List of Figures

3.1	Visualizing Composition Profile: The Composition profile along a line visualized post processing. . . . .	14
3.2	Visualizing Correlations . . . . .	21
3.3	Contours on the microstructure . . . . .	22
3.4	Velocity visualized on a quiver plot . . . . .	24
3.5	Cluster counting and labelling(HKLM Image) . . . . .	27
3.6	Visualizing Precipitates . . . . .	32
3.7	Visualizing Concave and Convex shapes . . . . .	33
3.8	Visualizing Convexity Number with Precipitate Growth . . . . .	35
3.9	Convexity number change with Microstructure Evolution ( $t_3 > t_2 > t_1$ )	36
3.10	Convexity change with different image divisions . . . . .	37
4.1	Visualizing Correlation for Isotropic Microstructures . . . . .	40
4.2	Visualizing Correlation for Hexagonal Microstructures . . . . .	40
4.3	Plotting Radial Auto Correlations . . . . .	42
4.4	Plotting radial auto-correlations vs time . . . . .	42
4.5	Comparing directional growth . . . . .	43
4.6	Particle Counting with time . . . . .	44
4.7	Precipitate Size statistics . . . . .	45
4.9	Comparison of Auto-correlation Heatmaps . . . . .	47
4.10	Left to Right: Increasing Composition, Top to bottom: Isotropic, Hexagonal and Cubic . . . . .	48
4.11	Auto-correlation variation with time for Hexagonal Microstructure with C=0.4 . . . . .	49
4.12	Comparison of Radial Auto-correlations with time between Isotropic, Hexagonal and Cubic for composition of 0.4 . . . . .	50
4.13	Comparison of Radial Auto-correlations between Isotropic, Hexagonal and Cubic with composition . . . . .	51

4.14 Comparison of Radial Auto-correlations of Hexagonal Microstructure with finer compositions . . . . .	52
4.15 Comparison of Mean Precipitate Size of composition 0.3 and 0.4 . . .	53
4.16 Comparing total precipitate area for the the compositions 0.3 and 0.4	54
4.17 Directional Growth( $t_1 < t_2 < t_3 < t_4$ ) . . . . .	55
4.18 Precipitate parameter visualization . . . . .	56
4.19 Coalescence: $t_1 < t_2 < t_3 < t_4$ . . . . .	57
4.20 Coalescence: Identification and Tracking . . . . .	58
4.21 Rapidly evolving precipitate to study elongation: $t_1 < t_2 < t_3 < t_4$ . .	59
4.22 Inclination is easily observable in the tracked precipitates . . . . .	60
4.23 Plotting $C_{11}$ in an evolving microstructure . . . . .	62
4.24 Comparing $C_{11}$ change with composition . . . . .	63
4.25 Comparing $C_{11}$ change with anisotropy . . . . .	64
4.26 Cross-correlation matrix of a cubic microstructure . . . . .	65
4.27 Radial correlations along angles . . . . .	66
4.28 Final feature vector . . . . .	66
4.29 Comparing changes in feature vectors in different evolution regimes .	67

# List of Equations

2.2 Cahn-Hilliard Equation . . . . .	5
2.3 Allen-Cahn equation . . . . .	5
2.6 Yeung Torquato cost function . . . . .	9
3.1 2-Point Correlation . . . . .	19
3.1 Fourier Space representation of 2-Point correlation . . . . .	19
3.2 Level set normal vector . . . . .	23
3.2 Level set velocity . . . . .	23
3.2 Moments for binary images . . . . .	29
3.2 Covariance matrix to calculate inclination and major/minor axis . . . . .	29
3.2 Tracking of precipitate based on maximum likelihood . . . . .	31
3.2 Convexity formulation . . . . .	33
3.2 Monte Carlo definition of convexity . . . . .	34

# Chapter 1

## Introduction

Deriving Process-Structure-Property relations for materials begins by extracting useful features from structures. This report aims to address the various techniques that can be used in the quantitative examination of microstructure images, and provide algorithms, workflows and conclusions based on their implementation. We also give a detailed overview of how our data is generated, and the numerical nuances and approximations involved in it. We test the algorithms presented on various computationally generated microstructures as a test case, and demonstrate the statistical parameters captured through plots, and compare these plots to the visual microstructural images.

Having a complete and robust description of microstructure statistics is important for two reasons. Firstly, it aids with the characterisation of microstructure images into classes, enabling easier identification of materials with desired properties. Secondly, properties can be predicted efficiently only with these robust features, that is, the improvement in this feature selection directly impacts the study of material property.

Tools of statistics and computation can help us analyse microstructure images, and this report aims to discuss the formulation, implementation and use cases of these tools. The data sets of microstructures used in this report were computationally generated through the Cahn-Hilliard equation, and the formulation of microstructure generation is described. We also give a detailed overview of the extraction of properties such as elastic constants from the base microstructural image, and explain the analytical calculations involved in the extraction of this property.

We discuss the important statistical parameters derived and then demonstrate their applications on computationally generated microstructure images. We begin with spatial probability distribution of states and discuss in detail the derivation and implementation of the algorithm. How to interpret the probability distribution is

discussed and tested on microstructures with varying anisotropy and composition. Probability distributions are a key part of literature, and thus special emphases is placed upon it.

Methods to quantify interface velocity and clustering are introduced and their implementation details laid down in detail in the form of pseudo-code and workflows. Since these are predecessors to other parameters, their implementation details are important to save on manpower and computation time. Methodology to track precipitates based on the clustering algorithms is presented, with emphasis on algorithmic selection and workflow.

The ability to isolate an individual precipitate and track its evolution in time opens the door to the extraction of geometric information from precipitates. These geometric features are calculated and tested on evolving precipitates in this report. Furthermore, additional shape specific properties like convexity are also derived and calculated. Interaction between precipitates is also studied based on these geometric and statistical parameters.

The penultimate chapter titled 'Results and Discussion' provides examples where the parameters mentioned above are extracted. The results obtained by plotting these parameters are compared to the microstructure images to verify the results and infer how the changes in statistical parameters correlate with the visual changes in evolving microstructure images. We also include results from the elastic modulus calculation and demonstrate how certain properties and our features are correlated.

The overall aim of this report is to provide a comprehensive description of the statistical parameters, their methods and how they can be correlated to material property.

# Chapter 2

## Literature Survey

**Process-Structure-Property (PSP)** relations have been an emerging field of study in the last decade, with significant advances in the domains of material characterization. Structure-Property relations, in particular have been extensively studied to accelerate the process of high throughput Computational Material design [1]. The idea behind this process is that, once we have sufficient databases of microstructure or other material characteristics, the process of developing new materials can be made faster by using the links in the Structure-Property relations. That is, we can predict the properties of new materials just by an interpolation of existing feature data of already processed materials. Furthermore, these relations can also provide us important insights into why certain features seen in the materials lead to distinct properties.

Structure property relations are built upon the foundation of identifying the most important features from experimental tests on materials. For instance, in the microstructure itself, its phases and their spatial distribution often govern how they respond to external stress or stimuli, and a mapping from this spatial information to the property would fast-forward the selection of materials. The key to develop these relations is to have a robust method of extraction of useful features which capture most of the information associated with the material and has clear implications on the properties associated with it. In microstructures, the most common metric to judge the usefulness of these features is their ability to capture statistical and geometric spatial information, all the while being sufficient enough to enable reconstruction of the original microstructure or ensembles of it. Ensembles here refer to two microstructures of the same material, which might differ due to difference in observation time or location. The reconstruction of the original microstructure or ensembles of it would collate all microstructures of a material into a single class, thereby making the material accessible and easier to identify [2][1]. Reconstruction methods are also

of importance, and efficiency of these methods is an improving field of study.

Once the features have been extracted, building structure-property relations involves training models on available experimental or simulated property data. A wide variety of methods to build these models have been suggested, and these have only picked up in number with the emergence of Machine Learning and their improved performance on classification and regression tasks [3]. The applications of these models are diverse, and shall be elaborated in further sections.

The next section introduces Phase Field Modeling, which is used to generate the data/microstructures used for our analysis. We then examine how we can extract the elastic modulus from our microstructures computationally, which becomes our data related to material property. The subsequent sections then discuss the type of features and the methods to extract them, followed by reconstruction methods for 2D microstructures. This will be followed by discussion on feature compression and building models for structure-property relations. Microstructure evolution is another topic touched upon, and the methods in image processing dissected. We also briefly touch upon the recent push into Neural Networks and their implications.

## 2.1 Phase Field Modeling

Phase field modeling is an important method to study the evolution of a wide variety of microstructures. In such models, microstructures are generally expressed in the form of field variables such as composition, and the kinetics and evolution can be predicted by differential equations derived from thermodynamics. These models can be used to predicted the interface behaviour, and are traditionally broken down into sharp and diffuse interface models. Sharp interface models have a discontinuity in the field variables such as composition, and have an infinitely sharp interface. These are distinctly advantageous since they allow us to define and track the interfaces. On the other hand, diffuse interface models traditionally have a continuous interpolation of the field variables along the interface, for instance the non-linear change in composition between two phases.

Fick's law predicts the movement of constituents down the gradient, and thus is ineffective in explaining the presence and evolution of interfaces in traditional microstructures. Models were thus required to explain the uphill diffusion of constituents. To account for this uphill diffusion and effect of interfaces on the evolution of microstructure, a modified free energy functional was introduced.

$$G = N_v \int_V [g_0(c) + \kappa |\nabla c|^2] dV \quad (2.1)$$

Here,  $G$  corresponds to the free energy,  $c$  the composition,  $g_0(c)$  the bulk free energy density per atom (traditionally a double well potential function) and  $\kappa$  the constant which incorporates the interface effects into the free energy.

Substituting these into the modified Fick's law, we obtain the now ubiquitous Cahn-Hilliard equation [4] as follows, with  $M$  as the mobility tensor, with  $g_0''$  and  $\kappa$  as constants.

$$\frac{\partial c}{\partial t} = M[g_0'' \nabla^2 c - 2\kappa \nabla^4 c] \quad (2.2)$$

Microstructures can also be described using non-conserved order parameters, whose evolution is traditionally expressed through the Allen-Cahn equation [5], which incorporates the order parameter ( $\phi$ ),  $\kappa$  to account for the boundary and a relaxation parameter ( $L$ ) which controls the driving force for order parameter change. The Allen-Cahn equation is as follows, with  $f$  traditionally being a double well free energy potential.

$$\frac{\partial \phi}{\partial t} = -L \frac{\partial f}{\partial \phi} + 2L\kappa \nabla^2 \phi \quad (2.3)$$

Given the differential equations describing the order parameter change, they can be solved through spectral techniques involving the use of Fast Fourier Transforms [6]. FFT based techniques are particularly useful because microstructures are typically constructed with periodic boundary conditions, which makes transformation into the Fourier space trivial.

## 2.2 Homogenisation Theory

Homogenisation is the calculation of effective behaviour of material based on the behaviour of its local constituents and their spatial distribution. More specifically to materials, it could involve the calculation of effective stresses or strains from local strains and stresses. Fundamental tenets to solve the homogenisation problem for stresses and strain is to solve the equation of mechanical equilibrium, a key part of this thesis.

Traditional ways to solve the local value problem involve the use of the Finite Element Method [7], which involves discretizing the unit cell into elementary regions, and calculating the values of the displacement field in these regions. Since microstructures are often generated with periodic boundary conditions, the strains and stresses in a unit cell are also periodic, which means they can be split into effective and periodic strain and stress fields [8]. For instance, in a fixed loading problem, where  $E$  corresponds to the effective strain and  $u^*$  corresponds to the periodic strain field, the net displacement at a location  $r$  can be generalized as follows.

$$u = E.r + u^* \quad (2.4)$$

Here  $E$  also corresponds to the mean strain tensor of the cell, that is:

$$E = \langle \{\epsilon_{ij}\} \rangle = \frac{1}{V} \int_{\Omega} \epsilon_{ij} d\Omega \quad (2.5)$$

The local stress field  $\sigma$  is periodic, and obeys the equation of mechanical equilibrium at each point on the lattice and the boundary ( $div(\sigma) = 0$ ) [9]. Furthermore, the traction ( $\sigma.n$ ) on the opposite sides of the unit cells must be anti-periodic [10]. Finally, the net stress must be equal to the mean stress, which gives us the three conditions to solve the equation of mechanical equilibrium.

While dealing with periodic boundary conditions, it is effective to use Fast Fourier Transform(FFT) based techniques to solve for the displacements and effective strains. The iterative procedure involves repeatedly solving the equation of mechanical equilibrium and effective strain to obtain the optimal local displacement field.

## 2.3 Spatial Features

Spatial features or characterisation of microstructure must capture both geometric and statistical information. Geometric features generally consist of parameters related to shape and size of the local states. Local states can be defined as the phases of our microstructure at a point in space. Statistical features capture features like size distribution of the local states, probability of finding local states in space and alignment or anisotropy associated with the microstructure. In literature, the statistical parameters most frequently studied are the n-point correlation functions and linear path function.

### 2.3.1 N-point correlation function

The N-point correlation function is defined as finding 'n' specific local states in a local neighbourhood. In practice, the 2-point correlation function is much more ubiquitous. It can be defined as the probability of finding 2 specific local states at a spatial vector  $r$  from each other. In isotropic microstructures, since the spatial distribution is direction invariant, a more common representation is in terms of  $S_2^i(r)$ , where  $i$  represents the  $i^{th}$  local state,  $r$  represents the distance and  $S_2^i(r)$  is a function denoting the probability of finding 2  $i^{th}$  local states at a distance  $r$  from each other.

Calculating  $S_2^i(r)$  can be done most efficiently through Fast Fourier Transform methods [11]. Here, we can use the Convolution theorem in the Fourier space to simplify computation. Averaging across directions after calculating 2 point correlation presents us with the  $S_2^i(r)$ .  $S_2^i(r)$  can also be calculated through Supervised Learning[20] as well.

2 Point correlations are efficient since they capture significant information about the sizes of the local states and their distribution in space. They also include information about the volume fraction of the 2 phases or local states. Visualization of the 2-point correlation function can capture both the anisotropy and sizes of individual local states.  $S_2^i(r)$  can be considered as the 2-point correlation function averaged over all directions.  $S_2^i(r)$  can give us information about the average distribution of states, but loses out on parameters representing anisotropy.

The correlations are particularly effective because they provide information to enable reconstruction of the original microstructures from the derived correlation function. Furthermore, for piece-wise uniform structures with smooth boundaries, the original microstructure can be completely reconstructed from only the 2-point correlation function[12], subject to a translation in space. This phenomenon is further demonstrated in [2] for simulated microstructures. For ensembles of similar microstructure, the 2-point correlations show sufficient similarity to be considered as acquisition invariant features, and are extensively used in characterisation.

### 2.3.2 Linear Path Function

The Linear Path Functions captures the amount of clustering in a specific local state in the microstructures. It can be defined as the probability that a randomly oriented line segment of length  $r$  is completely encapsulated by a single phase[13], say phase 1. Since the size of a phase is limited, the Linear path function, ie.  $L_{11}(r)$ , tends to 0 as  $r$  increases.  $L_{11}(r)$  is typically calculated by using Monte Carlo simulations, for

both 2D and 3D microstructures.

The Linear Path function can also have the angle of inclination of the line segments  $\theta$  and  $\phi$  as a parameters for 3D microstructures, the function thus becomes  $L_{11}(r, \theta, \phi)$  and  $L_{22}(r, \theta, \phi)$  for the phases 1 and 2 respectively. Both these functions add information about anisotropy, and would converge to  $L_{11}(r)$  and  $L_{22}(r)$  for isotropic microstructures.

Calculation of these generally involves Monte Carlo simulations. In [14], a method is presented where  $L_{11}(r)$  can be calculated by moving a 1D array across an image, an counting the number of phase interfaces. Considering isotropic microstructures, this can be only done along the horizontal and vertical directions to calculate samples for  $L_{11}(r)$  probability.

### 2.3.3 Geometric and Further statistical Parameters

In addition to the probability based descriptors, geometric features can add another layer of information. Traditional features of such kind provide information along 3 domains-

1. Composition : Described by the volume fraction of the phases
2. Dispersion : By calculating distances between precipitate centres
3. Size and Geometry: Calculated my measuring precipitate area and inclination

Distributions of dispersion-length, size and geometry can be calculated, usually as normal distributions. Descriptions based on these parameters have also proved effective in characterization and reconstruction of the original microstructure [15]. Furthermore, reconstruction of 3D microstructures from 2D image acquisitions have also proved successful, where 3D size and geometric parameters can be calculated by geometric operations on the 2D features [16].

Since experimental data is often noisy, it requires digital pre-processing before being used to generate these features. This process generally takes place through binarization of the grey scale image after smoothing through Gaussian kernels. These binary images are mapped onto the original image to increase clarity and aid clustering. This is generally followed by identifying individual clusters and their centres to calculate the dispersion, size and geometry [15].

## 2.4 Reconstruction Methods

The ability to reconstruct the original microstructure from the features is a metric to judge the quality of the features. Reconstruction methods traditionally involve a few basic common steps:

1. Generate a sample reconstruction.
2. Calculate feature vector from generated sample reconstruction.
3. Compare sample feature vector with original image feature vector through a cost function.
4. Exchange pixels in the sample such that the cost function is reduced, and report the image with a sufficiently low cost.

Gradient based methods to calculate the pixel exchanges or modifications based on cost function do not work with such problems well [11]. The most common method followed in reconstruction is the Yeong and Torquato (YT) reconstruction method [17]. The YT Reconstruction method is studied further.

### 2.4.1 The Yeong and Torquato (YT) reconstruction method

The primary reason the YT method is used is because of the presence of multiple local minima in the cost function. The YT method, also commonly referred to as *simulated annealing* solves these issues. The YT method can be understood in the following steps [18]:

Let us define a cost function based on the original features  $\hat{f}_j(r)$  and the model features  $f_j(r)$  (from our generated model) as follows:

$$E = \sum_{j=1}^m \sum_{r=0}^l |\hat{f}_j(r) - f_j(r)|^2 \quad (2.6)$$

Here  $r$  represents the various length scales, while  $j$  iterates over the different statistical parameters(eg. 2-point correlation, Linear path function etc.)

Then, the YT method works as follows:

1. Create an noisy sample image with the same volume fraction as the original image

2. Calculate  $f_j(r)$  from sample image, calculate E based on this  $f_j(r)$  and  $\hat{f}_j(r)$
3. Swap pixels in image and recalculate E(call it  $E_{new}$ )
4. If  $E_{new} < E$ , accept the swap. If  $E_{new} > E$ , accept it with a probability of  $\exp(-\frac{E_{new}-E}{T(k)})$ , where  $T(k)$  is defined as the Temperature at iteration  $k$ .  $T$  is chosen such that the Bernoulli random variable of pixel exchange acceptance has a probability of 0.5 at  $k=0$ (ie. first iteration) [17].

The 4<sup>th</sup> step is crucial to escape the local minima, and converge to the global minima for the reconstruction. The YT method is computationally expensive, and other methods like sequential quadratic programming methods [11] or Markov Random Fields [19] have also been used for reconstruction.

## 2.5 Building structure-property relations

Once the characterisation is complete and the features hold sufficient information about the microstructure, models can be built to map structure to properties. Since the features are often large in number, it is often reduced in size by linear combination of features through methods like Principal Component Analysis(PCA).

The process can be defined as follows:

1. Generate ensemble of microstructures and derive selected features from each microstructure.
2. Represent condensed microstructure features as the components of the features onto the principal axis.

It has been observed that the first 3 principal components can account for over 80% of the variance in the features [20]. Once this reduced order feature set is available, it can be used for purposes like clustering or classification of microstructures based on distances between the components[6]. Clustering algorithms like K-means clustering can be utilized for this clustering. Furthermore, components can also be used to predict bulk properties like Young's modulus by using linear regression [20]. Thus, this reduced order representation can simplify and speed up training in the supervised learning processes. The one drawback with PCA is that the principal components are often large in size and do not signal any physical or intuitive meaning, making them a mathematical operation.

Process-structure relations can also be predicted on the basis of components of feature vectors. For instance given a training set of temperature, aging time and microstructure of a sample, a classification model has been shown to predict the process parameters from a test set [21]. Thus, the flexibility of using feature vectors with supervised learning techniques gives us many avenues to tackle in the Process-Structure-Property domain.

Recent advances in Deep learning with the advent of Convolutional Neural Networks [22] and GPUs have accelerated the utilization of deep non-linear classifiers. Deep learning has been utilized to find low dimensional features from image ensembles to complement the 2-point correlations functions [23]. Deep neural networks have also been used to relate microstructure to the susceptibility to damage initiation [24]. Deep learning could be tremendously useful in the modelling of structure property relationships as well as generation microstructure ensembles. The availability of large amounts of data is what makes neural networks particularly effective, and with the increasing focus on material database creation, it is likely to be crucial.

## 2.6 Microstructure Evolution

Aside from deriving structure property relations and reconstruction of the microstructure, features derived can be used to study the evolution of the microstructure in time. Tracking the features as a function of time gives us important information about the evolution process like rate of phase separation, precipitate growth and agglomeration. Furthermore, effect of process on structure can be quantified through these features and thus process fine-tuning or adjusting can be improved.

Tracking the growth of a particular particle can explain particle kinetics, as explained in [25]. It is observed that precipitate growth is rapid in the initial stages of growth due to the diffusion of solute from the matrix phase. It is in this region that the phase separation is rapid, and the growth plateaus henceforth. That is, at this point in time all neighbouring phase has been absorbed into the precipitate. Roosen et al.[25] also show that the precipitate approaches it's Wolff shape as evolution continues. Furthermore, the elongation and inclination of the precipitate has been observed to increase in time using precipitate tracking techniques.

Multi-particle statistics have also been studied wherein the average size of the precipitates is tracked. The steep ascent in the average size of the precipitates is observed as expected. Shrinking of smaller particles leading to disappearance and agglomeration can also be observed through precipitate tracking techniques [25].

Aside from growth tracking of precipitates, interface velocity has also been of interest specially in simulated microstructure. Since the microstructures are represented as composition values in discrete spatial locations represented by pixels in an image, the discontinuity and non-differentiable nature of images is a challenge in calculating velocity. The sudden drop in composition in the phase interface rule out computational methods of calculating velocity since they involve numerically calculating the gradients. Thus, to be able to find the gradient at values in the image, the image needs to be smoothed by functions. The most common method of smoothing is by using Gaussian kernels. Other methods are also suggested like using a smoothed sign function [26], which deals with 2 specific problems of Gaussian kernels. Firstly, the location shift of the interface due to Gaussian smoothing is corrected, and the curvatures are more appropriately defined by using the smoothed sign function. Other methods like diffusion smoothing are also applied as mechanisms to improve calculation of gradients .

Once the image is smooth, Level Set Methods can be applied to calculate parameters like velocity, inclination and curvature [27]. Level set methods are numerical computations which can be used to calculate the movement of the interface, based on gradients. For level set methods to be useful, the image needs to be transformed to its Signed Distance Function(SDF), where locations inside the interface have a value  $> 1$ , outside  $< 1$  and the interface needs to have a value of 0. Level Set methods can also be easily extended to calculate 3D interface properties [26] like mean curvature, Gaussian curvature and normal velocity.

Applications of particle tracking are diverse, and it is a commonly studied phenomenon in fields like biology and earth science. Applications in material sciences can be seen in study of sintering using phase transformation [28] and in the modelling of battery separators [29].

# Chapter 3

## Methodology and Formulation

This chapter broadly has 3 sections. We first explain our data and how the effective strain response for our microstructures is calculated computationally; then describe our process to extract the elastic modulus from the loading conditions, and finally formalize our characterisation methods to describe microstructures.

### 3.1 Image Data Description and Pre-processing

#### 3.1.1 Data Description

The data generated is in the form of .dat files, with each .dat file representing one microstructural image. Each pixel in the image represents the composition at a spatial location, ie. the pixel values ranged between 0 and 1. The pixels with values  $\geq 0.5$  are considered as precipitates in our analysis. We analyse images from three sets of Cahn-Hilliard isotropic, cubic and hexagonal microstructures, with the composition for the minor component (precipitate) set at 0.3, 0.4 and 0.5 for each respectively. Furthermore, stress along different directions was applied to them to record their elastic behaviour.

#### 3.1.2 Data Pre-processing

Statistical analysis of shape, size and distribution often requires the image pixel values to be more differential for gradient calculations, and binary for image shape and size quantification. Thus these data pre-processing steps are essential to improve precision of the algorithms henceforth mentioned.

The images are first made continuous by convolution of the image with a  $5 * 5$

Gaussian filter to remove shape discontinuities, and make the gradient(1<sup>st</sup> derivative) smooth. After smoothing, the microstructure was binarized by replacing pixel values  $\geq 0.5$  by 1 and  $< 0.5$  by 0. This binary microstructure is henceforth used as the image for study. Figure 3.1 shows the composition profiles of the images before and post binarization.

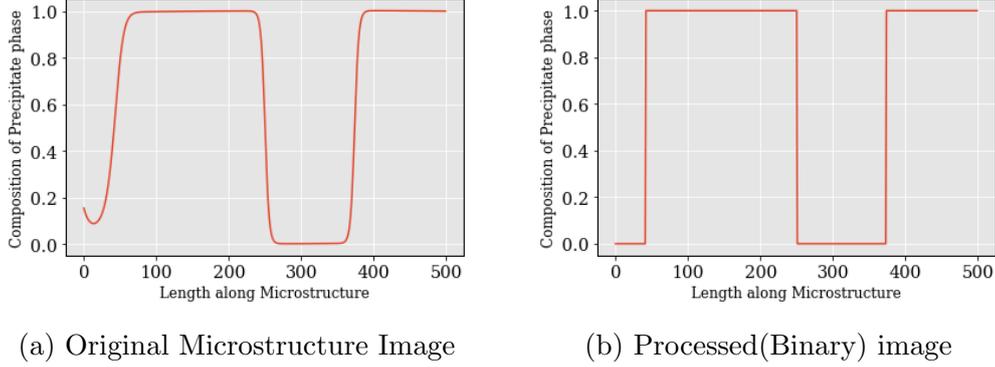


Figure 3.1: Visualizing Composition Profile: The Composition profile along a line visualized post processing.

## 3.2 Calculating homogenized strains

The raw composition microstructural data was used to record the response of the microstructure to applied stress. Let us consider a microstructure with a local composition  $c(r)$  in a domain of  $\Omega$ . The eigen strain is defined as the misfit strain due to the difference in lattice parameter, and is a function of composition defined as follows in our simulation [8].

$$\varepsilon_{ij}^0(c) = \beta(c)\varepsilon^T \delta_{ij} \quad (3.1)$$

Here  $\beta(c)$  is a chosen function of composition,  $\delta_{ij}$  the Kronecker delta function, and  $\varepsilon^T$  the strength of our eigen-strain. For our strain calculations, we assume  $\varepsilon^T$  to be 0 to remove the effect of eigen-strains from our strain calculation. We will include the term for eigen-strain calculation(which amounts to 0 in our simulation) henceforward to ensure continuity in the analytical expressions.

The total local strain as a function of displacement can be expressed as a 2<sup>nd</sup> rank tensor as follows:

$$\varepsilon_{ij} = \frac{1}{2} \left\{ \frac{\partial u_i}{\partial r_j} + \frac{\partial u_j}{\partial r_i} \right\} \quad (3.2)$$

Since our microstructures are diffused interfaces of matrix and precipitate, it becomes imperative to define the elastic modulus along the interface. Given the elastic modulus of the precipitate as  $C_{ijkl}^p$  and matrix as  $C_{ijkl}^m$ , we define the effective elastic modulus as a function of composition as follows.

$$C_{ijkl}(c) = C_{ijkl}^{eff} + \alpha(c)\Delta C_{ijkl} \quad (3.3)$$

$$\Delta C_{ijkl} = C_{ijkl}^p - C_{ijkl}^m \quad (3.4)$$

Here,  $C_{ijkl}^{eff}$  is defined as the volumetric weighted mean of the matrix and precipitate.

Based on Hooke's law the total elastic stress (in our case the total stress as well since we exclude eigen-strains) will correspond to :

$$\sigma_{ij}^{el} = C_{ijkl}\varepsilon_{ij}^{el} \quad (3.5)$$

If we represent the periodic strain by  $\varepsilon^*$ , the net strain thus becomes

$$\varepsilon_{ij} = E_{ij} + \varepsilon_{ij}^* \quad (3.6)$$

Combining the above 2 equations, and since divergence of stress at every point is 0 according to the equation of mechanical equilibrium, we get

$$\frac{\partial}{\partial r_j} \{C_{ijkl}(E_{kl} + \varepsilon_{kl}^* - \varepsilon_{kl}^0)\} = 0 \quad (3.7)$$

The equation above is constrained by the effective strain and compliance matrix through

$$E_{ij} = S_{ijkl}(\sigma_{kl}^A + \langle\{\sigma_{kl}^0\}\rangle - \langle\{\sigma_{kl}^*\}\rangle) \quad (3.8)$$

where  $\sigma_{kl}^A$  is the applied stress on the system, and  $S_{ijkl}$  is  $\langle\{C_{ijkl}\}\rangle^{-1}$ .

Substituting for the values of the periodic displacement  $u_l^*(r)$  for the periodic strain, the final equation of mechanical equilibrium thus becomes

$$\frac{\partial}{\partial r_j} \left\{ [C_{ijkl}^{eff} + \alpha(c)\Delta C_{ijkl}] \left( E_{kl} + \frac{\partial u_l^*(r)}{\partial r_k} - \varepsilon^T \delta_{kl} \beta(c) \right) \right\} = 0 \quad (3.9)$$

$$\left[ C_{ijkl}^{eff} \frac{\partial^2}{\partial r_j \partial r_k} + \Delta C_{ijkl} \frac{\partial}{\partial r_j} \left( \alpha(c) \frac{\partial}{\partial r_k} \right) \right] u_l^*(r) = C_{ijkl}^{eff} \varepsilon^T \delta_{kl} \frac{\partial \beta(c)}{\partial r_j} - \Delta C_{ijkl} E_{kl} \frac{\partial \alpha(c)}{\partial r_j} + \Delta C_{ijkl} \varepsilon^T \delta_{kl} \frac{\partial \alpha(c) \beta(c)}{\partial r_j} \quad (3.10)$$

This equation can be solved by using the Fast Fourier Transform Technique with the following iterative procedure.

To get an initial approximation of the periodic displacement, let us consider  $\Delta C_{ijkl} = 0$ , which gives us the Zeroth order displacement as follows

$$C_{ijkl}^{eff} \frac{\partial^2 u_l^*(r)}{\partial r_j \partial r_k} = C_{ijkl}^{eff} \varepsilon^T \delta_{kl} \frac{\partial \beta(c)}{\partial r_j} \quad (3.11)$$

Taking  $\sigma_{ij}^T = C_{ijkl}^{eff} \varepsilon^T \delta_{kl}$

$$C_{ijkl}^{eff} \frac{\partial^2 u_l^*(r)}{\partial r_j \partial r_k} = \sigma_{ij}^T \frac{\partial \beta(c)}{\partial r_j} \quad (3.12)$$

Taking the following equation in Fourier space gives us  $(u_l^*)^0_g = -JG_{il} \sigma_{ij}^T g_j \beta(c)_g$ , where the superscript on  $u_k^*$  as 0 implies the zeroth order approximation, with  $G_{il}^{-1}$  defined as  $C_{ijkl} g_j g_k$ . We use this initial approximation as the starting point for our calculations. The  $n^{th}$  order approximation from the  $n - 1^{th}$  is similarly calculated in [8] and [9] as the following

$$\{(u_l^*)^n\}_g = -JG_{il} \Lambda_{ij}^{n-1} g_j \quad (3.13)$$

where

$$\begin{aligned} \Lambda_{ij}^{n-1} = & \sigma_{ij}^T \{\beta(c)\}_g - \Delta C_{ijmn} E_{mn}^{n-1} \{\alpha(c)\}_g + \Delta C_{ijmn} \varepsilon^T \delta_{mn} \{\alpha[c(r)]\beta[c(r)]\}_g \\ & - \Delta C_{ijmn} \left\{ \alpha[c(r)] \frac{\partial (u_m^*)^{n-1}(r)}{\partial r_n} \right\}_g \end{aligned} \quad (3.14)$$

Using (3.8) and (3.13) in an iterative manner to estimate the value of  $E_{kl}$  and the displacements upto an error, we can determine the effective strain in a periodic 2-dimensional microstructure under an applied stress.

### 3.3 Calculation of elastic constants

We have thus far established a method to calculate the effective strain response of a binary periodic microstructure on an applied stress. We now use these loading conditions and effective strain pairs to estimate the effective elastic modulus ( $C_{ijkl}$ ) of these microstructure images.

The elastic modulus is a 4<sup>th</sup> rank tensor, which relates the stress and strain tensors through Hooke's law as

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl} \quad (3.15)$$

It is more common to represent this 4<sup>th</sup> rank tensor in the Voigt notation, which utilizes the symmetry in the crystal lattices to express it as a matrix. The Voigt notation for 3D lattices is represented as follows:

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{bmatrix} \quad (3.16)$$

Simplifying this further for 2-dimensional microstructures with tetragonal symmetry would give us the simplified stiffness matrix as follows

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{44} \end{bmatrix} \quad (3.17)$$

We use this stiffness matrix as the basis to estimate the values of  $C_{11}$ ,  $C_{22}$ ,  $C_{12}$  and  $C_{44}$ . This approximation simplifies and reduces the number of equations and loading conditions needed to solve for  $C_{ijkl}$ . The Hooke's law equations now simplify to the following:

$$\begin{aligned} \sigma_{11} &= C_{11}\epsilon_{11} + C_{12}\epsilon_{22} \\ \sigma_{22} &= C_{12}\epsilon_{11} + C_{22}\epsilon_{22} \\ \sigma_{12} &= 2C_{44}\epsilon_{12} \end{aligned}$$

By selecting our loading conditions (ie. selecting  $\sigma_{11}$ ,  $\sigma_{22}$  and  $\sigma_{12}$ ), we calculate  $\epsilon_{11}$ ,  $\epsilon_{22}$  and  $\epsilon_{12}$  and obtain 3 constituent equations. Notice that  $C_{44}$  operates independent of the the uniaxial stresses and strains, and thus can be obtained by a pure shear loading condition.

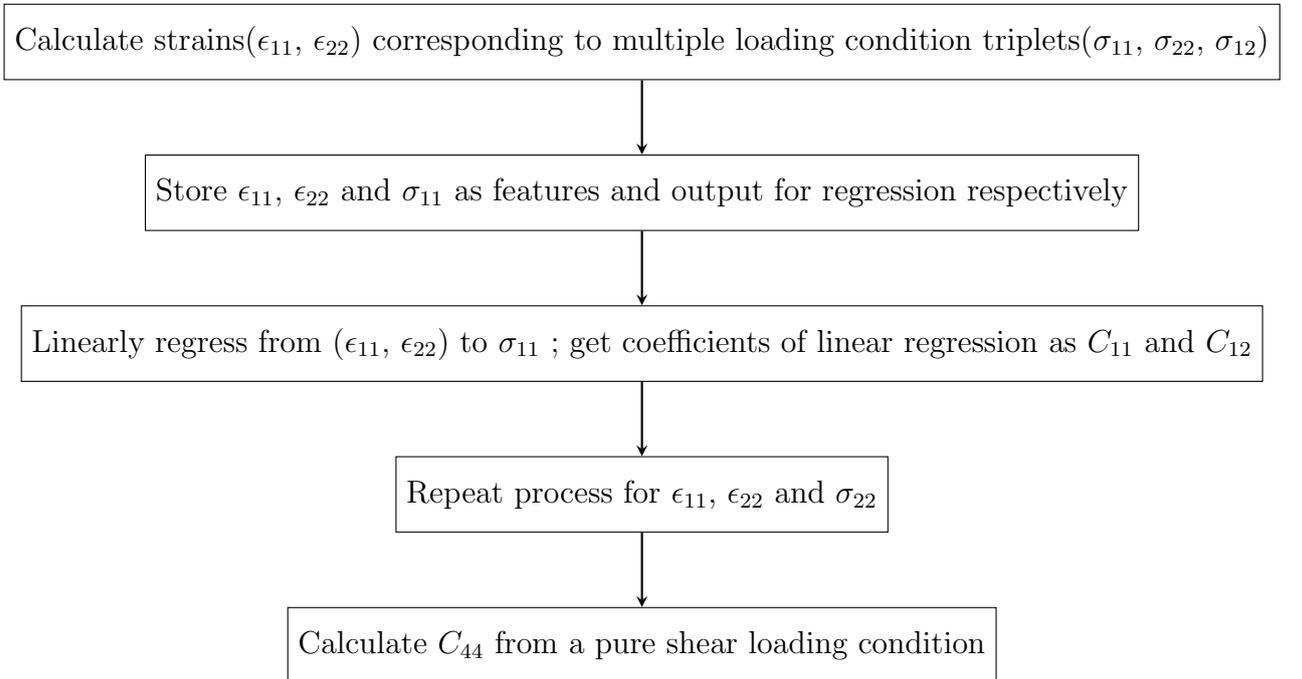
Estimating the values of  $C_{11}$  and  $C_{12}$  is equivalent to calculating the slopes of the best fit plane which fits the coordinates  $(\epsilon_{11}, \epsilon_{22}, \sigma_{11})$  for different loading conditions. Similarly, estimating the values of  $C_{12}$  and  $C_{22}$  is equivalent to calculating the slopes of the best fit plane which fits the coordinates  $(\epsilon_{11}, \epsilon_{22}, \sigma_{22})$  for different loading conditions.

Since there exist deviations from ideal Hooke's law or elastic behaviour due to our microstructure being a composite and the numerical disturbances in the calculations, we choose a best fit plane on multiple loading conditions as a better and more stable approximation of  $C_{ij}$  over a single or 2 loading conditions.

Thus, to find the best fit plane, we linearly regress from  $(\epsilon_{11}, \epsilon_{22})$  to  $\sigma_{11}$  and report the coefficients of regression as the values of  $C_{11}$  and  $C_{12}$ . Similarly, we regress from  $(\epsilon_{11}, \epsilon_{22})$  to  $\sigma_{22}$  to estimate  $C_{12}$  and  $C_{22}$ . The equivalence in the value of  $C_{12}$  can be used as a check to verify the calculations of the stiffness matrix.

Thus, by using multiple loading conditions we can estimate the stiffness matrix of our microstructure. In perfectly cubic microstructure,  $C_{11}$  and  $C_{22}$  are identical, which was used as the benchmark for our calculations.

The flowchart below demonstrates the steps in the estimation of the stiffness matrix  $C_{ij}$  for 2D 2-phase binary microstructures.



## 3.4 2 Point Statistics

### 3.4.1 Formulation

Quantifying the spatial distribution of states in a microstructure can help us capture the density of precipitate and define how isotropic or anisotropic a particular microstructure is. We calculate these distributions in terms of the microstructure probability function  $m_s^n$ , defined as the probability of finding the local state  $n$  at the spatial location  $s$ . In our binary microstructures, the  $m_s^n$  values are either 1 or 0, ie. a local state is either present or not present at a particular spatial location.

We extend this formulation to include 2 local states, and define  $f_t^{n_1 n_2}$  as the probability of finding local state  $n_1$  at a spatial location while simultaneously finding  $n_2$  at a spatial location  $t$  vector away from the original spatial location. This value can be calculated as follows [2]:

$$f_t^{n_1 n_2} = \frac{1}{S} \sum_{s=0}^{S-1} m_s^{n_1} m_{s+t}^{n_2} \quad (3.18)$$

We solve the above sum by taking the *LHS* and the *RHS* into the Fourier space and apply the Convolution theorem to the *RHS* expression. This gives us the Fourier transform of  $f_t^{n_1 n_2}$  as:

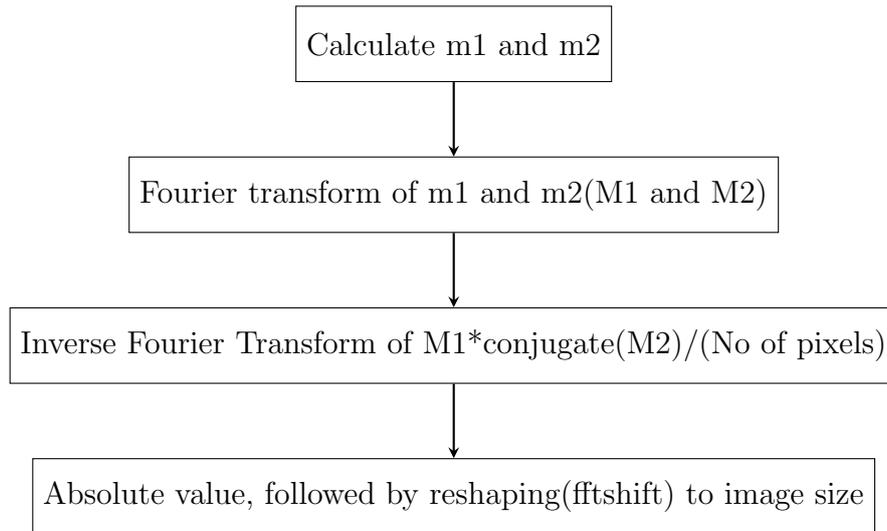
$$F_k^{n_1 n_2} = \mathfrak{F}\left(\sum_{s=0}^{S-1} m_s^{n_1} m_{s+t}^{n_2}\right) = \frac{1}{S} \sum_{s=0}^{S-1} m_s^{n_1} e^{-\frac{2\pi i s k}{S}} \sum_{z=0}^{S-1} m_z^{n_2} e^{\frac{2\pi i z k}{S}} \quad (3.19)$$

We then take the inverse Fourier transform of the above expression to calculate the 2 point correlations.

Here, if  $n_1$  and  $n_2$  are the same local state in our binary microstructures(1,1 or 0,0), the  $f_t^{n_1 n_2}$  values would represent the **auto-correlations** of the local states 1 and 0 respectively. Alternatively, if  $n_1$  and  $n_2$  are the different local states, ie. (1,0 or 0,1), the  $f_t^{n_1 n_2}$  values would correspond to the **cross-correlation** of the 2 local states.

### 3.4.2 Algorithm

We utilize the python library *numpy* to calculate the auto-correlation and cross-correlation values, since it supports swift calculation of Fourier and inverse-Fourier transform. The microstructure probability function  $m_s^{n_i}$  is calculated for each of the  $i^{th}$  local state, and stored as *numpy* arrays mi(m1, m2 etc.). The flowchart below summarizes the process followed in calculating the correlations.



### 3.4.3 Visualization

The probability correlation values are visualized by representing the probability values as colors of varying intensities, centred at the origin of the image. Since we take the Fourier and Inverse Fourier transform of the image, we use the *fftshift* command so that all the zero frequency values are closer to the centre. Since we are visualizing vectors at all directions, looking at them with the zero vector at the centre of the image aids comprehension. Thus, they are centered at the origin. Once the correlations are captured, we can calculate the radial probability distribution, as well as the probability distributions at various angular sectors. These visualizations are presented in the figures given below:

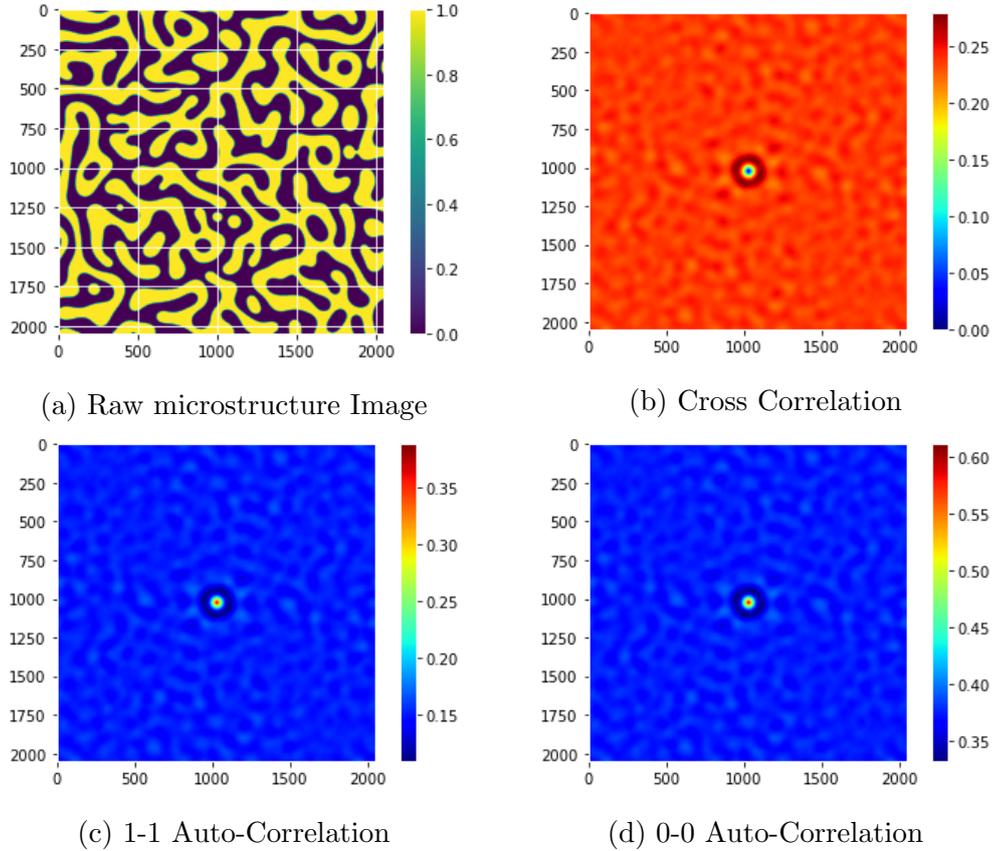


Figure 3.2: Visualizing Correlations

These visualizations and the probabilities were bench-marked against the *PYMKS python library* [30], where they gave identical results.

### 3.5 Level Set Methods and Velocity Formulation

Calculating velocity of interface is important since it gives us a good metric to judge pace of precipitate growth, and understand the inherent qualitative properties of the 2 phases. Furthermore, the statistical quantities we calculate can be validated further by investigating the actual velocity of the phase boundaries. We have used the Level Set Method [27] to quantify the orthogonal velocity at the interface of the 2 phases. Velocity studies are important because they let us visualize how a particular precipitate is growing, identify regions of rapid growth and distinguish between 2 similar looking microstructures.

### 3.5.1 Level Set Methods: Formulation

The problem statement can broadly be broken down into 3 parts, locating the interface, finding the normal to the interface to suggest direction of movement and finding the magnitude of the interface velocity. We study the 3 parts separately:

#### 3.5.1.1 Contouring or Interface Location

We define the interface pixels as the pixels which have at least one bordering (vertical, horizontal or diagonal) precipitate pixel. Thus for a single pixel precipitate, we would have 8 bordering interface pixels. The presence of diagonally neighbouring points do not harm our calculations, since we smooth the binary image, and the gradients formed post smoothing are nearly equal in orthogonal and diagonal neighbours. The figure below illustrates the contours found for a hexagonal microstructure. Because the contours are fine, the image has to be zoomed in to notice the contours as computer rendering makes the faint lines more transparent.

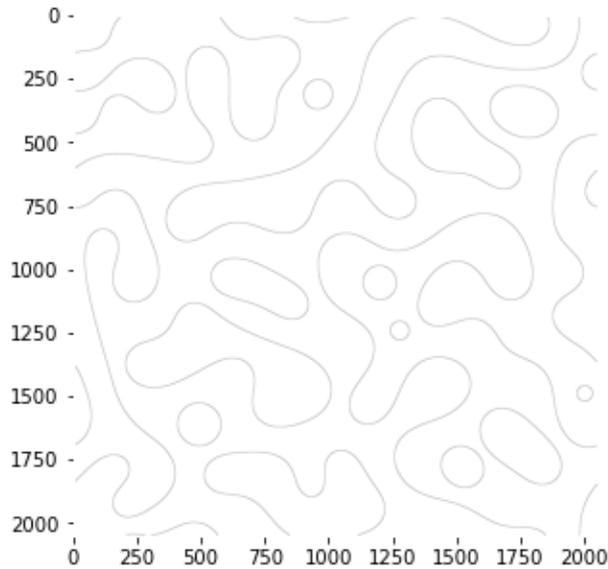


Figure 3.3: Contours on the microstructure

#### 3.5.1.2 Interface direction and velocity

The interfacial velocity and direction of a growing or shrinking precipitate can be captured by applying level set methods. This involves first converting the binary image into a continuous signed distance function( $\Phi$ ). We do this by first smoothing the

image by the convolution of a 5x5 Gaussian kernel on the image, and then subtracting 0.5 to make the interface pixels close to 0 to create the signed difference function( $\Phi$ ).

Once we have obtained the Signed Difference Function( $\Phi$ ), we can apply Level Set Algorithms to find the interface normal vector and the velocity. The normal vector  $\vec{N}$  and the velocity magnitude  $|\vec{V}|$ .

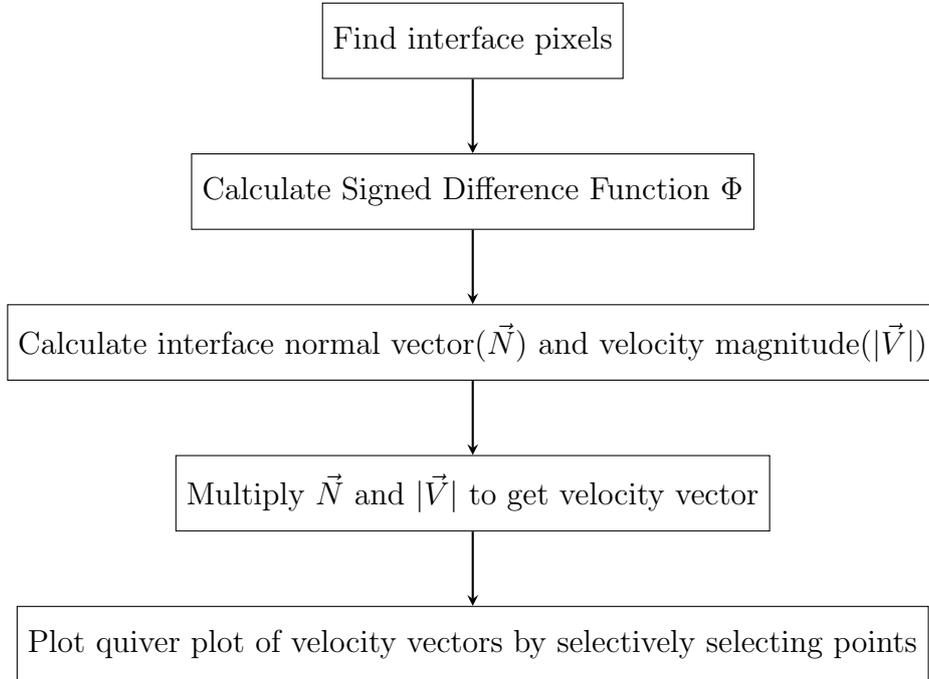
$$\vec{N} = -\frac{\nabla\Phi}{|\nabla\Phi|} \quad (3.20)$$

$$|\vec{V}| = \frac{\frac{d\Phi}{dt}}{|\nabla\Phi|} \quad (3.21)$$

Here, the sign of  $\frac{d\Phi}{dt}$  decides whether an interface is expanding or contracting. Furthermore,  $\frac{d\Phi}{dt}$  was calculated by interpolating a line through 4 points, to make the derivatives more representative of the actual curve.

### 3.5.2 Algorithm

We followed the following process while calculating the interface velocities of the precipitates in our microstructures.



### 3.5.3 Visualization

Since we are working with large size images, plotting the quiver plot by considering each interface pixel would inundate the figure with arrows. Furthermore, the nearby boundary pixels do not show significant deviation in velocity vectors. Thus we plot only every 8<sup>th</sup> velocity vector for better visualization.

The algorithm is tested on a continuously growing perfectly spherical microstructure, whose growth rate is calculated geometrically and validated.

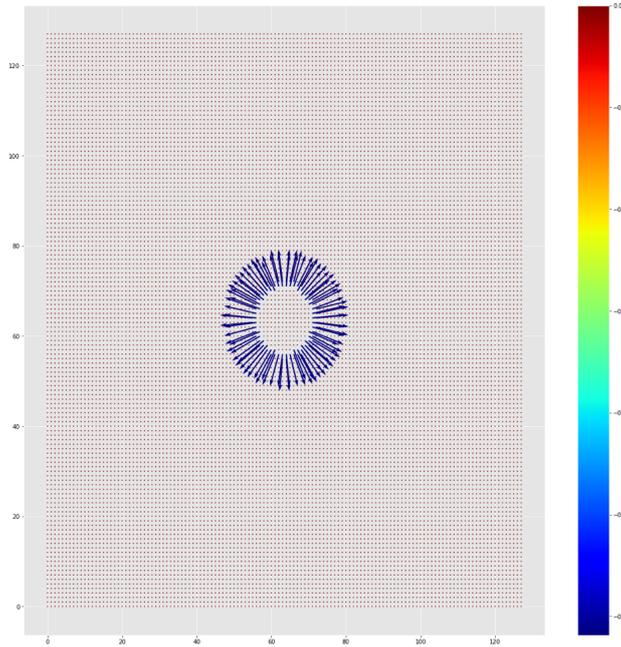


Figure 3.4: Velocity visualized on a quiver plot

## 3.6 Hoshen-Kopelman Algorithm and Application

### 3.6.1 Formulation: General Hoshen-Kopelman

The general Hoshen-Kopelman [31] is one of the most efficient algorithms for cluster counting and labelling since it takes only one pass through the image to label the clusters in the image. We take a sample binary microstructure image and apply the regular Hoshen-Kopelman algorithm, which does not account for periodic boundary conditions to test our algorithm for Non-periodic images. The general algorithm without Periodic Boundary Conditions(PBC) is implemented as follows [32], without considering the boundary pixels:

---

**Algorithm 1** Hoshen Kopleman algorithm for Non-PBC

---

**Result:** A matrix of labels

```
maxlabel=0 for  $x$  in 0:no of columns and  $y$  in 0:no of rows do  
  if occupied[ $x,y$ ] then  
    left = occupied[ $x-1,y$ ]  above = occupied[ $x,y-1$ ]  
    if  $left \neq 0$  and  $above == 0$  then  
      | label[ $x,y$ ] = find(left)  
    end  
    if  $left == 0$  and  $above \neq 0$  then  
      | label[ $x,y$ ] = find(above)  
    end  
    if  $left == 1$  and  $above == 1$  then  
      | union(left,above), label[ $x,y$ ] = find(left)  
    end  
  end  
end
```

---

Here, the union and find methods are derived from the common union-find algorithm. The union find algorithm is one of the most intensively studied and analyzed algorithms in the world, and the Weighted Union Find with Path Compression technique is implemented to optimize the efficiency of the algorithm. The technique is useful as it can shorten the trees which are formed in the union find algorithms and also account for complexity in the nodes which are formed.

The Hoshen-Kopelman algorithm works well for non-periodic boundary conditions, but requires modifications for periodic boundary conditions. The specific problem with periodic boundary conditions is that the edge pixels could belong to the same cluster, and thus the above algorithm fails.

### 3.6.2 Formulation : Hoshen Kopleman for PBC

To account for periodic boundary conditions, we follow the original Hoshen-Kopelman algorithm to completion. Post completion, we run the algorithm again only on the edge labels/pixels with two key changes:

1. We do not allot a new label value to a cluster encountered for the first time, instead we let it retain the earlier label
2. We introduce the periodic boundary condition to the 1<sup>st</sup> row and column of the

image by defining it's left and above boundary pixel by adding a imaginary row and column to the left and top of the image.

The algorithm for periodic boundary conditions is as follows:

---

**Algorithm 2** Hoshen Kopleman algorithm for PBC

---

**Result:** A matrix of labels

```

maxlabel=0 for  $x$  in 0:no of columns and  $y$  in 0:no of rows do
  if occupied[ $x,y$ ] then
    left = occupied[ $x-1,y$ ], above = occupied[ $x,y-1$ ]
    if left != 0 and above == 0 then
      | label[ $x,y$ ] = find(left)
    end
    if left == 0 and above != 0 then
      | label[ $x,y$ ] = find(above)
    end
    if left == 1 and above == 1 then
      | union(left,above), label[ $x,y$ ] = find(left)
    end
    if left == 0 and above == 0 then
      | maxlabel=maxlabel+1, label[ $x,y$ ] = maxlabel
    end
  end
end
for  $x$  in 0:no of columns and  $y$  in 0:n0 of rows do
  if occupied[ $x,y$ ] and edge[ $x,y$ ] then
    left = occupied[ $x-1,y$ ] (PBC), above = occupied[ $x,y-1$ ] (PBC)
    if left != 0 and above == 0 then
      | label[ $x,y$ ] = find(left)
    end
    if left == 0 and above != 0 then
      | label[ $x,y$ ] = find(above)
    end
    if left == 1 and above == 1 then
      | union(left,above), label[ $x,y$ ] = find(left)
    end
  end
end

```

---

### 3.6.3 Visualization

We visualize the cluster labelled microstructure by color coding each precipitate based on its number. We can also visualize a single precipitate by projecting only one label. In Figure 3.5, each precipitate is uniquely represented by a colour, whose numerical value can be found by using the colour-bar. The colour-bar also gives us information like the total number of precipitates, which in Figure 3.5 are around 225. This figure will henceforth be called as the Hoshen-Kopelman Labelled Microstructure(HSLM) image of a microstructure.

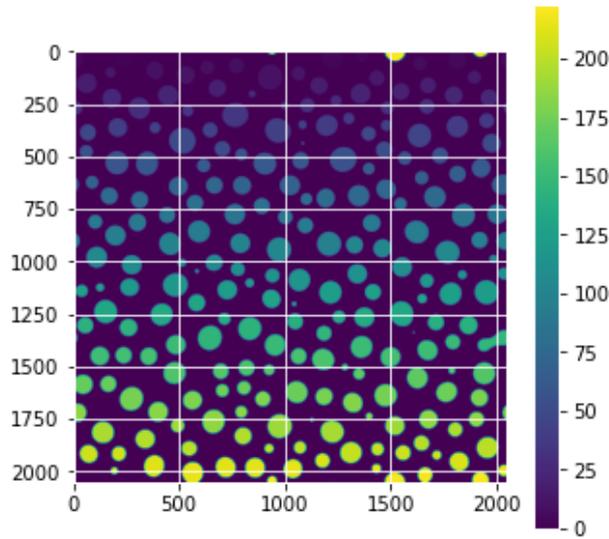


Figure 3.5: Cluster counting and labelling(HKLM Image)

## 3.7 Shape/size quantification and Precipitate Tracking

### 3.7.1 Shape/size quantification

Identifying features of precipitates like size, inclination etc can help us study precipitate growth and evolution in detail. Once we have identified a precipitate (by using the Hoshen-Kopelman algorithm), we can derive many features of the precipitate by analyzing the binary microstructure of the precipitate.

The two main proponents which are required to track and study a precipitate are the centre of gravity and the  $2^{nd}$  order moment of the binary shape. The centre of gravity is associated with the spatial position of the precipitate and the moment

is associated with the shape/inclination of the image. What makes it challenging is to extend the usually simplistic analysis on generic binary images to images with periodic boundary conditions. We will investigate the 2 problems separately in this section.

### 3.7.1.1 Centre of gravity

Centre of gravity for generic non-periodic binary images is directly obtained by orthogonal counting of precipitate pixels in both directions, and then finding the weighted mean of the resulting array. Since we are working with periodic boundary conditions, we first move from a linear array to an angular array, and do the weighted means in the rotational space [33]. The physical equivalent of our approach would be folding the image to calculate the COG. The algorithm for one direction, say  $x$ , is described as follows. Given a directional array  $x_i$  going from 1 to  $X$  and a weight distribution  $m_i$  which adds up to  $M$ :

$$\theta_i = \frac{2\pi x_i}{X}$$

$$\alpha_i = \cos \theta_i$$

$$\beta_i = \sin \theta_i$$

$$\bar{\alpha}_i = \frac{1}{M} \sum_i m_i \alpha_i$$

$$\bar{\beta}_i = \frac{1}{M} \sum_i m_i \beta_i$$

$$\bar{\theta} = \arctan(-\bar{\beta}_i, \bar{\alpha}_i) + \pi$$

$$X_{cog} = \frac{X\bar{\theta}}{2\pi} \tag{3.22}$$

Thus we can find the Centre of gravity along the two orthogonal directions separately to get the final COG of a binary precipitate.

### 3.7.1.2 Moments

Once we have calculated the values of  $X_{cog}$  and  $Y_{cog}$ , we can find the inclination, major and minor axis of a binary precipitate by using the values of second order moments. The  $p^{th}$  and  $q^{th}$  order moment of a binary image is calculated as follows:

$$M_{pq} = \sum_{i,j \in Obj} i^p j^q \quad (3.23)$$

We can thus calculate  $M_{20}$ ,  $M_{02}$ ,  $M_{11}$  and  $M_{00}$  from the above equation. We then calculate the following matrix and find the Eigen vectors and Eigen-values to get the inclination and major/minor axis.

$$\mu_{20} = \frac{M_{20}}{M_{00}} - \bar{x}^2, \mu_{02} = \frac{M_{02}}{M_{00}} - \bar{y}^2, \mu_{11} = \frac{M_{11}}{M_{00}} - \bar{x}\bar{y}$$

$$cov = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad (3.24)$$

We can find the angle of inclination and major/minor axis by finding the eigen-vectors of the above matrix.

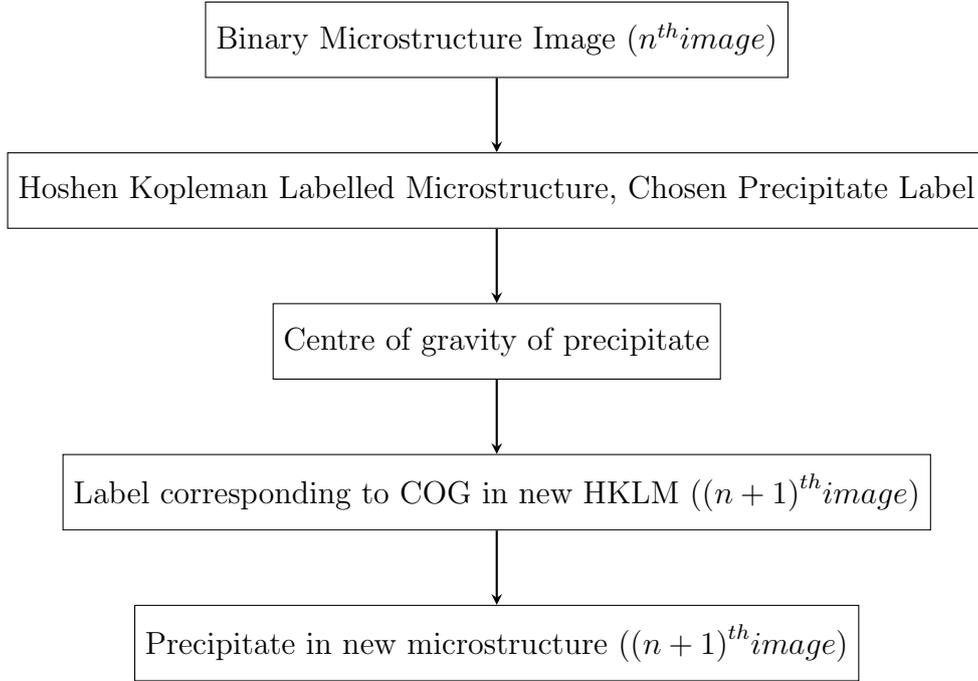
### 3.7.2 Precipitate tracking

Since our microstructures are evolving with time, it becomes imperative to track a particular precipitate and study its growth. The Hoshen-Kopelman algorithm does an excellent job at counting the clusters and labelling them from 1 to n. The major problem faced in tracking is identifying which precipitate we are currently tracking, as the Hoshen-Kopelman labels will change for each precipitate in each time frame of image acquisition. That is, the Hoshen-Kopelman Algorithm can label the precipitates, but cannot tell us that 2 different labels corresponding to 2 different images across time are actually the same precipitate, which has grown/shrunk in the evolution process.

The problem statement thus obtained is, given a **Hoshen-Kopelman Labelled Microstructure (HSLM)** image and the label of the precipitate being tracked, we need to find out the label for the same corresponding precipitate in the next/new HSLM image. We suggest 2 approaches to solve this problem:

### 3.7.2.1 Centre of Gravity(COG) based tracking

This approach relies on the fact that in gradually growing microstructure, the centre of gravity corresponding to a precipitate will lie on the same precipitate in the next time step of image acquisition. The flowchart below explains the steps in the algorithm.



Centre of mass based tracking techniques are most commonly used in image processing, but rely on the assumption that the COG will lie inside the precipitate. For concave microstructures this often fails. We thus suggest a better way to tackle in problem.

### 3.7.2.2 Probabilistic tracking

The primary issue with the COG based algorithms is the fact that as shape of precipitates move away from convexity, the Centre of Mass may or may not lie on the precipitate. Thus point based methods of tracking fail as the microstructure shapes grow more complex eg. Spinodal microstructure. Thus, to solve this problem, we take a probabilistic approach of identifying the most likely precipitate in the  $(n + 1)^{\text{th}}$  image from the chosen precipitate in the  $n^{\text{th}}$  image. Since we know that only a single precipitate in the  $(n + 1)^{\text{th}}$  image corresponds to the precipitate in the  $n^{\text{th}}$  image, we simply superimpose the precipitate of study onto the  $(n + 1)^{\text{th}}$  image and find the cluster with the most number of pixels matching. That is, we identify which cluster of precipitates is most likely to be the precipitate we are studying.

This method can be formalized by the following process. Given spatial location  $s$ ,

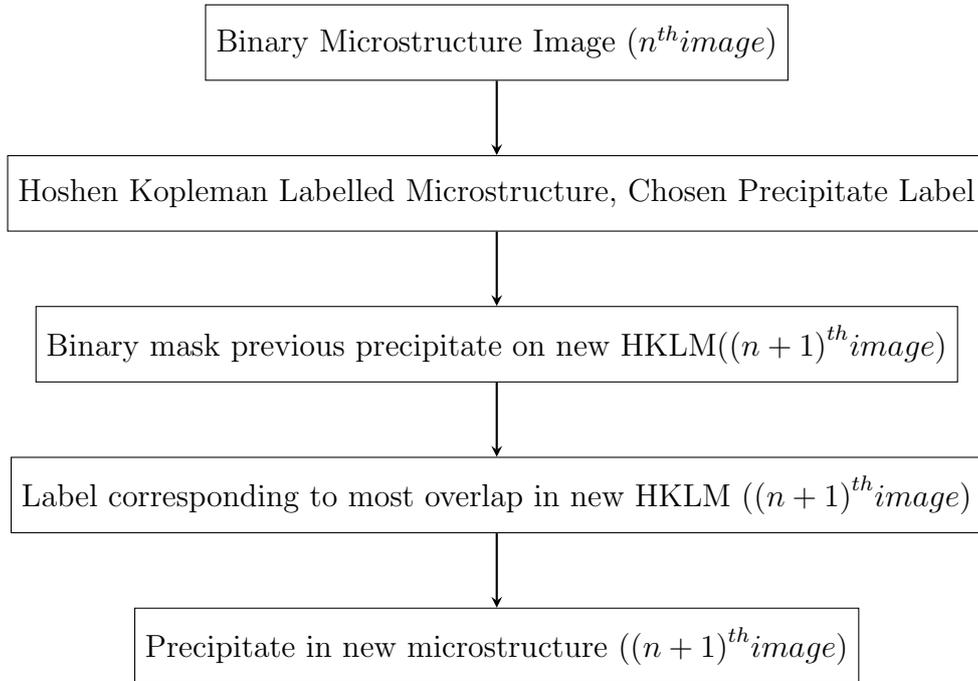
Let us define a unique function for each image called  $H_n(s)$ , which gives the Hoshen-Kopelman label for each spatial location of  $s$  for image number  $n$ .

Let  $K_n$  be the Hoshen Kopelman label associated with the precipitate in image  $n$  for all values of  $s$  which lie inside the precipitate.

Our goal is to find  $K_{n+1}$  for the  $(n + 1)^{th}$  image such that for all values for  $s$  with  $H_n(s) = K_n$ , probability of  $H_{n+1}(s) = K_{n+1}$  is maximized.

$$\max P(K_{n+1} | H_n(s) = K_n) \quad (3.25)$$

The flowchart defines the steps taken:



Once we have labelled the precipitate we are studying, we can calculate many quantitative parameters representing the precipitate distribution and shape/size. We list down parameters and features we can calculate:

1. Centre of gravity and it's movement
2. Precipitate size distribution
3. Interface velocity

4. Precipitate area change(growth/shrinking)
5. Agglomeration of precipitates
6. Sphericity and inclination
7. Major and minor axis
8. 2 point statistics of microstructure

### 3.7.3 Visualization

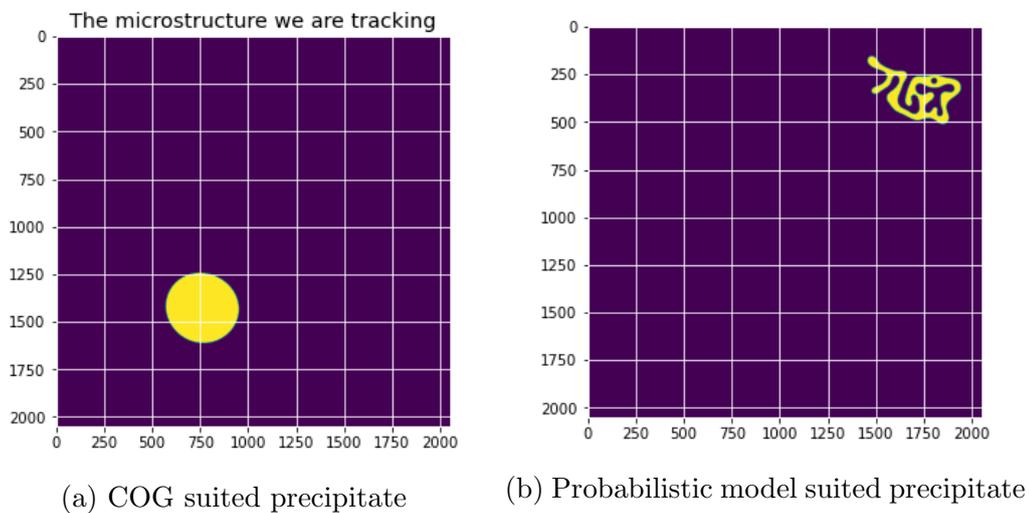


Figure 3.6: Visualizing Precipitates

## 3.8 Convexity measures

### 3.8.1 Definition

We define Convexity in terms of a number called Convexity number, a probabilistic measure of how convex a binary image is. The convexity number should have the following features for it to be a good indicator of probability [34]:

1. A number lies between  $(0,1]$ , with a perfectly convex shape having a value of 1
2. The number increases as the convexity of the image increases

A shape is perfectly convex if given any 2 points taken on the shape, the line joining the 2 points entirely lies in the shape. Analogous to this definition, we define our convexity to measure the likeliness of this line joining 2 points on the precipitate to lie on the precipitate. The convexity measure we calculate is the following:

*Given 2 points on the precipitate, what is the probability that the midpoint of these 2 points lies on the precipitate as well.*

We simplify to account only for the midpoint since the midpoint lying on the line is often a good indicator of other points on the line, and just calculating the midpoint saves us the computation time of calculating all the points in the line. The measure can thus be formally written as the following:

For points  $p_i$  in the binary image space  $I$  and  $m_{ij}$  being the midpoints of 2 points  $p_i$  and  $p_j$ , we calculate:

$$P_{convexity} = P(m_{ij} = 1 | p_i = 1 \cap p_j = 1) \quad (3.26)$$

This convexity measure is extremely difficult to compute, and theorems like the Convolution theorem we applied while calculating auto-correlations fail because of the presence of the extra midpoint term in the expression. We thus calculate this quantity by using Monte Carlo techniques.

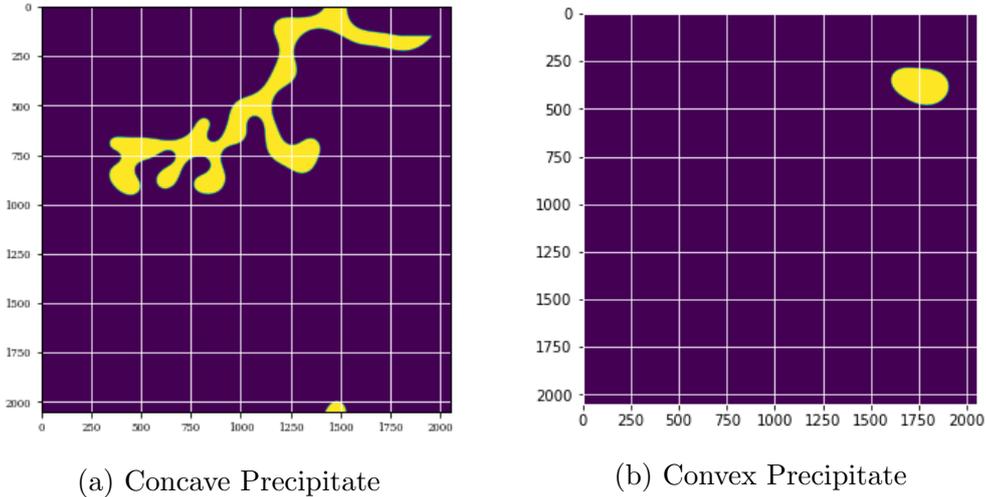


Figure 3.7: Visualizing Concave and Convex shapes

### 3.8.2 Monte Carlo method

Since calculating the exact probability measure is too computationally expensive, we use a Monte Carlo method to sample points from the image and calculate the probability measure. We run the simulation till the probability value converges, which we notice generally happens after 30,000 sampled points for 2000\*2000 resolution images.

The formulation of the method is simple. We randomly select 2 points on the image which lie on the object. We then find the point in the middle of the line joining these 2 points, and check whether this point also lies in the object. The net probability measure we calculate then becomes:

$$P_{convexity} = \frac{\text{No of midpoints on object}}{\text{Number of pairs of points chosen on object}} \quad (3.27)$$

### 3.8.3 Testing convexity on precipitate growth

We track 2 precipitates in the spinodal regime who can be observed to be moving towards convexity and test if our convexity measure is increasing in time and finally constant at 1 when the precipitate is perfectly convex. The 2 precipitates perfectly followed our hypothesis, which can be seen in the figures given below.

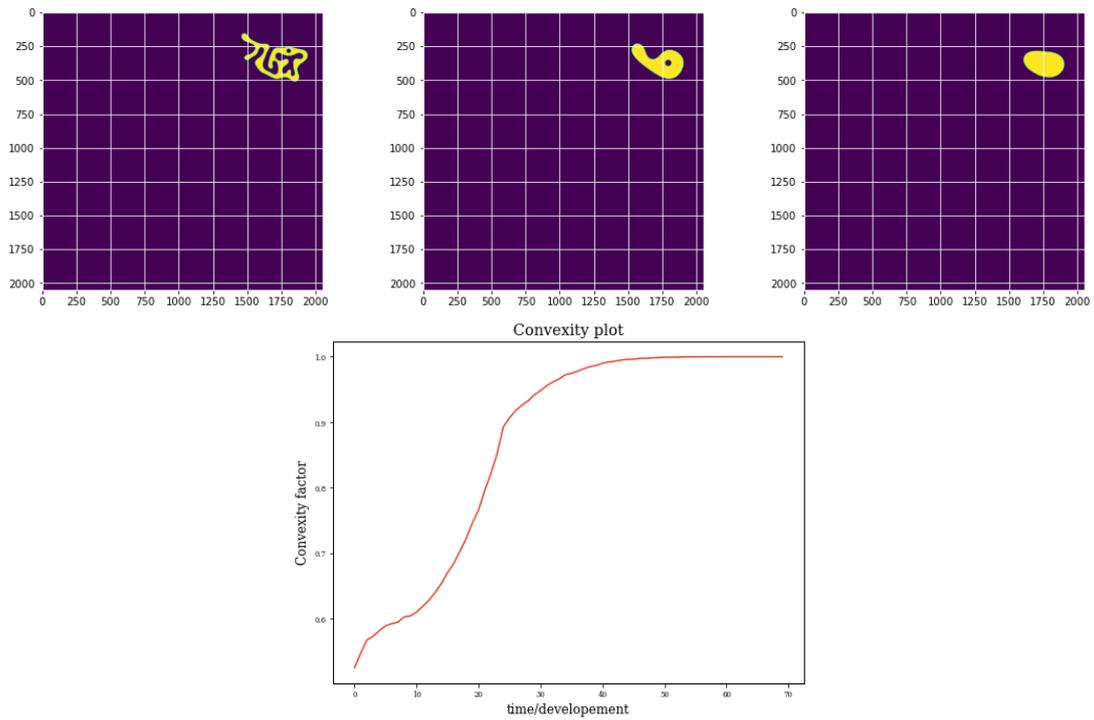
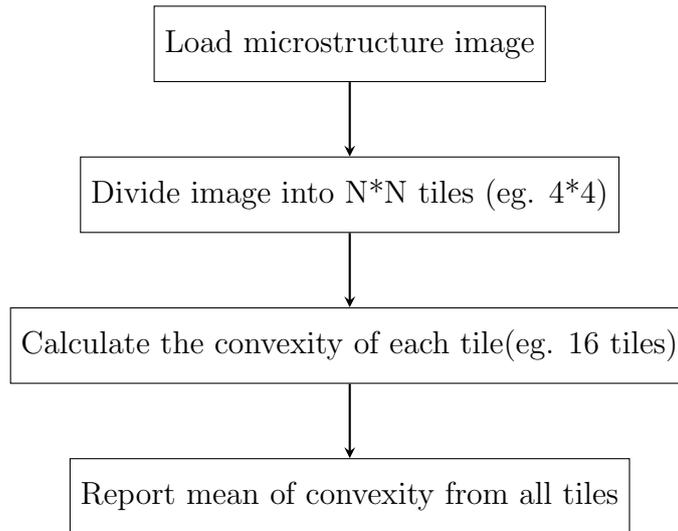


Figure 3.8: Visualizing Convexity Number with Precipitate Growth

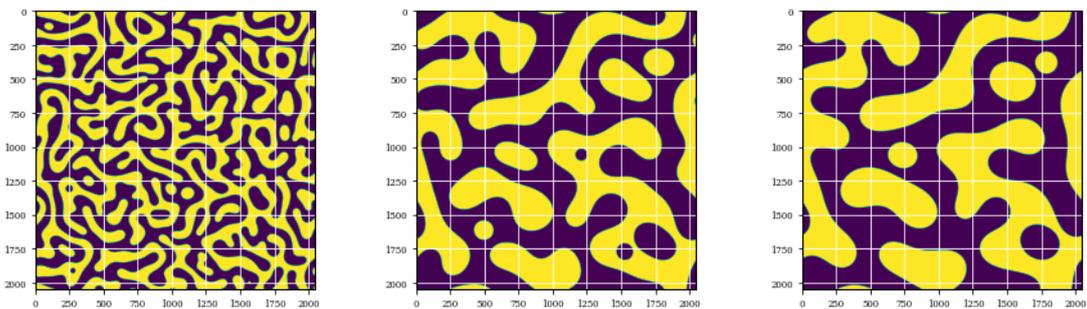
### 3.8.4 Convexity for complete microstructural images

Since up until now we have talked only about convexity of individual precipitates, it only makes sense to define convexity in a form to quantify complete microstructural evolution with multiple precipitates. The traditional form of convexity that we define in the earlier section fails when we apply it to more than 1 precipitate in an image, and thus we require a new transformed form of convexity.

We call this form of convexity **Short Range Averaged Convexity**. Rather than do the Monte Carlo random point sampling on the entire image, we break the image down into  $N \times N$  smaller images, calculate the convexity for each of these smaller images (called Short Range Convexity), and report the average convexity of these smaller bits. This convexity can be imagined as a measure to ascertain at what length scale the microstructure is locally convex in. The flowchart describing the steps followed is as follows:



The Short Range Average Convexity values of a developing microstructure showed a steadily increasing value when using a  $N=4$ , ie. using 16 tiles. The microstructural evolution and the convexity values are shown below. We can see the steady increase in convexity values clearly with evolution, which verifies our model perfectly.



(a) time =  $t_1$

(b) time =  $t_2$

(c) time =  $t_3$

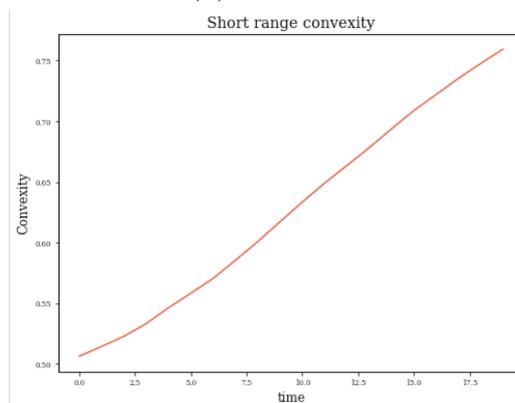


Figure 3.9: Convexity number change with Microstructure Evolution ( $t_3 > t_2 > t_1$ )

### 3.8.5 Calculating Short Range Averaged Convexity using different box sizes

Since our convexity measures for Short Range Averaged Convexity depends on the number of bits we divide the image into, it becomes imperative to statistically describe the significance of smaller or larger box sizes. Intuitively, smaller box sizes (breaking image into more parts) should lead to higher convexity values for any microstructure, since a microstructure generally tries to obtain perfect convexity along smaller regions first, and then through agglomeration or growth, becomes larger and convex on bigger length scales.

We do this experiment on Hexagonal microstructures of 3 compositions, 0.3, 0.4 and 0.5, and vary the number of parts we break the image into in the increasing power of 2. That is, we check the convexity values for 2x2, 4x4, 8x8 and 16x16 boxes. The results from the experiment can be seen in the figures given below.

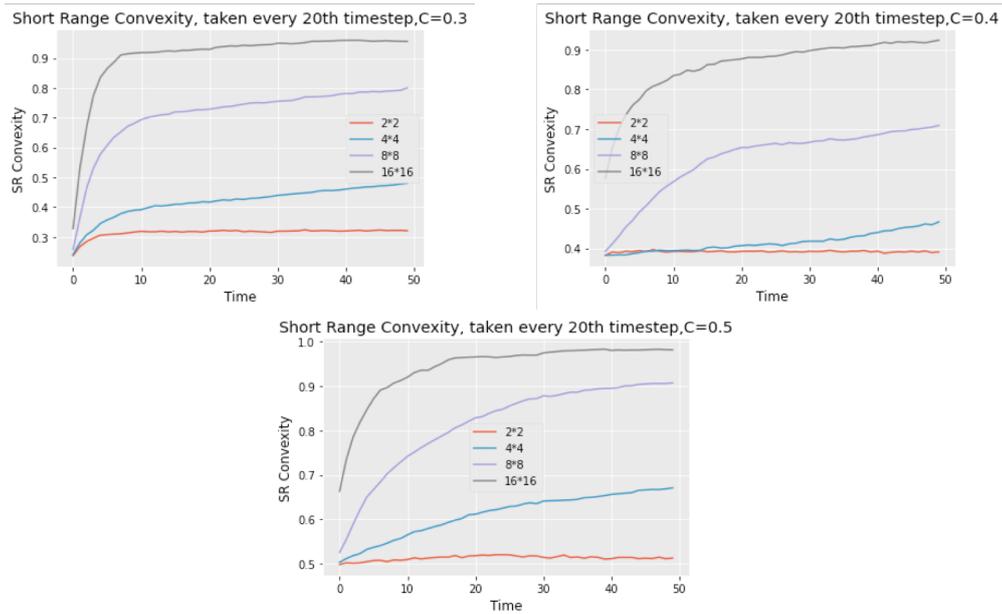


Figure 3.10: Convexity change with different image divisions

The plots above illustrate that the more divisions you make (and decrease the individual box sizes), more rapid is the Short Range Averaged Convexity rise. Furthermore, after certain time-steps of microstructural evolution, the higher-division convexity's converge to values close to one. This implies that microstructures develop in a way such that the convexity values increase and asymptotically converge to 1.

# Chapter 4

## Results and Discussion

This chapter is divided into 5 parts for easier understanding.

In the first section (Section 4.1) we use some of the tools and measures discussed in the previous chapter to analyse some sample microstructures and derive conclusions based on these results. The inferences derived from this section will clarify how the outputs from the algorithms in Chapter 3 need to be understood.

In the second section (Section 4.2), we compare the macroscopic properties of the Isotropic, Cubic and Hexagonal microstructures across time, and identify key distinguishing features between these microstructures. We use the conclusions drawn from Section 4.1 as basis for our comparison.

In the third section (Section 4.3), we test methods of precipitate tracking to identify key events and track precipitate growth. This section is meant to give the reader a sense of the wide range of evolutionary parameters that can be tracked by the clustering and tracking algorithms presented in this report.

In Section 4.4, we show the calculations of elastic modulus from our microstructures and comment on their accuracy. We also show how this property evolves with the microstructure evolution, and how it is influenced by composition and anisotropy.

Finally, Section 4.5 demonstrates how the properties calculated in Section 4.4 and the features calculated in Sections 4.1, 4.2 and 4.3 can be correlated to get an effective understanding of structure property relationships.

## 4.1 Characterisation examples and use cases

### 4.1.1 Datasets used for comparison

The Cahn-Hilliard generated microstructures are of three types, the perfectly isotropic, and ones with hexagonal and cubic anisotropy. Each of these were generated for 200,000 time-steps, and every 200<sup>th</sup> image was sampled out for analysis, ie. we had 1000 images to apply statistics to. Upon visual examination of these 1000 images, and noting down the statistical changes observed between different times, we observed that there was little significant change in statistical features between alternate images, ie. images 200 time-steps apart. Thus, to save on computing time we sample only every 30<sup>th</sup> image out of the 1000 images for feature extraction, which gives us enough time-series information to draw conclusions from, since within each 30 continuous images, the statistical difference is insignificant.

### 4.1.2 2-Point Correlations

The 2-Point correlations are calculated as explained in Chapter 3. We visualize the precipitate cross-correlation and auto-correlation heat map, and also calculate these metrics on a distance scale, ie. calculating the values on a radial scale by averaging the probability along a thin circular ring. Furthermore, we extend this analysis to capture angular features along particular angular sectors, to capture anisotropy. These 3 use cases are presented in the subsections ahead.

#### 4.1.2.1 Using 2-Point correlation to compare anisotropy

The auto-correlations represented as a probability heat-map is calculated for microstructures at different stages of their evolution. The points on the vector space (with the centre of the image as the origin) with color higher up on the color-bar represent higher probability of occurrence of the particular local state. For instance, at the centre (vector (0,0)), the probability will always be exactly equal to the probability of finding the local state (ie. 0.3 if we calculate auto-correlations for composition 0.3). Furthermore, we can generally observe rings of higher and lower probability close to the origin, which are good indicators of the periodicity in the microstructures.

For instance, in the below figures, you can observe the auto and cross correlations calculated for the isotropic case of precipitate phase composition 0.5, which demonstrate the properties mentioned in the previous paragraph. We can clearly observe that the auto-correlations and cross-correlations for the isotropic microstruc-

tures have no anisotropy, and are perfectly spherically contoured as shown below. The high probability spherical rings are also perfectly visible around the centre, showing the approximate judgement of precipitate sizes.

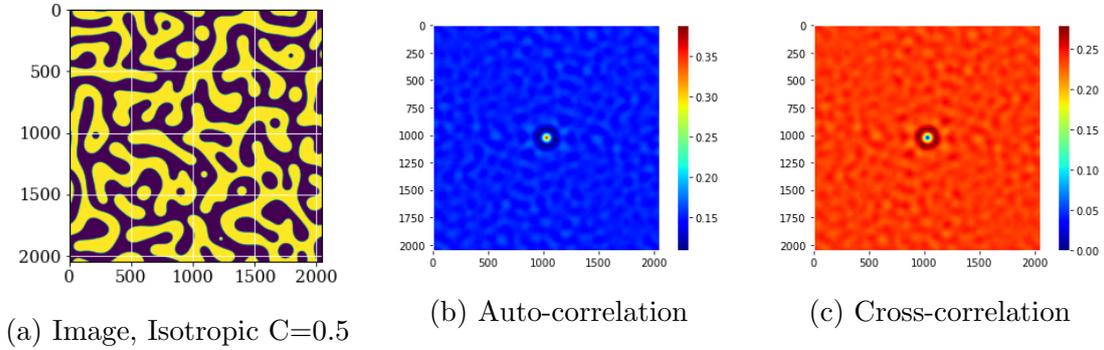


Figure 4.1: Visualizing Correlation for Isotropic Microstructures

For the hexagonal microstructure of composition of precipitate phase 0.5, visualizing the probability correlation heat-map gives expected results. For the hexagonal case, we can clearly see the anisotropy in the precipitate distribution in the figures below in the form of the hexagons of high and low probability around the centre, which adds credence to our formulation.

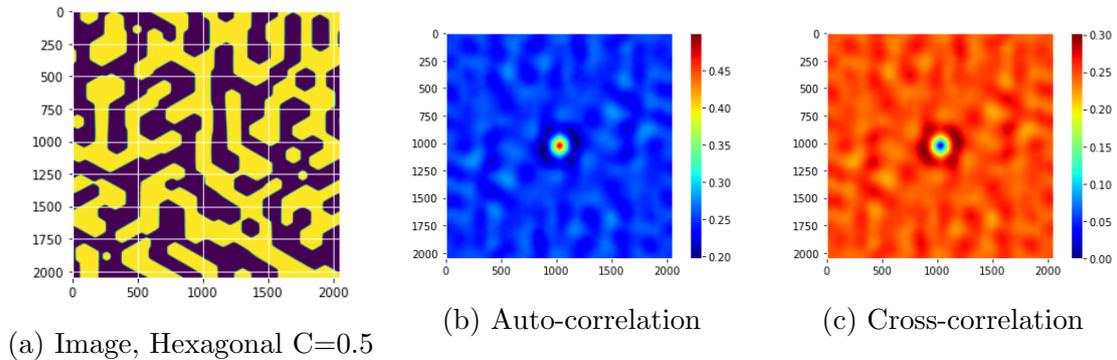


Figure 4.2: Visualizing Correlation for Hexagonal Microstructures

By using these correlation heatmaps, we can get a fair idea about the anisotropy shape/directions, and also judge the particle sizes by looking at the sized of the rings around the centre. Thus, these heatmaps capture important macroscopic features from the binary microstructure images.

#### 4.1.2.2 Using 2-Point Radial Correlations to study microstructure evolution

Although the correlation heatmaps give us significant information about the anisotropy and approximate sizes of the precipitates, these are not easily numerically quantifiable through heatmaps. Thus, we plot the radial correlations( $S_2(r)$ ) as a measure, derived directly from the probability heatmaps above, which can quantify the understanding of the particle sizes.

*We calculate the radial auto-correlation as the auto-correlation taken and averaged at a particular radial distance.*

This is calculated by simply taking a correlation plot, and finding the average probability at thin rings of radius  $r$  around the centre. This way, we can capture more easily the particle distribution at a radial distance  $r$  from the origin. This plot gives us important information about the probability distribution of precipitates, and can help us identify the particle size and inter particle distance of a microstructure.

The x-coordinate represents the radial distance from the origin, while the y axis represents the probability of finding the same local state(in our auto-correlations case - the precipitate) at that radial distance. The first x-coordinate minima of the radial auto-correlation curve corresponds to the precipitate size(ie. where the likelihood of finding the precipitate is lowest), while the first maxima after the origin is an indication of the inter-particle distance(ie. where the likelihood of finding another precipitate is highest again).

These 2 measures can be plotted across time to study precipitate size growth and distribution in microstructural evolution. As we can observe in the figures below for the Hexagonal Microstructures as an example, the plots are largely oscillatory, implying subsequent higher and lower probability regions of finding precipitate in a microstructure, which holds well with our microstructural image. Furthermore, as we make this plot at increasing intervals of time, the curve moves towards the rights, ie. the precipitate size and inter-particle distance both increase as the microstructure evolves, which also holds true with our images.

Additionally, the difference in probability values of precipitates at different compositions in the Figure 4.4 clearly shows us how composition changes affect particle sizes and distances. Here C03, C04 and C05 correspond to the compositions of 0.3, 0.4 and 0.5 respectively.

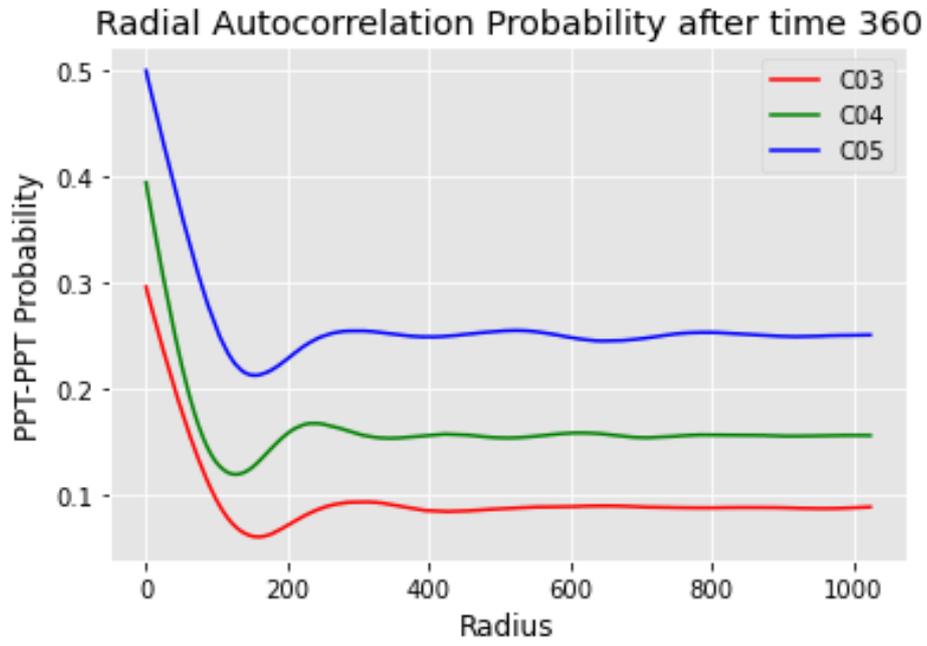


Figure 4.3: Plotting Radial Auto Correlations

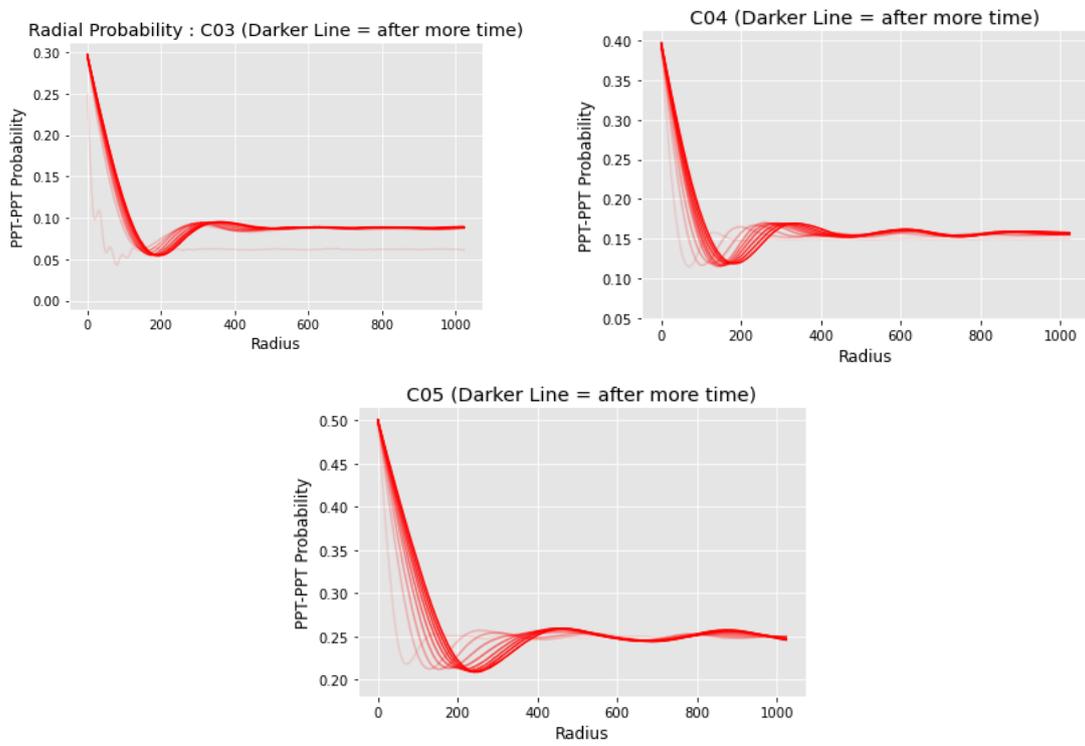


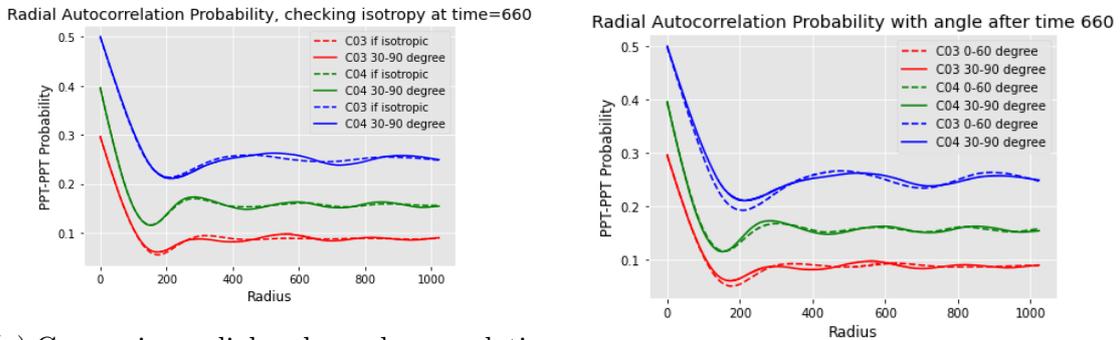
Figure 4.4: Plotting radial auto-correlations vs time

The radial auto-correlation change with time gives us a good indication of precipitate growth as we can clearly see the radial curve spread out/decrease in frequency as the microstructure evolves.

#### 4.1.2.3 Contrasting directions of anisotropy with correlations along directions

In cases of anisotropy, it becomes imperative to establish precipitate growth in different directions, as a single radial correlation does not give us information about the anisotropy in the system. This can be done simply by plotting the radial correlation only between certain angles ie. sectors to compare the evolution along these angles. That is, rather than calculating the mean probability over a ring around the origin, we calculate the mean along an angular arc on the ring.

This can be done in 2 ways, comparing growth/size along a particular direction to the average growth/size, or comparing distributions between directions of anisotropy. This itself becomes a test for anisotropy, as a perfectly isotropic microstructure will show identical radial behavior along all angles and directions for small radii in the neighbourhood.



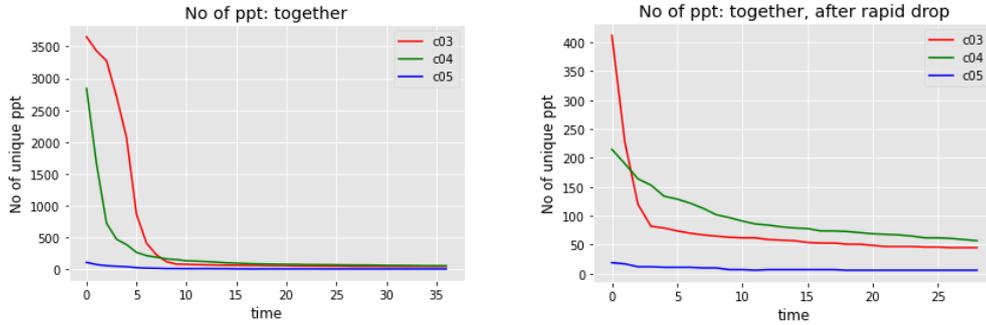
(a) Comparing radial and angular correlation as an anisotropy test

(b) Comparing 2 different angles of growth

Figure 4.5: Comparing directional growth

We can also compare angles of growth to understand in which directions the anisotropy is prominent. We compare 2 sectors of 60 degrees each in the curves above.

### 4.1.3 Using Clustering algorithms for precipitate counting and size estimation



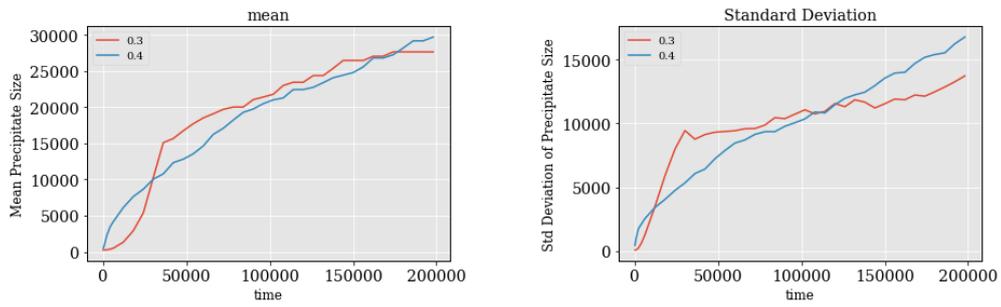
(a) Plotting number of particles

(b) Close up view of particle count

Figure 4.6: Particle Counting with time

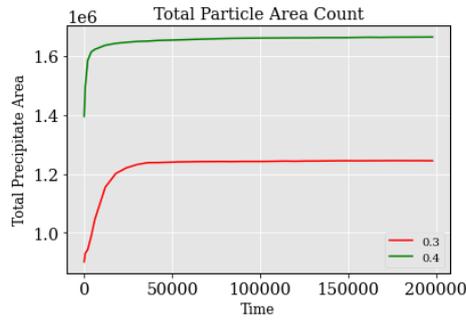
Since we have implemented the Hoshen-Kopelman algorithms to segment individual clusters of precipitates, we can also easily capture the macroscopic properties like the number of precipitates, average precipitate size and standard deviation across both composition and time. These features are more versatile in the initial time periods of microstructural development, and asymptotically converge as the microstructure grows to larger precipitates.

For instance, the figure below shows the mean, standard deviation and precipitate count comparison of Hexagonal microstructures of precipitate phase composition 0.3 and 0.4 across time. We also plot the total precipitate area in Figure 4.28, which is an indicator of how quickly complete phase separation takes place from the precipitate to the matrix phase(as indicated by the asymptote in the figure).



(a) Mean

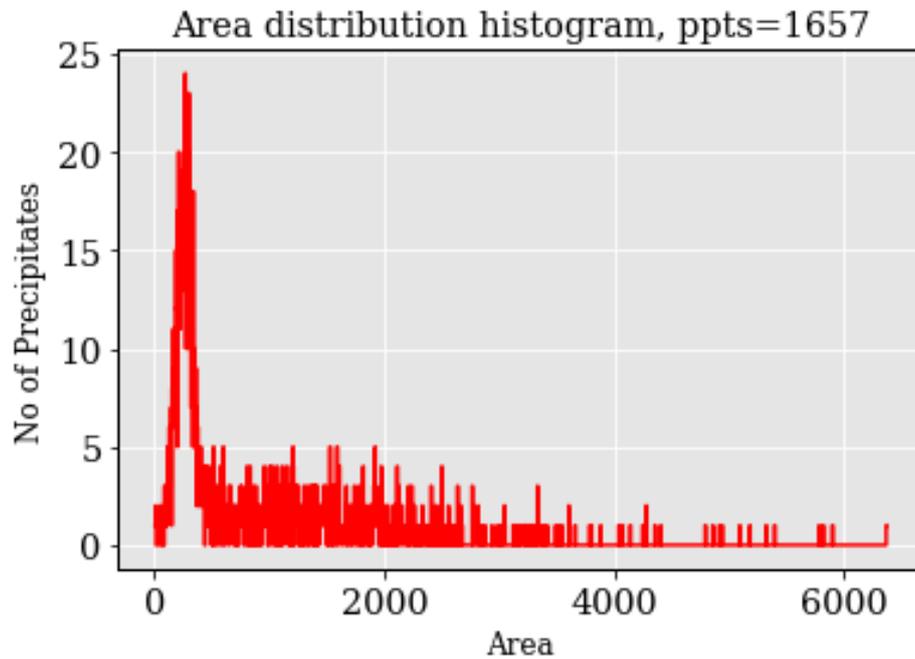
(b) Std. Deviation



(c) Total Precipitate Area

Figure 4.7: Precipitate Size statistics

Furthermore, we can also try and fit the precipitate areas into an area and log area distribution to better understand the precipitate size distribution by plotting histograms. The histograms are only useful with large number of precipitates in the image, and show prominence within the initial stages of the microstructure development. The typical histograms of really early stage microstructures are shown in the figures given below.

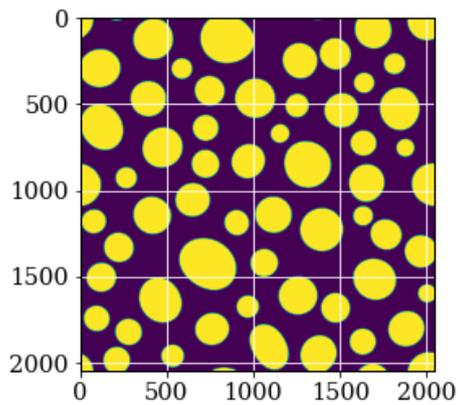


(a) Precipitate Size Distribution

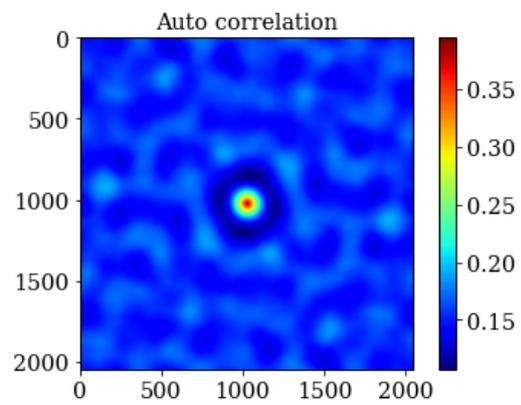
## 4.2 Comparison between Isotropic, Hexagonal and Cubic Microstructures

### 4.2.1 Auto-Correlation Heat-maps

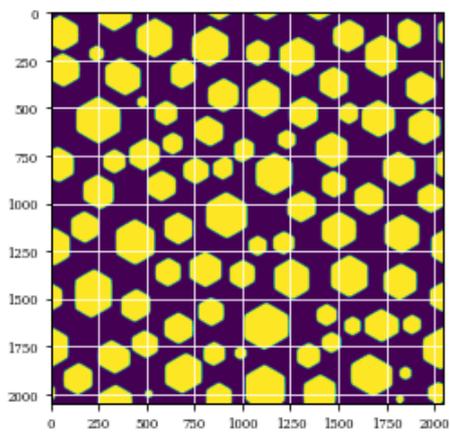
The Auto-Correlation heatmaps of the three microstructure types can give us clear information about the type of anisotropy and the precipitate size. The figure below illustrates this behaviour for all three types of microstructure (Isotropic, Hexagonal and Cubic). The image and the auto-correlation corresponding to that image are shown below.



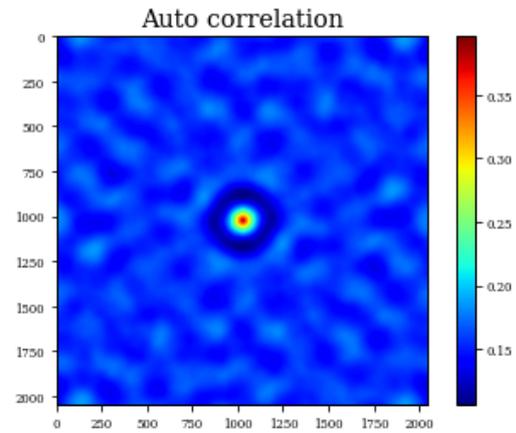
(a) Isotropic Image



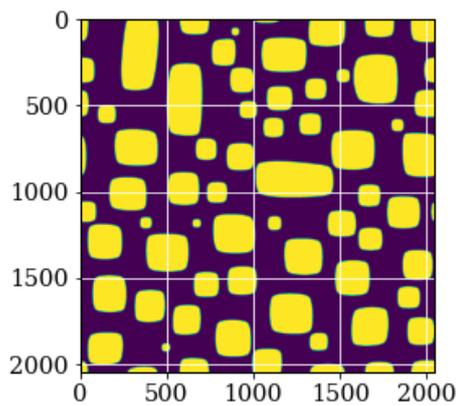
(b) Isotropic Auto-correlation



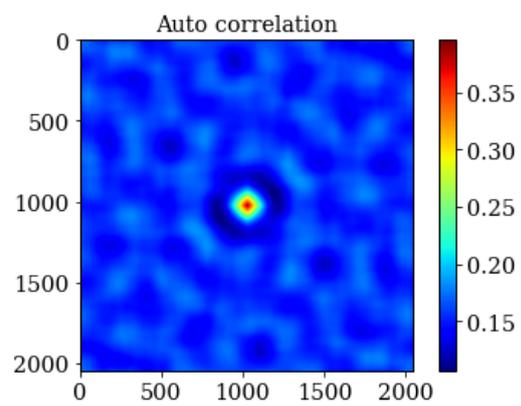
(c) Hexagonal Image



(d) Hexagonal Auto-correlation



(e) Cubic Image



(f) Cubic Auto-correlation

Figure 4.9: Comparison of Auto-correlation Heatmaps

The striking feature which is easily noticeable is the probability distribution along the shapes of the original image, that is, given the auto-correlations of an image, we

can easily identify the underlying shape anisotropy in the microstructure. The plots below are a zoomed in and high contrast image of Fig 4.9.

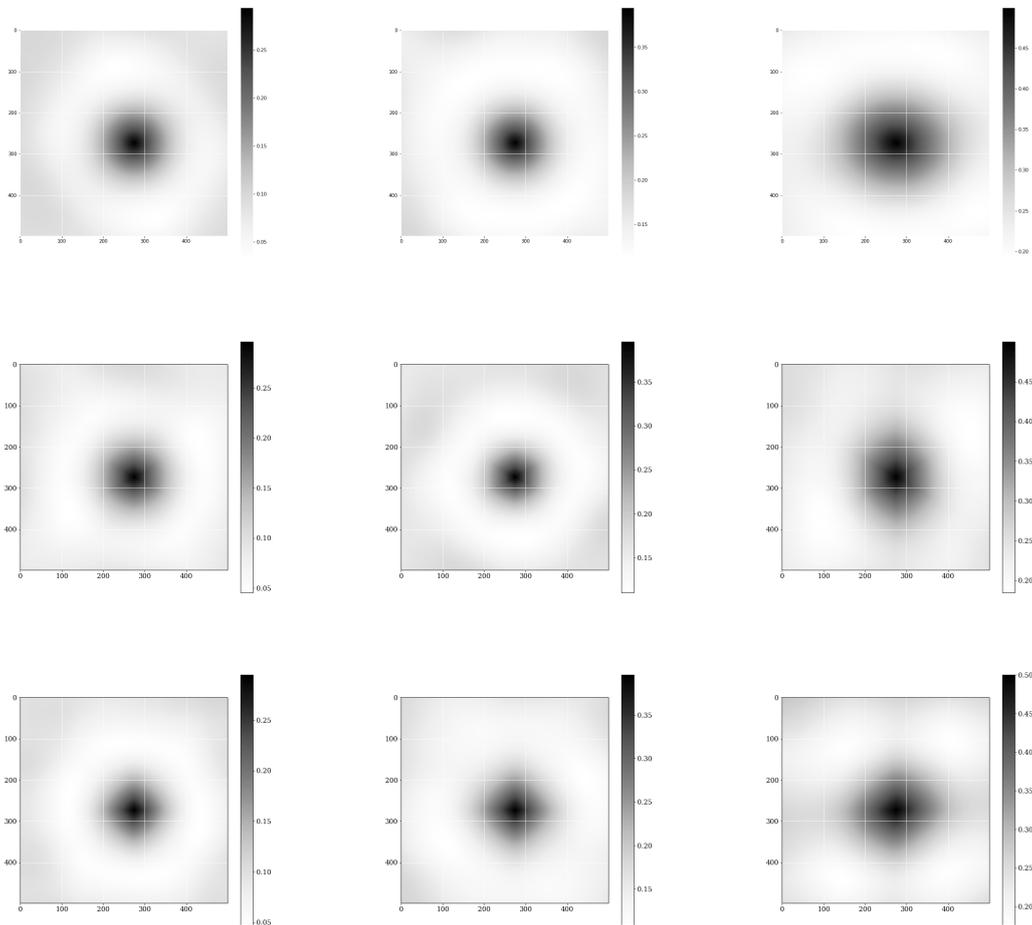


Figure 4.10: Left to Right: Increasing Composition, Top to bottom: Isotropic, Hexagonal and Cubic

We can also compare auto-correlations at different time of evolution. The figures below Fig 4.10 show how the auto-correlations vary for hexagonal microstructures of precipitate phase composition 0.4 during evolution.

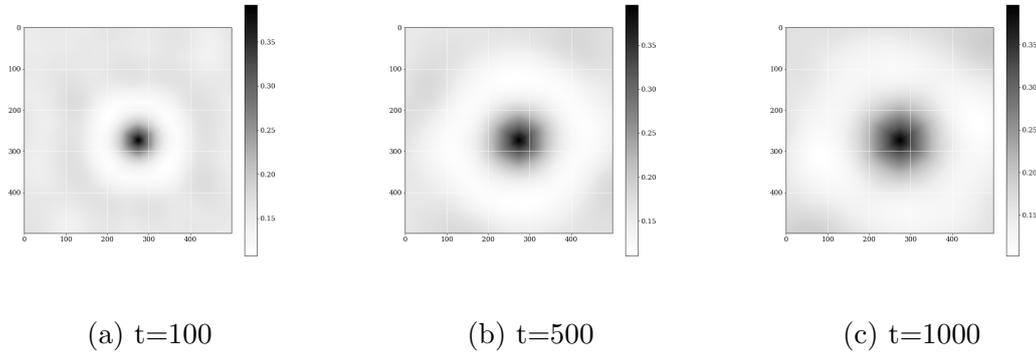


Figure 4.11: Auto-correlation variation with time for Hexagonal Microstructure with  $C=0.4$

It is fairly visible in Fig 4.11 that as time of evolution increases, the size of the precipitates as indicated by the size of the dark 'high probability' region also increases. The next Section 4.2.2 discusses the quantification of this region.

## 4.2.2 Radial Correlations

We calculate the Radial Auto-Correlations (as explained in Section 4.1.2.2 ) for the three microstructures, which give us easily quantifiable information about the precipitate sizes and distribution. We make the plots to compare how the radial auto-correlation changes based on composition and anisotropy.

The plots below compare the radial auto-correlations for the 3 types of microstructure(at a composition of 0.4) at 4 different time-steps, initial stage of development, middle phases and towards the end. We can see quite clearly that the radial distribution of different anisotropic microstructures vary significantly in the initial stages, where there are more precipitates, and gradually becomes similar as the microstructures evolve.

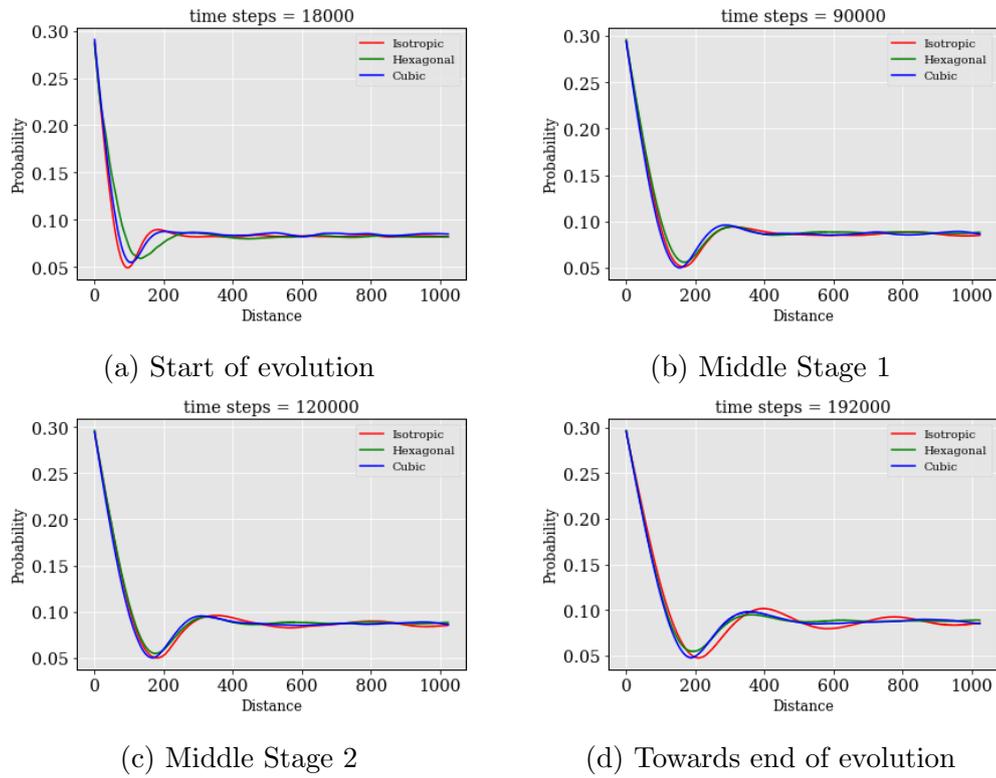


Figure 4.12: Comparison of Radial Auto-correlations with time between Isotropic, Hexagonal and Cubic for composition of 0.4

Furthermore, if we look at the 3 microstructure types separately, and plot the radial auto-correlations at different compositions, we can see how increasing or decreasing the composition affects the probability distribution of precipitates. The figure below illustrates that as the composition increases, the probability of finding the precipitates in space increases, while the particle size tends to also increase (as demonstrated by the shift of the first minima towards the right). This expected behaviour was followed in the Isotropic and Cubic case, while the Hexagonal microstructures showed some deviation from this behaviour, ie. the average precipitate size at similar time does not increase with increasing composition.

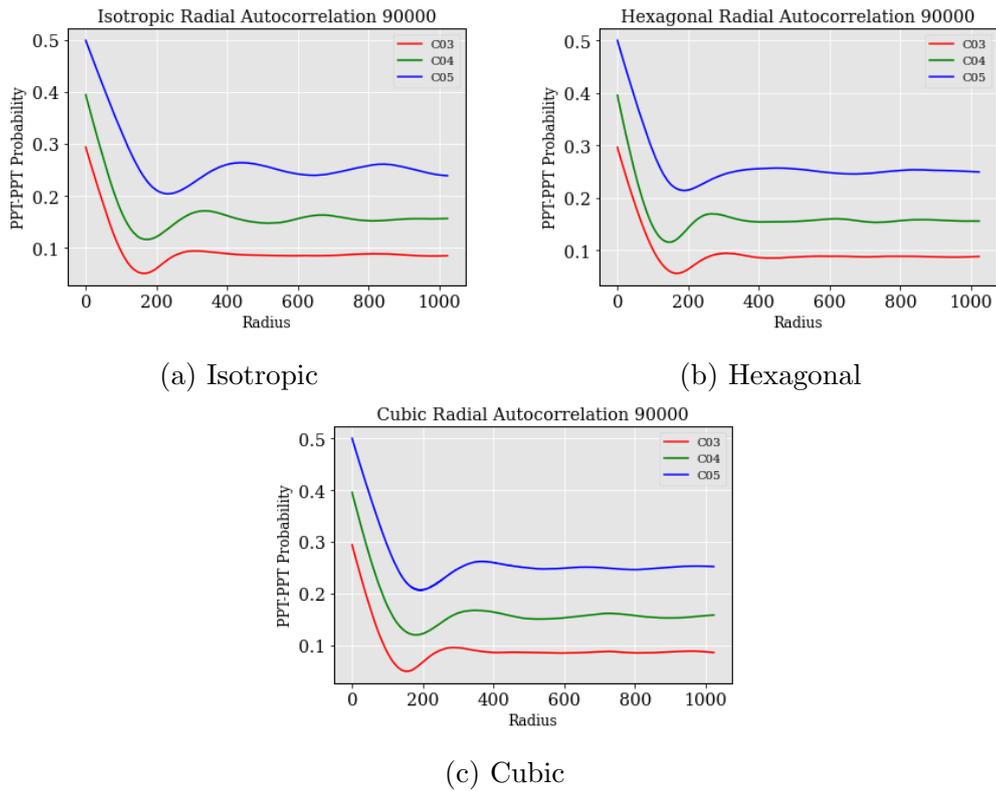


Figure 4.13: Comparison of Radial Auto-correlations between Isotropic, Hexagonal and Cubic with composition

Quite clearly, the Cubic and Isotropic microstructures show the expected behaviour between the compositions of 0.3 and 0.4, that is the probability increases and the first minima(indicator of precipitate size) moves to the right. The hexagonal microstructure shows the probability increase, but the minima at composition 0.4 is to the left of the minima at composition 0.3. This would indicate that the average precipitate size in the composition 0.4 is smaller than the size at composition 0.3. Thus, to study this more closely, we take the Hexagonal microstructures at further finer compositions of 0.3, 0.35, 0.4, 0.41, 0.42, 0.43, 0.45 and 0.5 to see at what composition ranges this behaviour is present. The plots below show the same.

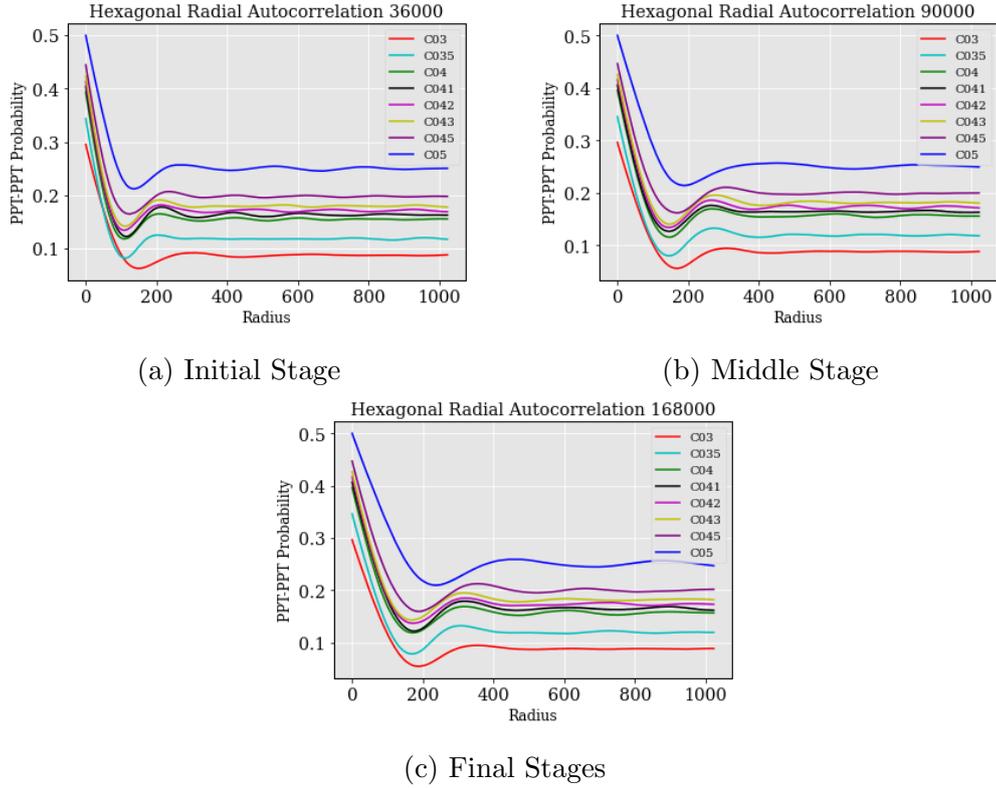


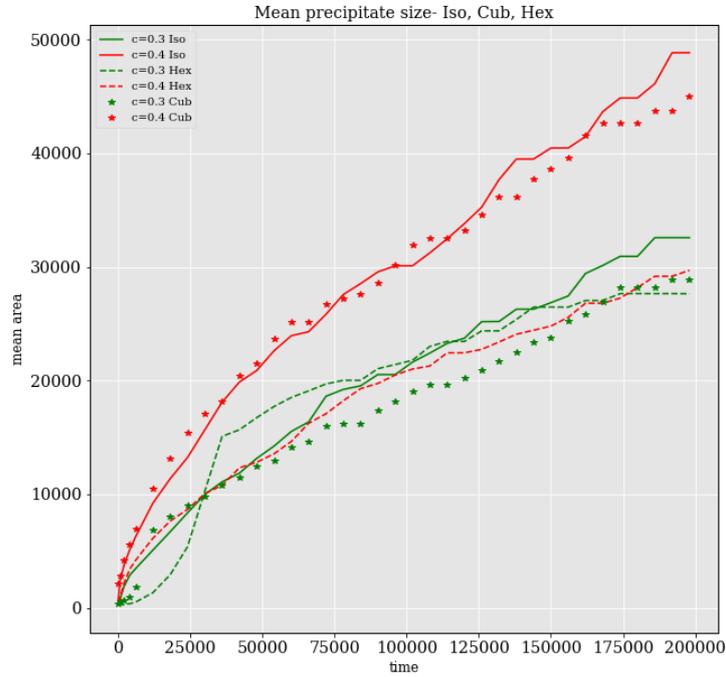
Figure 4.14: Comparison of Radial Auto-correlations of Hexagonal Microstructure with finer compositions

It's clear that the average precipitate size seems to be decreasing from Composition 0.3 to 0.42, and after 0.42 till the composition of 0.5, the estimate of precipitate size shows increments. Furthermore, if we look closely at the graph at the x coordinate of the first maxima after origin (an indicator of inter-precipitate distance), the inter-precipitate distance seems to have reduced too, which is also counter intuitive since we expect the inter-precipitate distance to increase as precipitates grow. One idea which could explain both these peculiar phenomena is that in the hexagonal case, due to the increase in the number of precipitates, the number of larger precipitates reduce to accommodate many smaller precipitates. That would explain the decrease in the precipitate size and the inter-precipitate distance. We will study and test this hypothesis in the next section(Section 4.2.3).

### 4.2.3 Precipitate Size Statistics

Let us first look at the mean particle area/size for the 3 Microstructure types(Isotropic, Hexagonal and Cubic) along different compositions. The plots below demonstrate this

statistical parameter at 2 different compositions of 0.3 and 0.4.



(a) Mean Precipitate Size with time

Figure 4.15: Comparison of Mean Precipitate Size of composition 0.3 and 0.4

What is really interesting in the plots above is that the mean precipitate size increases with composition for both cubic and isotropic microstructures, but not for hexagonal microstructures. The mean precipitate size remains nearly equal for both compositions (0.3 and 0.4) in the case of hexagonal microstructures.

So we can conclude that the mean precipitate size does not increase in the hexagonal case significantly between the composition ranges 0.3 to 0.45, but it does in the Isotropic and the Cubic case. This is primarily due to the increase in the number of smaller and similarly sized precipitates. Possible explanations of this phenomena could be due to the constraint in the directions of agglomeration of hexagonal microstructures due to their microscopy.

Finally, another important characteristic to study is the total separation of the precipitate generating phase, that is, to find whether the components are fully separated or not. This can be easily seen when we plot the total area of precipitates, and upon complete separability, this area should be almost constant, or asymptotically constant. Plotting this for the 3 microstructure types gives us the following asymptotic curves.

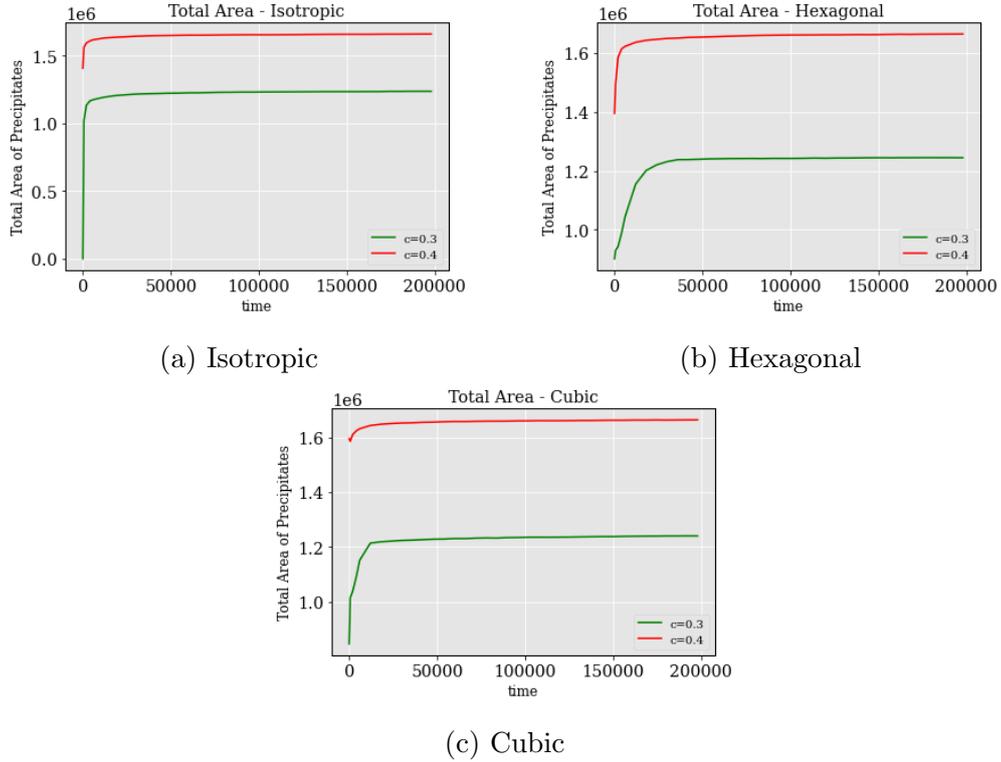


Figure 4.16: Comparing total precipitate area for the the compositions 0.3 and 0.4

We can observe very clearly in the above figures that the total area very quickly grows and asymptotically converges to its maximum value. That is, the initial growth stage in evolving microstructures is driven by dissolution of component from the matrix phase, leading to rapid component separability. The growth after this initial stage becomes slower, something which will be captured in Section 4.3.1.

### 4.3 Precipitate tracking

There are specific phenomenon of interest when studying microstructural evolution in precipitates such as directional growth or elongation, coalescence, inclination and convexity. We take such test cases and demonstrate how our algorithms can identify and track these phenomenon in the subsequent sections. We use isotropic microstructures of compositions 0.3, 0.4 and 0.5 as the test dataset to observe such phenomenon. The features associated with the precipitate being tracked and it's change in time is plotted to demonstrate the above mentioned phenomenon.

### 4.3.1 Directional growth

Some precipitates grow more in a particular direction non-uniformly, due to the presence of external precipitates or concentration concentrations. Detection of this phenomenon is usually a manual process. We apply our clustering and precipitate tracking algorithms to observe such a precipitate. Figure. 4.17 shows a precipitate growing in one direction.

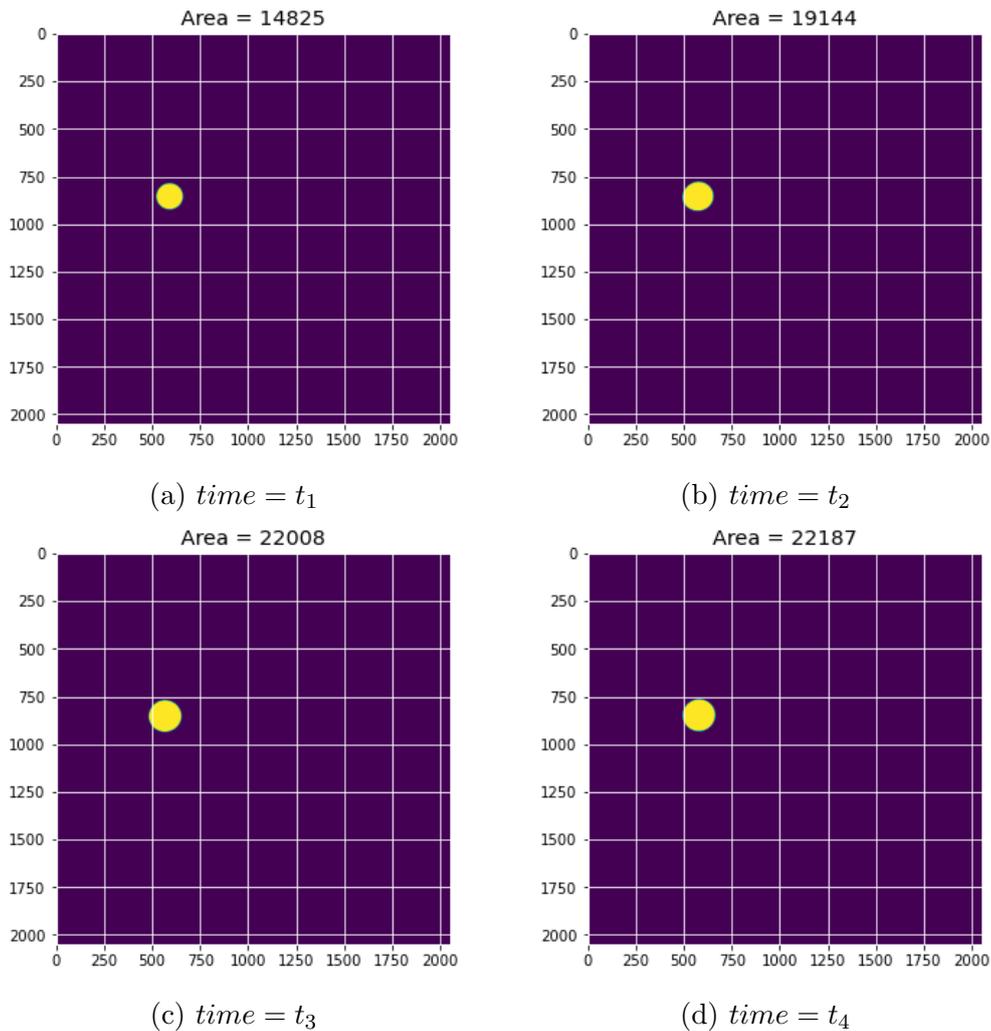


Figure 4.17: Directional Growth( $t_1 < t_2 < t_3 < t_4$ )

In bi-directional symmetric growth, although the axis of the precipitate might grow, the COG remains largely steady. The movement of the COG in a particular direction is a indication of how uni-directional the precipitate growth is. Furthermore, directional growth can also be captured by the elongation of the The plots in Fig. 4.18

capture these features. It is evident that the elongation(captured by sphericity) and the x-coordinate of the precipitate centre see sudden changes due to this directional precipitate growth.

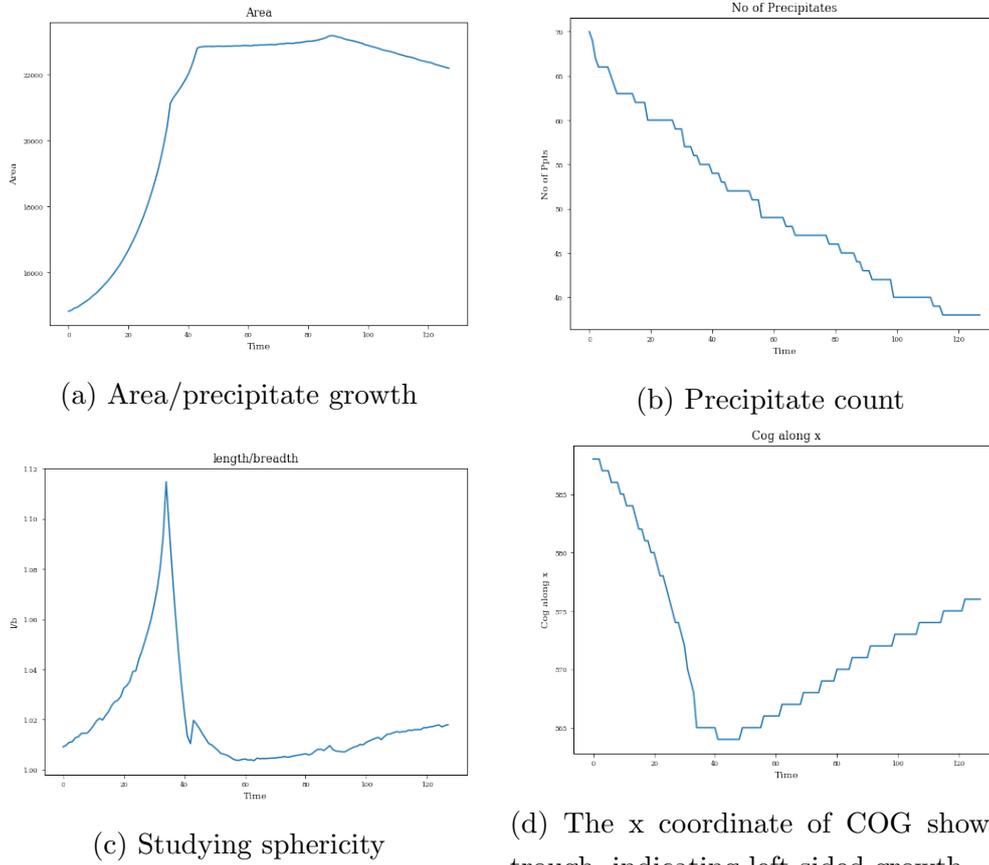


Figure 4.18: Precipitate parameter visualization

### 4.3.2 Identifying Precipitate Coalescence

Many times, 2 precipitates coalesce to form a single precipitate. Identification of such types of coalescence is easily noticed by observing the discontinuities in the area of the tracked precipitate. The point of coalescence sees a significant jump in the precipitate tracked area. Furthermore the elongation ( $\frac{l}{b}$ ) also sees a discontinuity and usually rises rapidly, since coalescence starts with an elongated structure.

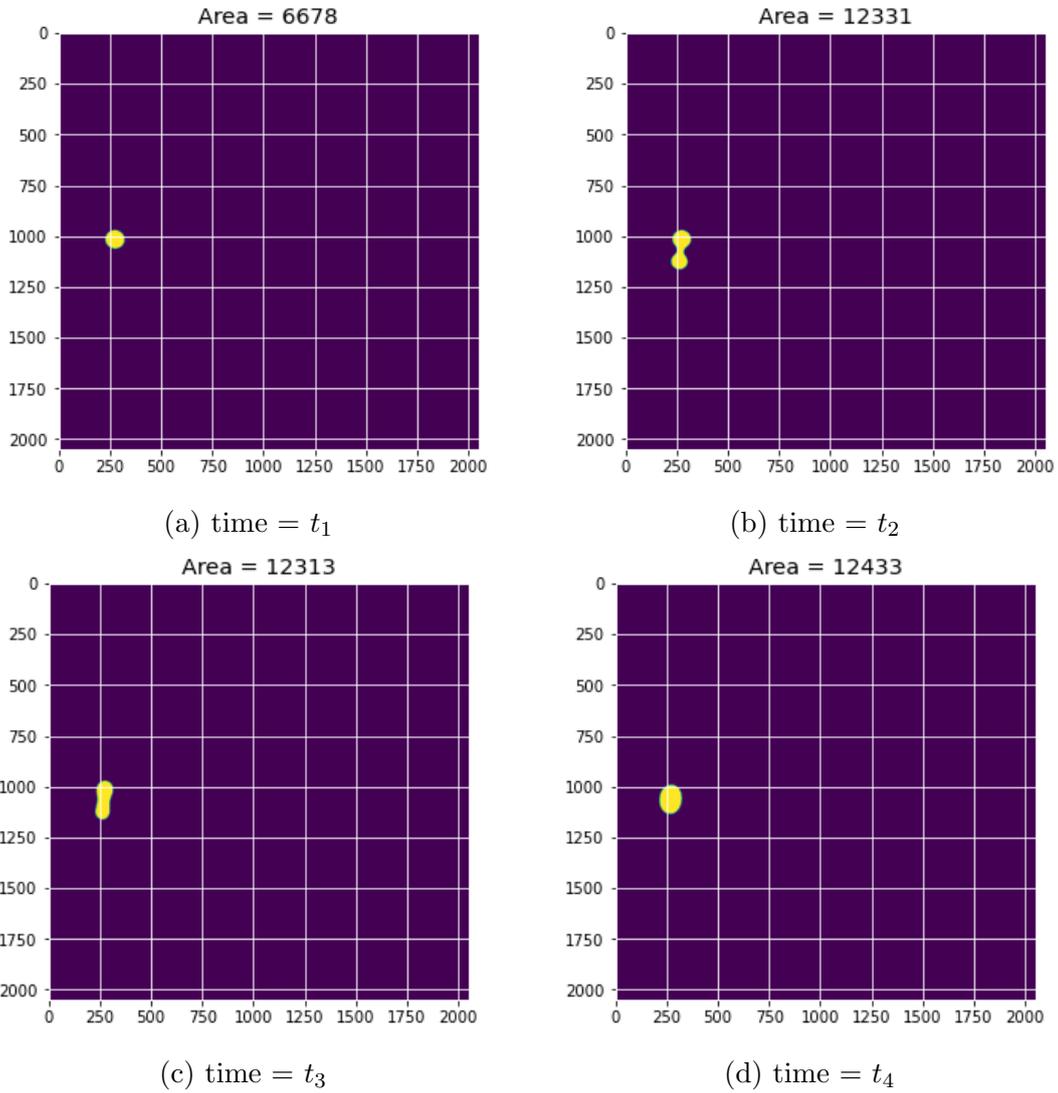
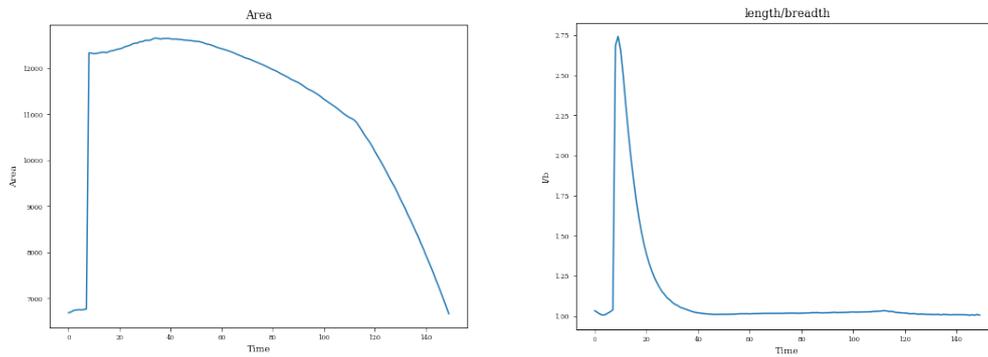
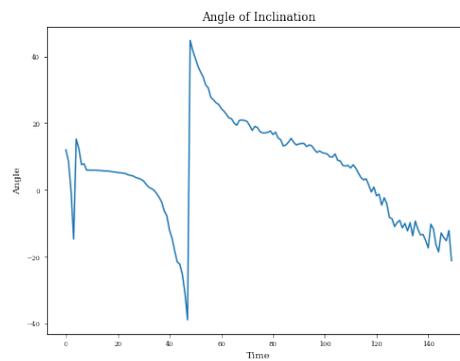


Figure 4.19: Coalescence:  $t_1 < t_2 < t_3 < t_4$

We track the microstructure and calculate the area, length/breadth ratio and inclination of the precipitate. It is easily observable that all quantities have a discontinuity in time, especially the area of the precipitate. Thus, given such plots we can identify instances of coalescence in the microstructure.



(a) Discontinuity in area implies start of coalescence (b) Sudden elongation is presage to future coalescence

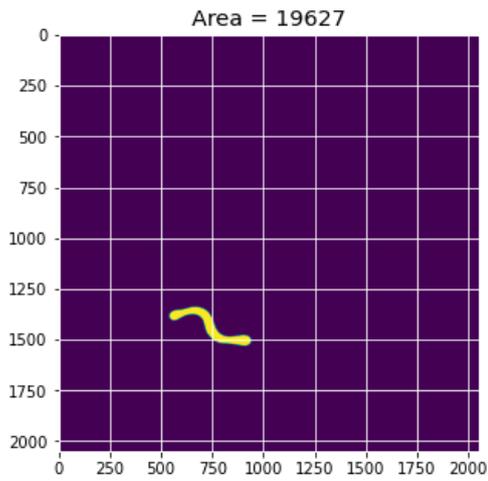


(c) Angle of inclination determines direction of agglomeration

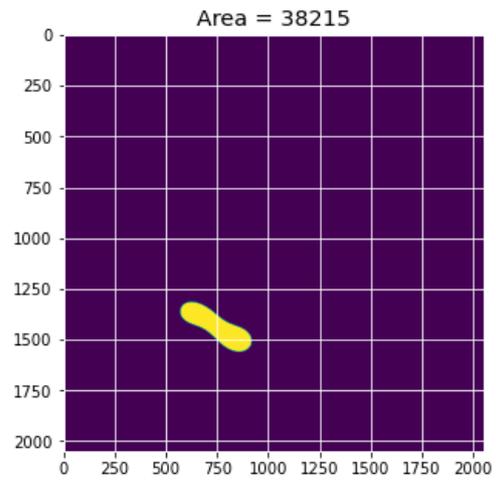
Figure 4.20: Coalescence: Identification and Tracking

### 4.3.3 Precipitate inclination

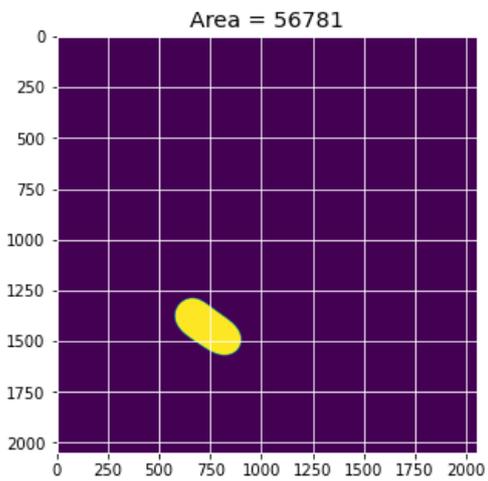
The presence of a perfectly spherical precipitate leads to no angle of inclination, so in such cases random noise is usually observed. But more often, due to anisotropy, we generally observe a detectable angle of inclination and elongation ratio. Angle of inclination is calculated by fitting the best ellipse on the precipitate and calculating the inclination with a vertical line, while elongation is defined as the ratio of major and minor axis of the same ellipse. We study one such precipitate growth here, where the inclination and elongation is easily observable.



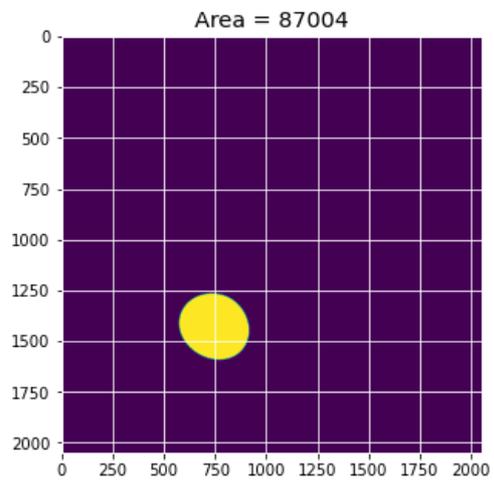
(a) time= $t_1$



(b) time= $t_2$

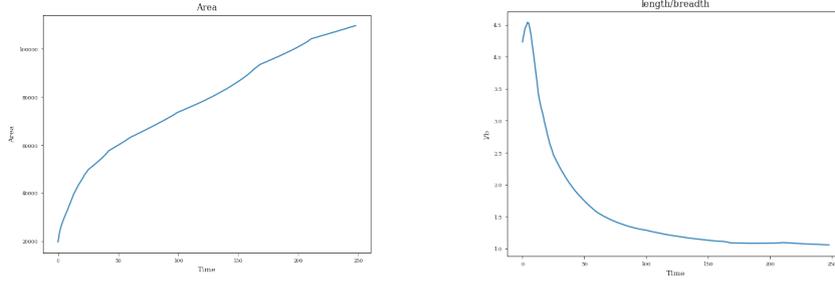


(c) time= $t_3$

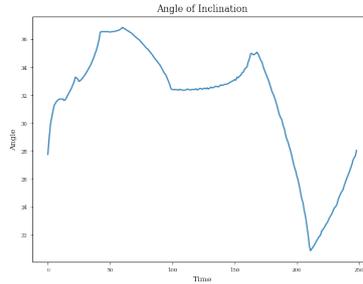


(d) time= $t_4$

Figure 4.21: Rapidly evolving precipitate to study elongation:  $t_1 < t_2 < t_3 < t_4$



(a) We observe a steady precipitate growth (b) The elongation is easily observable



(c) The angle hovers around the 36° mark(y-axis)

Figure 4.22: Inclination is easily observable in the tracked precipitates

It is easy to observe that the elongation ratio drops rapidly as the precipitate becomes more spherical. Moreover, the angle of inclination remains constant at 36°. Thus, by using these curves, we can estimate the shape change observed in a precipitate during its evolution.

## 4.4 Elastic constants

Based on the algorithm mentioned in Section 3.3, we calculate the stiffness matrix of our microstructures. We calculate them for 3 types of microstructures namely isotropic, cubic and hexagonal for precipitate compositions of 0.3, 0.4 and 0.5 for over 1000 time steps of evolution. For the calculation of the shear modulus or  $C_{44}$ , we use a non-dimensional pure shear stress state of 0.5.

The matrix phase elastic modulus or  $C_{ij}^m$  were set as  $C_{11}^m = 250$ ,  $C_{12}^m=100$  and  $C_{44}^m=75$ . The precipitate phase elastic modulus or  $C_{ij}^p$  were set as  $C_{11}^p = 500$ ,  $C_{12}^p=200$  and  $C_{44}^p=150$ . The ratios were adjusted to ensure that both the matrix and precipitate phase we individually isotropic by the Zener anisotropy parameter ( $2C_{44}/(C_{11}-C_{12}) = 1$ ).

We choose 4 loading conditions to set up Hooke's law equations, as mentioned in the table below.

Loading Conditions for Stiffness Matrix calculation			
Loading Number	$\sigma_{11}$	$\sigma_{22}$	$\sigma_{12}$
Loading 1	0.0	-0.5	0.0
Loading 2	0.4	-0.4	0.0
Loading 3	0.3	0.0	0.0
Loading 4	0.3	0.2	0.0

For each loading condition, the strain values are calculated by solving the equation of mechanical equilibrium, to get 4 stress-strain responses for each microstructure. For instance, the stress strain response for a typical 2D cubic composite microstructure of composition 0.3 is given below

Stress-strain values						
Loading Number	$\sigma_{11}$	$\sigma_{22}$	$\sigma_{12}$	$\epsilon_{11}$	$\epsilon_{22}$	$\epsilon_{12}$
Loading 1	0.0	-0.5	0.0	0.000776	-0.001956	0.0
Loading 2	0.4	-0.4	0.0	0.002186	-0.002185	0.0
Loading 3	0.3	0.0	0.0	0.001174	-0.000466	0.0
Loading 4	0.3	0.2	0.0	0.000864	0.000317	0.0

Given these loading conditions, by the method of regression mentioned in Section 3.3, we can estimate the values of  $C_{11}$  and  $C_{12}$ . We also calculate the 95% confidence interval for the values of  $C_{11}$  and  $C_{12}$  to ensure a good fit. The table below demonstrates the obtained values of  $C_{11}$  and  $C_{12}$  for the microstructure image represented in table above.

$C_{ij}$	Lower 95% confidence	$C_{ijkl}$ value	Upper 95% confidence
$C_{11}$	303.22541	303.22548	303.22555
$C_{12}$	120.30360	120.30364	120.30368
$C_{22}$	303.38537	303.38540	303.38544

These values are well within the upper and lower bounds from the rule of mixtures, which bounds  $C_{11}$  and  $C_{22}$  in [294,325] and  $C_{12}$  in [117,130]. We also check for the Hashin-Shtrikman bounds [35] for our compositions and elastic values. Our bulk modulus(K) bounds are fairly close to the bounds derived from Hashin-Shtrikman, as demonstrated below.

Composition	$C_{11}$	$C_{12}$	K	K upper bound	K Lower bound
0.3	303	121	181.69	184.61	181.57
0.4	326	129	194.96	197.72	194.11
0.5	354	138	210.52	211.76	207.69

Similarly, the upper and lower bounds for shear modulus from the Hashin-Shtrikman method are in the interval [105,107], while our calculations resulted in a shear modulus of 96.40 for a composition of 0.5. Similar behaviour was observed at different compositions, where the observed shear modulus was 10% lower than the Hashin-Shtrikman lower bound. We expect this to be an effect of our shear loading condition and the effect of shear loading on periodic boundary conditions, and thus this will be investigated further.

We can also observe how the elastic constants change as the microstructure evolves, and the change in the constants on changing the composition.

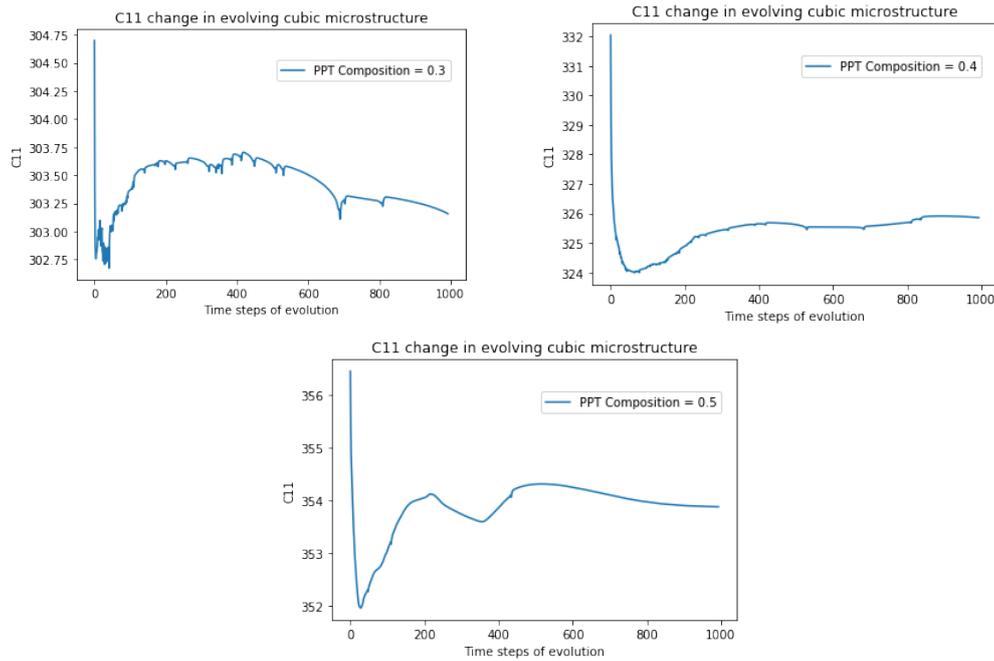


Figure 4.23: Plotting  $C_{11}$  in an evolving microstructure

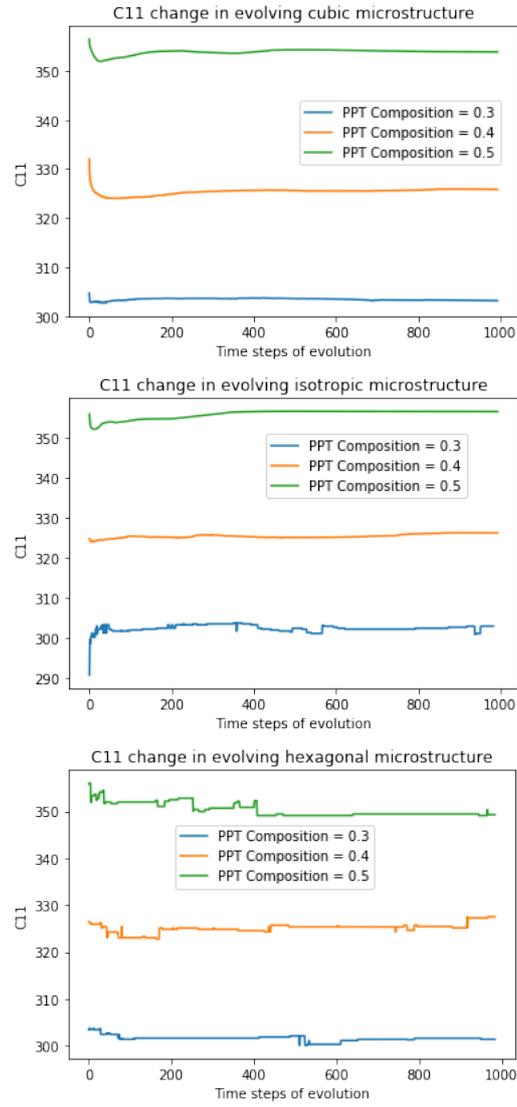


Figure 4.24: Comparing  $C_{11}$  change with composition

We can also fix the composition and study the effect of anisotropy on the elastic constants. The figure below shows the  $C_{11}$  of isotropic, cubic and hexagonal microstructures evolving in time with varying compositions.

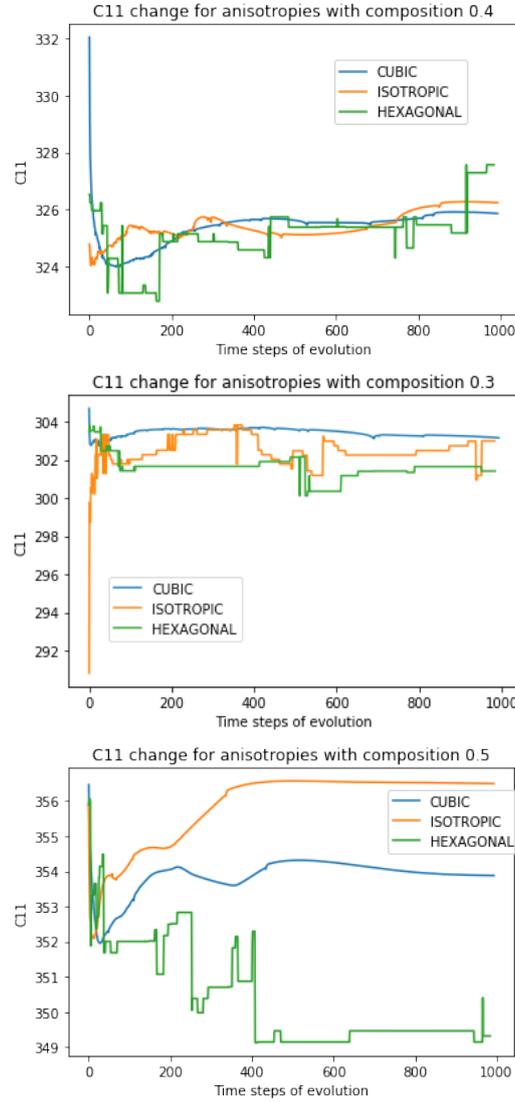


Figure 4.25: Comparing  $C_{11}$  change with anisotropy

From Fig. 4.25, it is clear that for the compositions 0.3 and 0.4, anisotropy does not play a big part in the elastic constants since the elastic modulus are fairly close to each other. Only at a 50% precipitate composition is there a tangible difference in  $C_{11}$ . This could be because the spatial distribution of the varying precipitates only dominates at a 0.5 composition, when the microstructure looks like fine spinodes. Furthermore, the tetragonal approximations taken to calculate  $C_{11}$  or  $C_{12}$  leads to a higher error bar on the elastic modulus, for instance the  $C_{12}$  values' 95% confidence interval can range from 138 to 140, while we report a  $C_{12}$  of 139. Thus, better measures or techniques are needed to perfectly quantify the properties of hexagonal composites.

Now that we have calculated the trend-lines for the elastic modulus or stiffness matrix, we try to correlate this property with the characterisation techniques we have discussed before.

## 4.5 Microstructure Property Correlation

We now try to explain the trends in the elastic constants in the evolving microstructures based on the statistical measures we have calculated. Let us consider the flattening of  $C_{11}$  curve in cubic microstructures after the initial stages of evolution, as shown in Figure 4.23. The  $C_{11}$  values flatten out near 325 after 600 time steps, while they show significant volatility in the first 100 time steps.

To capture this, let us consider the cross-correlations of the cubic microstructures, an example shown below.

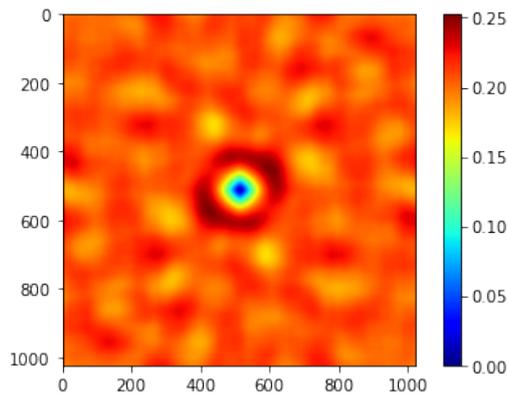


Figure 4.26: Cross-correlation matrix of a cubic microstructure

Traditionally, a radial correlation plot would be used to express the precipitate distribution. To incorporate a feature of anisotropy, let us consider how the probability in the correlation matrix changes at different angles, like 0, 45, 90 and 135 degrees. This gives us 4 plots which capture probability differences along 4 directions.

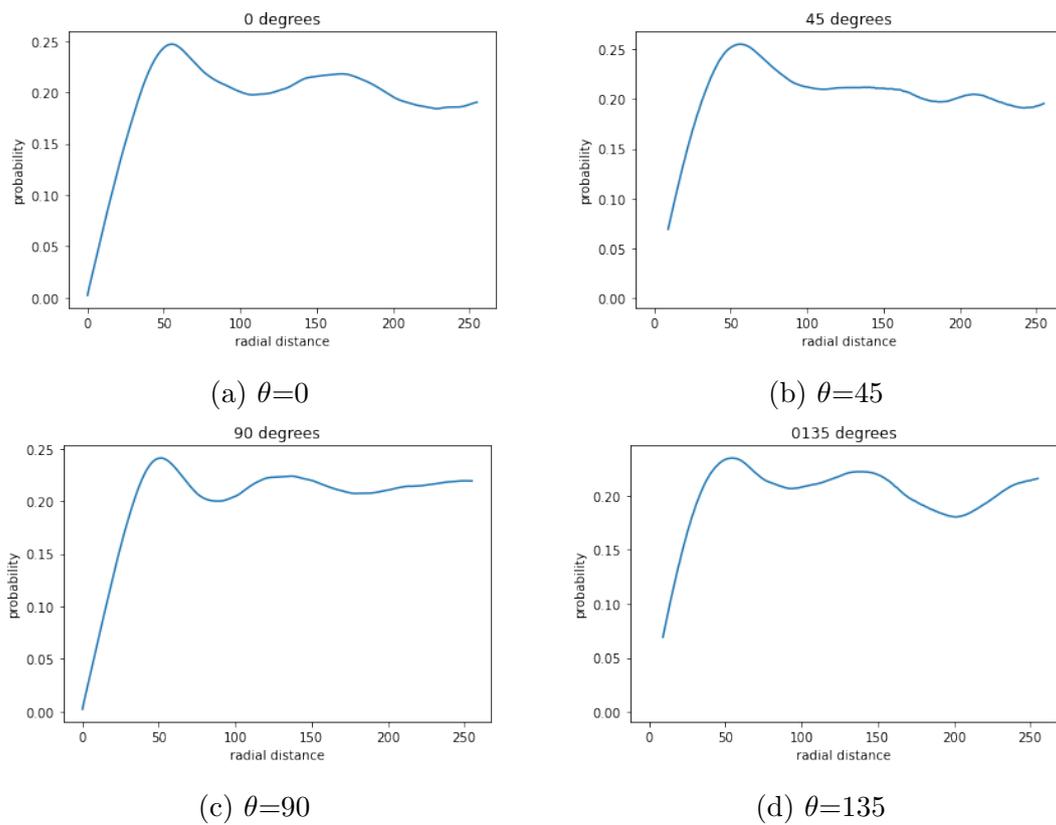


Figure 4.27: Radial correlations along angles

These 4 act as our feature vectors, and can thus be concatenated into a single feature vector to represent our microstructure, which gives us a single plot capturing the radial distribution and anisotropy.

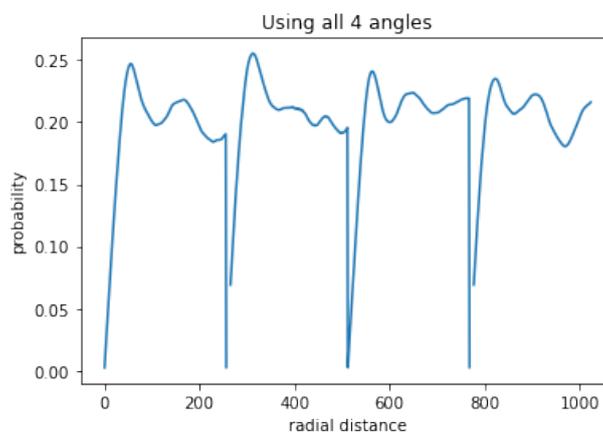
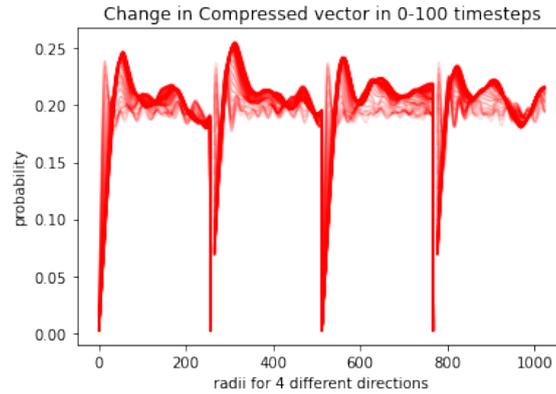


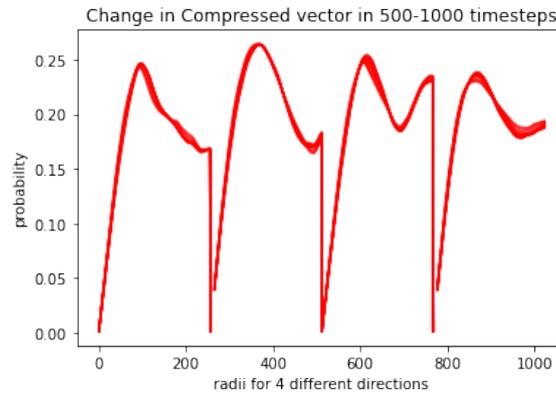
Figure 4.28: Final feature vector

Let us now see how these probability values along particular angles change with the evolution of the microstructure, in the initial and latter stages of the microstructure respectively.

To capture this, we superimpose these plots for the different time steps, and visualize the higher time steps by darker lines. The two figures show how much these feature vectors vary in the first 100 time-steps and the last 500 time-steps respectively.



(a) Initial evolution



(b) Latter stages of evolution

Figure 4.29: Comparing changes in feature vectors in different evolution regimes

As seen in Figure 4.29, there is significant variation in the feature vector in the first 100 time steps, as demonstrated by the band of shaded red lines in Fig. 4.29a.

In latter stages, the feature vectors are fairly close, and appear co-linear in Fig. 4.29b. This demonstrates comprehensively that the flattening of elastic modulus curve in latter stages evolving microstructures is correlated with the constant nature of their feature vectors.

# Chapter 5

## Summary

Data oriented material design is dependent on efficient and useful feature extraction from sources like microstructure images. Algorithms and workflows are needed to simplify extraction of statistical and geometric features, since manual extraction is time consuming. Physical tests and experiments are also time and cost consuming, which makes computational data generation an important aspect of materials design.

Firstly, this report aims to identify the features best suited for characterisation and provide robust computationally efficient algorithms for their extraction. By comparing various algorithms and features from literature, and implementing them on computationally generated microstructures, the algorithms and computations required are laid out in this report. The results and plots obtained validate the algorithms, and also provide a visual method of interpreting microstructures. The features extracted quantify the inherent composition distribution, and help draw conclusions from them.

Secondly, we lay out robust computational techniques to calculate the properties associated with microstructures, and provide statistical frameworks and approximations that simplify their extraction. This can be thought of as an exercise in building the data that can then be used to look for correlations or patterns with our structures.

Finally, we give examples to show how this structure-property correlation can be developed. This is the end goal that we hoped to reach, to fast forward the prediction of structures which lead to desirable material properties.

# Bibliography

- [1] Surya R Kalidindi, Andrew J Medford, and David L Mcdowell. “Vision for data and informatics in the future materials innovation ecosystem”. In: *JOM* 68.8 (2016), pp. 2126–2137.
- [2] David T Fullwood, Stephen R Niezgod, and Surya R Kalidindi. “Microstructure reconstructions from 2-point statistics using phase-recovery algorithms”. In: *Acta Materialia* 56.5 (2008), pp. 942–948.
- [3] Liyuan Huang and Chen Ling. “Practicing deep learning in materials science: An evaluation for predicting the formation energies”. In: *Journal of Applied Physics* 128.12 (2020), p. 124901.
- [4] John W Cahn and John E Hilliard. “Free energy of a nonuniform system. I. Interfacial free energy”. In: *The Journal of chemical physics* 28.2 (1958), pp. 258–267.
- [5] David A Porter and Kenneth E Easterling. *Phase transformations in metals and alloys (revised reprint)*. CRC press, 2009.
- [6] Hideki Kobayashi and Ryoichi Yamamoto. “Implementation of Lees–Edwards periodic boundary conditions for direct numerical simulations of particle dispersions under shear flow”. In: *The Journal of chemical physics* 134.6 (2011), p. 064110.
- [7] Jean-Claude Michel, Hervé Moulinec, and Pierre Suquet. “Effective properties of composite materials with periodic microstructure: a computational approach”. In: *Computer methods in applied mechanics and engineering* 172.1-4 (1999), pp. 109–143.
- [8] MP Gururajan. “Elastic inhomogeneity effects on microstructures: a phase field study”. In: *Department of Metallurgy, Indian Institute of Science, Bangalore* 560012 (2006).

- [9] Hervé Moulinec and Pierre Suquet. “A numerical method for computing the overall response of nonlinear composites with complex microstructure”. In: *Computer methods in applied mechanics and engineering* 157.1-2 (1998), pp. 69–94.
- [10] A Anthoine. “Derivation of the in-plane elastic characteristics of masonry through homogenization theory”. In: *International journal of solids and structures* 32.2 (1995), pp. 137–163.
- [11] DT Fullwood, SR Kalidindi, SR Niezgoda, A Fast, and N Hampson. “Gradient-based microstructure reconstructions from distributions using fast Fourier transforms”. In: *Materials Science and Engineering: A* 494.1-2 (2008), pp. 68–72.
- [12] MG Rozman and Marcel Utz. “Uniqueness of reconstruction of multiphase morphologies from two-point correlation functions”. In: *Physical review letters* 89.13 (2002), p. 135501.
- [13] Binglin Lu and Salvatore Torquato. “Lineal-path function for random heterogeneous materials”. In: *Physical Review A* 45.2 (1992), p. 922.
- [14] H Singh, AM Gokhale, SI Lieberman, and S Tamirisakandala. “Image based computations of lineal path probability distributions for microstructure representation”. In: *Materials Science and Engineering: A* 474.1-2 (2008), pp. 104–111.
- [15] Hongyi Xu, Dmitriy A Dikin, Craig Burkhart, and Wei Chen. “Descriptor-based methodology for statistical characterization and 3D reconstruction of microstructural materials”. In: *Computational Materials Science* 85 (2014), pp. 206–216.
- [16] Hongyi Xu, Yang Li, Catherine Brinson, and Wei Chen. “A descriptor-based design methodology for developing heterogeneous microstructural materials system”. In: *Journal of Mechanical Design* 136.5 (2014).
- [17] CLY Yeong and Salvatore Torquato. “Reconstructing random media”. In: *Physical review E* 57.1 (1998), p. 495.
- [18] Ramin Bostanabad, Yichi Zhang, Xiaolin Li, Tucker Kearney, L Catherine Brinson, Daniel W Apley, Wing Kam Liu, and Wei Chen. “Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques”. In: *Progress in Materials Science* 95 (2018), pp. 1–41.
- [19] Xingchen Liu and Vadim Shapiro. “Random heterogeneous materials via texture synthesis”. In: *Computational Materials Science* 99 (2015), pp. 177–189.

- [20] Noah H Paulson, Matthew W Priddy, David L McDowell, and Surya R Kalidindi. “Reduced-order structure-property linkages for polycrystalline microstructures based on 2-point statistics”. In: *Acta Materialia* 129 (2017), pp. 428–438.
- [21] Yuksel C Yabansu, Almambet Iskakov, Anna Kapustina, Sudhir Rajagopalan, and Surya R Kalidindi. “Application of Gaussian process regression models for capturing the evolution of microstructure statistics in aging of nickel-based superalloys”. In: *Acta Materialia* 178 (2019), pp. 45–58.
- [22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.
- [23] Nicholas Lubbers, Turab Lookman, and Kipton Barros. “Inferring low-dimensional microstructure representations using convolutional neural networks”. In: *Physical Review E* 96.5 (2017), p. 052111.
- [24] Aditya Venkatraman, David Montes de Oca Zapiain, and Surya R Kalidindi. “Reduced-Order Models for Ranking Damage Initiation in Dual-Phase Composites Using Bayesian Neural Networks”. In: *JOM* (2020), pp. 1–11.
- [25] Andrew R Roosen and W Craig Carter. “Simulations of microstructural evolution: anisotropic growth and coarsening”. In: *Physica A: Statistical Mechanics and Its Applications* 261.1-2 (1998), pp. 232–247.
- [26] C-L Park, Peter W Voorhees, and K Thornton. “Application of the level-set method to the analysis of an evolving microstructure”. In: *Computational materials science* 85 (2014), pp. 46–58.
- [27] James Albert 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.
- [28] Sudipta Biswas, Daniel Schwen, Hao Wang, Maria Okuniewski, and Vikas Tomar. “Phase field modeling of sintering: Role of grain orientation and anisotropic properties”. In: *Computational Materials Science* 148 (2018), pp. 307–319.
- [29] Hongyi Xu and Chulheung Bae. “Stochastic 3D microstructure reconstruction and mechanical modeling of anisotropic battery separators”. In: *Journal of Power Sources* 430 (2019), pp. 67–73.
- [30] D Wheeler, D Brough, T Fast, S Kalidindi, and A Reid. “PyMKS”. In: *Materials Knowledge System in python, figshare* 2016 (2014).

- [31] Joseph Hoshen and Raoul Kopelman. “Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm”. In: *Physical Review B* 14.8 (1976), p. 3438.
- [32] Tobin Fricke. “The hoshen-kopelman algorithm”. In: *URL: <http://www.ocf.berkeley.edu/~fricke/projects/hoshenkopelman/hoshenkopelman.html>* (2004).
- [33] Linge Bai and David Breen. “Calculating center of mass in an unbounded 2D environment”. In: *Journal of Graphics Tools* 13.4 (2008), pp. 53–60.
- [34] Esa Rahtu, Mikko Salo, and Janne Heikkila. “A new convexity measure based on a probabilistic interpretation of images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.9 (2006), pp. 1501–1512.
- [35] Zvi Hashin and Shmuel Shtrikman. “A variational approach to the theory of the elastic behaviour of multiphase materials”. In: *Journal of the Mechanics and Physics of Solids* 11.2 (1963), pp. 127–140.