VISVESVARAYA TECHNOLOGICAL UNIVERSITY- BELAGAVI

Project Report on

Monte Carlo Simulation of Grain Growth and Sintering

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

MECHANICAL ENGINEERING

By

Sirish Mulugu	Pradhay Prashanth	Vidyut Rao
1PI13ME165	1PI13ME107	1PI13ME183

Under the Guidance of

Dr. Anirban Chakraborty
Assistant Professor - Chemistry
Department of Science and Humanities
PESIT, Bangalore - 560085

Carried out at

DEPARTMENT OF MECHANICAL ENGINEERING
P E S INSTITUTE OF TECHNOLOGY
(Autonomous Institute under VTU, Belagavi)
100 Feet Ring Road, BSK III Stage, Bangalore – 560085

2017

VISVESVARAYA TECHNOLOGICAL UNIVERSITY- BELAGAVI

DEPARTMENT OF MECHANICAL ENGINEERING

P E S INSTITUTE OF TECHNOLOGY

(Autonomous under VTU, Belagavi), Bangalore – 560085

CERTIFICATE

Certified that the final year project work entitled "Monte Carlo Simulations of Grain Growth and Sintering", is a bona fide work carried out by Vidyut Rao, Sirish Mulugu and Pradhay Prashanth bearing University Seat Number (1PI13ME183), (1PI13ME165) and (1PI13ME107) respectively, in partial fulfilment for the award of ***Bachelor of Engineering*** *in* ***Mechanical Engineering*** *of the Visvesvaraya Technological University, Belagavi, during the year 2017. It is certified that all corrections/suggestions indicated during internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the said degree.*

Signature of Guide

Dr. Anirban Chakraborty

Assistant Professor - Chemistry

Department of Science and Humanities, PESIT

Signature of HOD

Dr. V. Krishna

Professor and HOD

Dept. of Mech. Engg, PESIT

Signature of Principal

Dr. K S Sridhar

Principal

PESIT

Examiners

Signature

1.

2.

Declaration

We Sirish Mulugu (1PI13ME165), Pradhay Prashanth (1PI13ME107), Vidyut Rao (1PI13ME183) hereby declare that the thesis entitled "***MONTE CARLO SIMULATIONS OF GRAIN GROWTH AND SINTERING*"** has been carried under the guidance of **Dr. Anirban Chakraborty** , Assistant professor PESIT in partial fulfilment of the Bachelor of Engineering in Mechanical Engineering, of **Visvesvaraya Technological University-Belagavi**.

We also declare that to the best of our knowledge and belief, the work reported here does not form part of any other thesis or dissertation on the basis of which a degree or award was confirmed on earlier occasion by a BE student.

Sirish Mulugu

(1PI13ME165)

Pradhay Prashanth

(1PI13ME107)

Vidyut Rao

(1PI13ME183)

Abstract

The thermodynamics and kinetics of grain growth is a problem that has actively been pursued in the area of material science. Experiments have been performed both to obtain accurate values of thermodynamic and kinetic parameters, that had led to theoreticians from diverse background to attempt providing a framework for the understanding of this technologically important problem. The growing interest to model grain growth in recent times has led to the proliferation of numerous stochastic simulation techniques. The present work aims at studying grain growth and the different phenomena associated using Monte Carlo simulations.

On doing so one can apply the results and core concepts to the simulation of liquid phase sintering. Selective Laser Sintering employs a laser to sinter a coated power. The mechanism follows that of liquid phase sintering and much like grain growth, it can be modelled using stochastic simulation techniques.

The core mechanisms in sintering include vacancy annihilation, pore migration and solution re-precipitation. Working on MATLAB, this work aims to develop concise and highly functional algorithms that can achieve the aforementioned mechanisms.

The project's eventual goal is to be able to accurately predict the microstructure of a 3D printed product, printed by employing the Selective Laser Sintering mechanism. Using the microstructure at hand, it would be useful to gauge the strength of the final product. Most 3D printed products have poor strength along the longitudinal axis (normal to powder layers), this work aims at calculating said strength.

Acknowledgement

This project would not have been possible without the kind support and help of many individuals and our University. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Dr. Anirban Chakraborty** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in completing the project.

We would like to express my gratitude towards **Dr. Suneel Motru and Dr. D Sethuram** for their kind co-operation and encouragement which helped us in completion of this project.

We also thank the HOD **Dr. V Krishna** and our Principal **Dr. K S Sridhar** for their continuous support

We would like to express our special gratitude and thanks to our parents for giving us their endless support throughout.

My thanks and appreciations also go to my friends in developing the project and people who have willingly helped me out with their abilities.

Lastly, our sincere thanks to **PES Institute of Technology** for providing us this wonderful opportunity to carry about this project.

Table of Contents

Table of Figures

1. Introduction

Sintering of powders is one of the most important processes for the development of polycrystalline materials. The microstructure of a material is of fundamental importance in the processing of ceramics and metals since it affects the physical properties of the final product. Progress in our ability to satisfactorily predict microstructure and its properties has been quite slow owing to complexity of physical processes involved. The complete prediction of microstructural development in polycrystalline solids as a function of time and temperature is a major of objective in materials science.

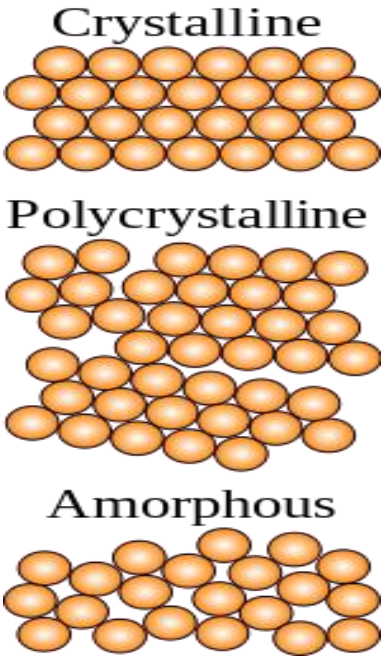

Figure 1: Different Crystal Structures

Grain size is a very important characteristic for evaluating properties of the materials as it directly affects the strength of the polycrystalline material under observation. Since the past few decades, there has been a growing interest to model grain growth by using stochastic simulation techniques. The present work aims at studying grain growth and the different phenomena associated using Monte Carlo simulations.

On completing the preliminary 2D simulation of grain growth, similar procedures were adopted to obtain the 3 dimensional counterpart. These results were then verified against the results obtained from implementing analytical methods.

Post simulation of grain growth, sintering, more specifically – Liquid Phase Sintering was studied with the goal of eventually analysing the strength of 3D printed products, which employ Selective Laser Sintering (SLS) as one of the processes.

2. Literature Review

2.1 Grain Growth

Grain growth as a process is viewed as the final stage of annealing. **Annealing** is a heat treatment that alters the physical and sometimes chemical properties of a material to increase its ductility and reduce its hardness, making it more workable. It involves heating a material to above its recrystallization temperature, maintaining a suitable temperature, and then cooling.

In annealing, atoms migrate in the crystal lattice and the number of dislocations decreases, leading to the change in ductility and hardness. Heat increases the rate of diffusion by providing the energy needed to break bonds. The movement of atoms has the effect of redistributing and eradicating the dislocations in metals and (to a lesser extent) in ceramics. This alteration to existing dislocations allows a metal object to deform more easily, increasing its ductility.

The amount of process-initiating Gibbs free energy in a deformed metal is also reduced by the annealing process. In practice and industry, this reduction of Gibbs free energy is termed stress relief.

The three stages of the annealing process that proceed as the temperature of the material is increased are: **recovery**, **recrystallization**, and **grain growth**.

2.1.1 Recovery

Recovery is a process by which deformed grains can reduce their stored energy by the removal or rearrangement of defects in their crystal structure. These defects, primarily dislocations, are introduced by plastic deformation of the material and act to increase the yield strength of a material. Since recovery reduces the dislocation density the process is normally accompanied by a reduction in a materials strength and a simultaneous increase in the ductility. As a result, recovery may be considered beneficial or detrimental depending on the circumstances. It is so called because there is a recovery of the electrical conductivity due to a reduction in dislocations.

2.1.2 Recrystallization

Recrystallization is a process by which deformed grains are replaced by a new set of defects-free grains that nucleate and grow until the original grains have been entirely consumed. Recrystallization is usually accompanied by a reduction in the strength and hardness of a material and a simultaneous increase in the ductility. Thus, the process may

be introduced as a deliberate step in metals processing or may be an undesirable by-product of another processing step. The most important industrial uses are the softening of metals previously hardened by cold work, which have lost their ductility, and the control of the grain structure in the final product. Recovery competes with recrystallization, as both are driven by the stored energy, but is also thought to be a necessary prerequisite for the nucleation of recrystallized grains.

2.1.3 Grain Growth

Grain growth is the increase in size of grains in a material at high temperature. This occurs when recovery and recrystallization are complete and further reduction in the internal energy can only be achieved by reducing the total area of grain boundary.

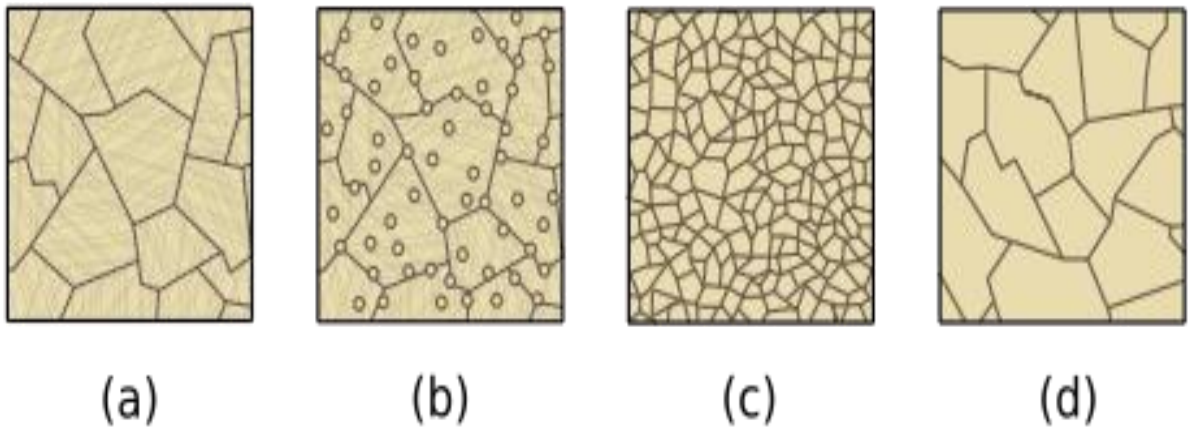

Figure 2: Recrystallization of a metallic material (a → b) and crystal grains growth (b → c → d)

Grain growth has long been studied primarily by the examination of sectioned, polished and etched samples under the optical microscope. Although such methods enabled the collection of a great deal of empirical evidence, particularly with regard to factors such as temperature or composition, the lack of crystallographic information limited the development of an understanding of the fundamental physics. Nevertheless, the following became well-established features of grain growth:

I. Grain growth occurs by the movement of grain boundaries and not by coalescence (i.e. like water droplets)
II. Boundary movement is discontinuous and the direction of motion may change suddenly.

III. One grain may grow into another grain whilst being consumed from the other side
IV. The rate of consumption often increases when the grain is nearly consumed
V. A curved boundary typically migrates towards its centre of curvature
VI. When grain boundaries in a single phase meet at angles other than 120 degrees, the grain included by the more acute angle will be consumed so that the angles approach 120 degrees.

2.1.4 Driving force for grain growth

The boundary between one grain and its neighbour (grain boundary) is a defect in the crystal structure and so it is associated with a certain amount of energy. As a result, there is a thermodynamic driving force for the total area of boundary to be reduced. If the grain size increases, accompanied by a reduction in the actual number of grains per volume, then the total area of grain boundary will be reduced.

2.1.5 Types of grain growth:

I. Ideal Grain Growth

Ideal grain growth is a special case of normal grain growth where boundary motion is driven only by local curvature of the grain boundary. It results in the reduction of the total amount of grain boundary surface area i.e. total energy of the system. Additional contributions to the driving force by e.g. elastic strains or temperature gradients are neglected.

II. Normal Grain Growth

Here the microstructure evolves from state A to B (in this case the grains get larger) in a uniform manner, as shown.

III. Abnormal Grain Growth

In this case the changes occur heterogeneously and specific transformed and untransformed regions may be identified. Abnormal or discontinuous grain growth is characterised by a subset of grains growing at a high rate and at the expense of their neighbours and tends to result in a microstructure dominated by a few very large grains. In order for this to occur the subset of grains must possess some advantage over their competitors such as a high grain boundary energy, locally high grain boundary mobility, favourable texture or lower local second-phase particle density.

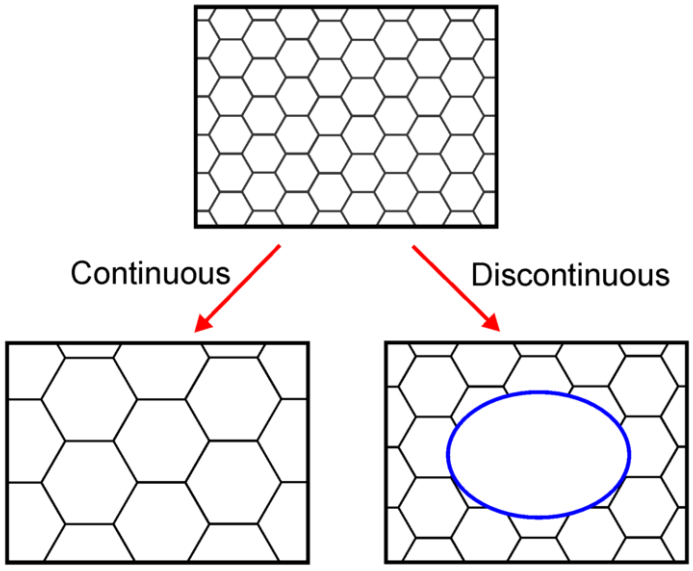

Figure 3: Distinction between continuous (normal) grain growth, where all grains grow at roughly the same rate, and discontinuous (abnormal) grain growth, where one grain grows at a much greater rate than its neighbours.

2.2 Sintering

Sintering is the process of compacting and forming a solid mass of material by heat or pressure without melting it to the point of liquefaction.

Sintering happens naturally in mineral deposits or as a manufacturing process used with metals, ceramics, plastics, and other materials. The atoms in the materials diffuse across the boundaries of the particles, fusing the particles together and creating one solid piece. Because the sintering temperature does not have to reach the melting point of the material, sintering is often chosen as the shaping process for materials with extremely high melting points such as tungsten and molybdenum. The study of sintering in metallurgy powder-related processes is known as powder metallurgy.

Sintering is effective when the process reduces the porosity and enhances properties such as strength, electrical conductivity, translucency and thermal conductivity; yet, in other cases, it may be useful to increase its strength but keep its gas absorbency constant as in filters or catalysts. During the firing process, atomic diffusion drives powder surface elimination in different stages, starting from the formation of necks between powders to final elimination of small pores at the end of the process.

The driving force for densification is the change in free energy from the decrease in surface area and lowering of the surface free energy by the replacement of solid-vapor interfaces. It forms new but lower-energy solid-solid interfaces with a total decrease in free energy occurring. On a microscopic scale, material transfer is affected by the change in pressure and differences in free energy across the curved surface. If the size of the particle is small (and its curvature is high), these effects become very large in magnitude. The change in energy is much higher when the radius of curvature is less than a few micrometres, which is one of the main reasons why much ceramic technology is based on the use of fine-particle materials.

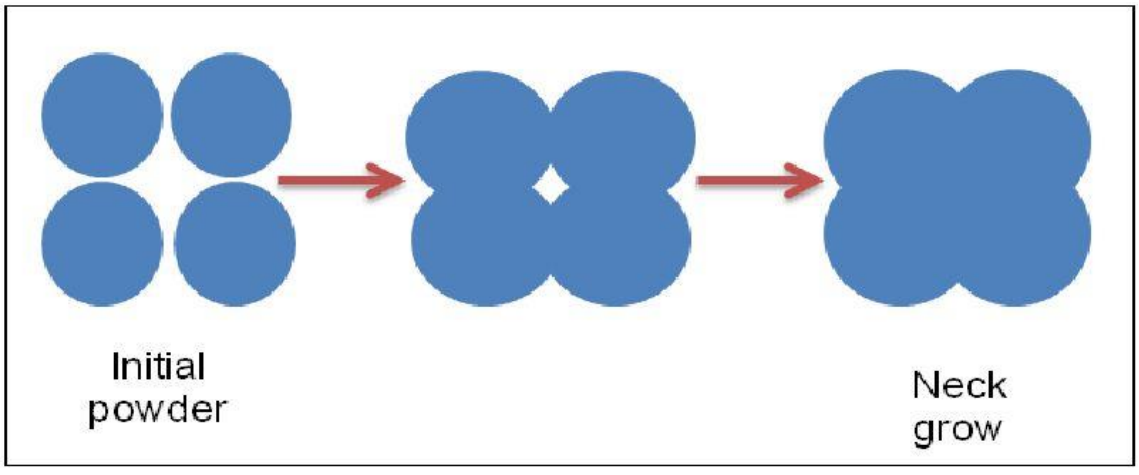

Figure 4: Basic visualisation of Sintering

2.2.1 Sintering mechanisms

Sintering occurs by diffusion of atoms through the microstructure. This diffusion is caused by a gradient of chemical potential – atoms move from an area of higher chemical potential to an area of lower chemical potential. The different paths the atoms take to get from one spot to another are the sintering mechanisms. The six common mechanisms are:

1. Surface diffusion – Diffusion of atoms along the surface of a particle
2. Vapour transport – Evaporation of atoms which condense on a different surface
3. Lattice diffusion from surface – atoms from surface diffuse through lattice
4. Lattice diffusion from grain boundary – atom from grain boundary diffuses through lattice
5. Grain boundary diffusion – atoms diffuse along grain boundary
6. Plastic deformation – dislocation motion causes flow of matter

Also one must distinguish between densifying and non-densifying mechanisms. 1–3 above are non-densifying– they take atoms from the surface and rearrange them onto another surface or part of the same surface. These mechanisms simply rearrange matter inside of porosity and do not cause pores to shrink. Mechanisms 4–6 are densifying mechanisms – atoms are moved from the bulk to the surface of pores thereby eliminating porosity and increasing the density of the sample.

2.2.2 Types of Sintering

1. Ceramic sintering

Sintering is part of the firing process used in the manufacture of pottery and other ceramic objects. These objects are made from substances such as glass, alumina, zirconia, silica, magnesia, lime, beryllium oxide, and ferric oxide. Some ceramic raw materials have a lower affinity for water and a lower plasticity index than clay, requiring organic additives in the stages before sintering.

2. Sintering of metallic powders

Most, if not all, metals can be sintered. This applies especially to pure metals produced in vacuum which suffer no surface contamination. Sintering under atmospheric pressure requires the use of a protective gas, quite often endothermic gas. Sintering, with subsequent reworking, can produce a great range of material properties. Changes in density, alloying, or heat treatments can alter the physical characteristics of various products.

3. Plastics sintering

Plastic materials are formed by sintering for applications that require materials of specific porosity. Sintered plastic porous components are used in filtration and to control fluid and gas flows. Sintered plastics are used in applications requiring wicking properties, such as marking pen nibs. Sintered ultra-high molecular weight polyethylene materials are used as ski and snowboard base materials. The porous texture allows wax to be retained within the structure of the base material, thus providing a more durable wax coating.

4. Electric current assisted sintering

These techniques employ electric currents to drive or enhance sintering. The primary purpose of is the industrial scale production of filaments for incandescent lamps by compacting tungsten or molybdenum particles. The applied current was particularly effective in reducing surface oxides that increased the emissivity of the filaments.

5. Liquid phase sintering

For materials which are difficult to sinter, a process called liquid phase sintering is commonly used. Materials for which liquid phase sintering is common are Si3N4, WC, SiC, and more. Liquid phase sintering is the process of adding an additive to the powder which will melt before the matrix phase.

Due to the numerous advantages of Liquid Phase Sintering, it finds the greatest application in industry and is the type of sintering this work aims to model.

2.2.3 Liquid Phase Sintering

During liquid phase sintering a liquid phase coexists with a particulate solid at the sintering temperature. The liquid phase usually enhances the rate of interparticle bonding during sintering. Accompanying interparticle bonding are significant changes in the pore structure and compact properties including strength, ductility, conductivity, magnetic permeability, and corrosion resistance.

Classic Sequence of Stages

The classic liquid phase sintering system densifies in three overlapping stages .Initially, the mixed powders are heated to a temperature where a liquid forms. With liquid formation there is rapid initial densification due to the capillary force exerted by the wetting liquid on the solid particles. The elimination of porosity occurs as the system minimizes its surface energy. During rearrangement, the compact responds as a viscous solid to the capiIIary action. The elimination of porosity increases the compact viscosity. As a

consequence the densification rate continuously decreases. The amount of densification attained by rearrangement is dependent on the amount of liquid, particle size, and solubility of the solid in the liquid. Usually finer particles give better rearrangement. Full density (zero porosity) is possible by rearrangement if enough liquid is formed. It is estimated that 35 volume percent liquid is needed to obtain full density by rearrangement processes.

1. **Rearrangement** – As the liquid melts capillary action will pull the liquid into pores and also cause grains to rearrange into a more favorable packing arrangement.
2.**Solution-Precipitation** – In areas where capillary pressures are high (particles are close together) atoms will preferentially go into solution and then precipitate in areas of lower chemical potential where particles are not close or in contact. This is called "contact flattening". This densifies the system in a way similar to grain boundary diffusion in solid state sintering. Ostwald ripening will also occur where smaller particles will go into solution preferentially and precipitate on larger particles leading to densification.
3. **Final Densification** – densification of solid skeletal network, liquid movement from efficiently packed regions into pores.

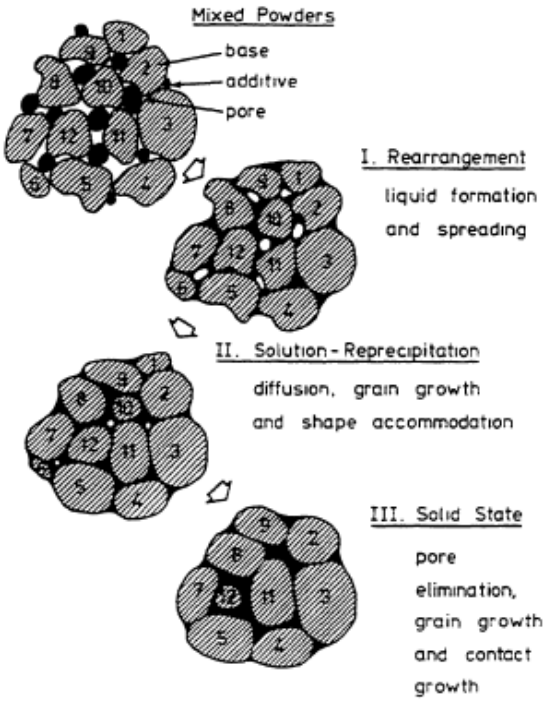

Figure 5: Classic Sequence of Stages

2.2.4 Densification, verification and grain growth

Sintering in practice is the control of both densification and grain growth. Densification is the act of reducing porosity in a sample thereby making it more dense. Grain growth is the process of grain boundary motion and Ostwald ripening to increase the average grain size. Many properties (mechanical strength, electrical breakdown strength, etc.) benefit from both a high relative density and a small grain size. Therefore, being able to control these properties during processing is of high technical importance. Since densification of powders requires high temperatures, grain growth naturally occurs during sintering. Reduction of this process is key for many engineering ceramics.

For densification to occur at a quick pace it is essential to have (1) an amount of liquid phase that is large in size, (2) a near complete solubility of the solid in the liquid, and (3) wetting of the solid by the liquid. The power behind the densification is derived from the capillary pressure of the liquid phase located between the fine solid particles. When the liquid phase wets the solid particles, each space between the particles becomes a capillary in which a substantial capillary pressure is developed.
Densification requires constant capillary pressure where just solution-precipitation material transfer would not produce densification. For further densification, additional particle movement while the particle undergoes grain-growth and grain-shape changes occurs. Shrinkage would result when the liquid slips between particles and increase pressure at points of contact causing the material to move away from the contact areas forcing particle centres to draw near each other.

The sintering of liquid-phase materials involves a fine-grained solid phase to create the needed capillary pressures proportional to its diameter and the liquid concentration must also create the required capillary pressure within range, else the process ceases. The vitrification rate is dependent upon the pore size, the viscosity and amount of liquid phase present leading to the viscosity of the overall composition, and the surface tension. Temperature dependence for densification controls the process because at higher temperatures viscosity decreases and increases liquid content. Therefore, when changes to the composition and processing are made, it will affect the vitrification process.

2.3 Selective Laser Sintering

Selective laser sintering (SLS) is an additive manufacturing (AM) technique that uses a laser as the power source to sinter powdered material (typically metal), aiming the laser automatically at points in space defined by a 3D model, binding the material together to create a solid structure.

SLS is a relatively new technology that so far has mainly been used for rapid prototyping and for low-volume production of component parts. Production roles are expanding as the commercialization of AM technology improves.

An additive manufacturing layer technology, SLS involves the use of a high power laser (for example, a carbon dioxide laser) to fuse small particles of plastic, metal, ceramic, or glass powders into a mass that has a desired three-dimensional shape. The laser selectively fuses powdered material by scanning cross-sections generated from a 3-D digital description of the part (for example from a CAD file or scan data) on the surface of a powder bed. After each cross-section is scanned, the powder bed is lowered by one layer thickness, a new layer of material is applied on top, and the process is repeated until the part is completed.

Because finished part density depends on peak laser power, rather than laser duration, a SLS machine typically uses a pulsed laser. The SLS machine preheats the bulk powder material in the powder bed somewhat below its melting point, to make it easier for the laser to raise the temperature of the selected regions the rest of the way to the melting point.

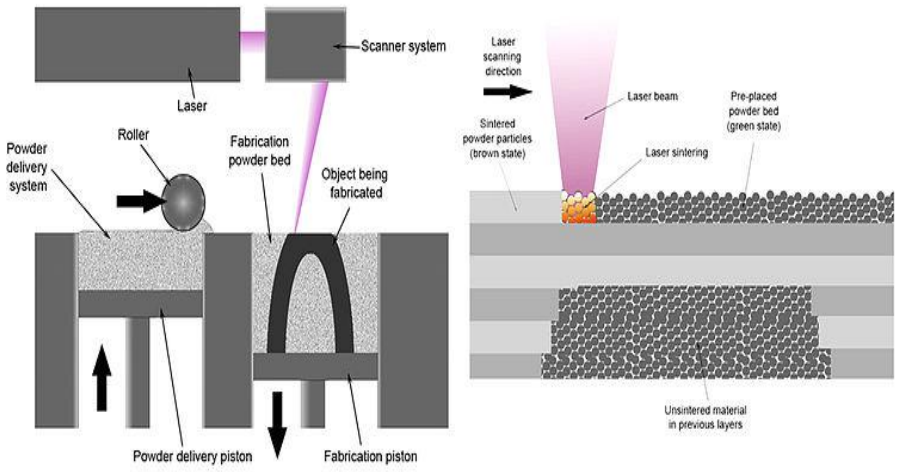

Figure 6: Selective Laser Sintering Process and Setup.

2.3.1 SLS mechanisms

There are two types of powder sintering mechanisms: **solid-phase sintering** and **liquid-phase sintering**. Solid-phase sintering is a thermal process, which occurs at temperatures below the melting point of the powders. In this case, the driving force for binding adjacent powder particles is a physical diffusion from one particle to another. Usually, the rate of diffusion is very low. Therefore, solid-phase sintering is a rather slow process. In its turn, liquid-phase sintering is much faster. As SLS is characterised by comparatively short time of laser radiation-powder interaction, it is too short to initiate solid-phase sintering. Thus, SLS can only be realised by the liquid-phase mechanism.

In case of single-component powders the liquid phase arises due to the surface melting of particles and the powder is sintered by joining the solid non-melted cores of particles.

In case of two-component powders, melting one of the components possessing the lower melting point forms the liquid phase; this liquid phase acts as binder (the binder powder in powder mixtures and the coating in coated powders). Usually, the heat applied to the system by the laser causes the binder to melt completely. However, in principle, both complete and surface melting of binder can take place. The powder is sintered by joining the solid particles of the main component having a higher melting point using the molten binder material

The formation of interparticle contacts is the most critical stage of the SLS process. The figure below shows the possible ways of contact formation during SLS of both single- and two-component powders (including powder mixtures and coated powders).

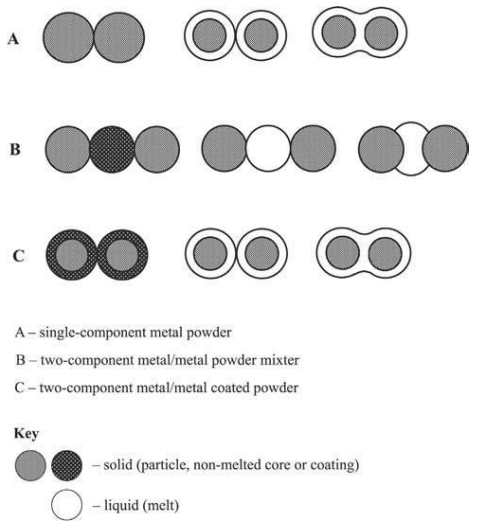

Figure 7: Liquid phase sintering of two component system

2.4 Probabilistic/Stochastic Methods

In probability theory and related fields, a stochastic or random process is a mathematical object usually defined as a collection of random variables. Historically, the random variables were associated with or indexed by a set of numbers, usually viewed as points in time, giving the interpretation of a stochastic process representing numerical values of some system randomly changing over time, such as the growth of a bacterial population, an electrical current fluctuating due to thermal noise, or the movement of a gas molecule. Stochastic processes are widely used as mathematical models of systems and phenomena that appear to vary in a random manner.

"Stochastic" means being or having a random variable. A stochastic model is a tool for estimating probability distributions of potential outcomes by allowing for random variation in one or more inputs over time. The random variation is usually based on fluctuations observed in historical data for a selected period using standard time-series techniques. Distributions of potential outcomes are derived from a large number of simulations (stochastic projections) which reflect the random variation in the input(s).

2.4.1 The Monte Carlo Method

Monte Carlo methods (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other approaches. Monte Carlo methods are mainly used in three distinct problem classes: optimization, numerical integration, and generating draws from a probability distribution.

In physics-related problems, Monte Carlo methods are useful for simulating systems with many coupled degrees of freedom, such as fluids, disordered materials, strongly coupled solids, and cellular structures (Potts model, interacting particle systems, McKean-Vlasov processes, kinetic models of gases). Other examples include modelling phenomena with significant uncertainty in inputs such as the calculation of risk in business and, in math, evaluation of multidimensional definite integrals with complicated boundary conditions. In

application to space and oil exploration problems, Monte Carlo–based predictions of failure, cost overruns and schedule overruns are routinely better than human intuition or alternative "soft" methods.

In principle, Monte Carlo methods can be used to solve any problem having a probabilistic interpretation. By the law of large numbers, integrals described by the expected value of some random variable can be approximated by taking the empirical mean (a.k.a. the sample mean) of independent samples of the variable. When the probability distribution of the variable is parametrized, mathematicians often use a Markov Chain Monte Carlo (MCMC) sampler. The central idea is to design a judicious Markov chain model with a prescribed stationary probability distribution. That is, in the limit, the samples being generated by the MCMC method will be samples from the desired (target) distribution.[7] By the ergodic theorem, the stationary distribution is approximated by the empirical measures of the random states of the MCMC sampler.

For example, consider a circle inscribed in a unit square. Given that the circle and the square have a ratio of areas that is $\pi/4$, the value of π can be approximated using a Monte Carlo method:

I. Draw a square, then inscribe a circle within it.
II. Uniformly scatter objects of uniform size over the square.
III. Count the number of objects inside the circle and the total number of objects.
IV. The ratio of the two counts is an estimate of the ratio of the two areas, which is $\pi/4$. Multiply the result by 4 to estimate π.

In this procedure the domain of inputs is the square that circumscribes our circle. We generate random inputs by scattering grains over the square then perform a computation on each input (test whether it falls within the circle). Finally, we aggregate the results to obtain our final result, the approximation of π.

There are two important points to consider here: Firstly, if the grains are not uniformly distributed, then our approximation will be poor. Secondly, there should be a large number of inputs. The approximation is generally poor if only a few grains are randomly dropped into the whole square. On average, the approximation improves as more grains are dropped.

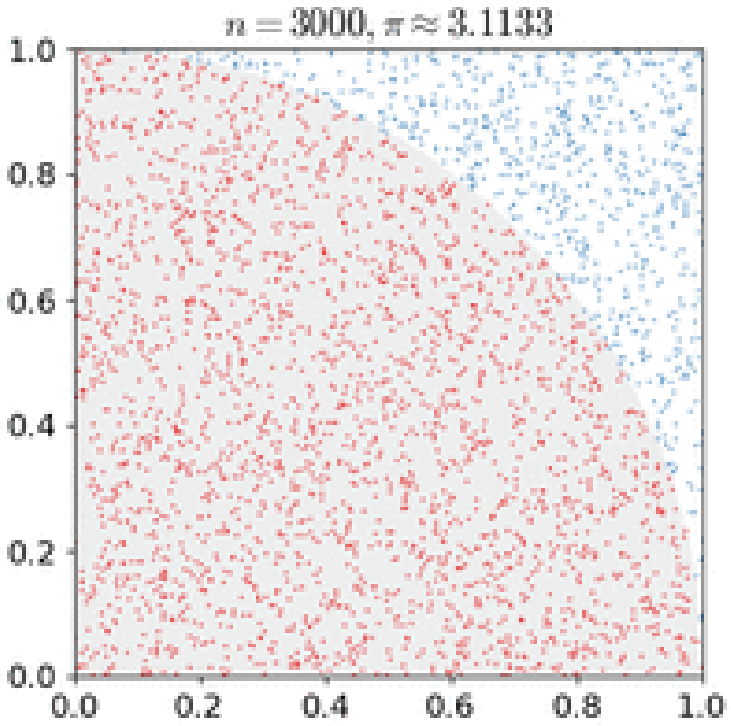

Figure 8: Estimation of the value of pi using Monte Carlo Method

2.4.2 Applications of MCS

Monte Carlo methods are especially useful for simulating phenomena with significant uncertainty in inputs and systems with a large number of coupled degrees of freedom. Areas of application include:

Physical sciences

Monte Carlo methods are very important in computational physics, physical chemistry, and related applied fields, and have diverse applications from complicated quantum chromodynamics calculations to designing heat shields and aerodynamic forms as well as in modelling radiation transport for radiation dosimetry calculations. In statistical physics Monte Carlo molecular modelling is an alternative to computational molecular dynamics, and Monte Carlo methods are used to compute statistical field theories of simple particle and polymer systems.

Engineering

Monte Carlo methods are widely used in engineering for sensitivity analysis and quantitative probabilistic analysis in process design. The need arises from the interactive, co-linear and non-linear behaviour of typical process simulations.

Other applications include:

- **Climate change and radiative forcing**

  The Intergovernmental Panel on Climate Change relies on Monte Carlo methods in probability density function analysis of radiative forcing.

- **Computational biology**

  Monte Carlo methods are used in various fields of computational biology, for example for Bayesian inference in phylogeny, or for studying biological systems such as genomes, proteins, or membranes. The systems can be studied in the coarse-grained or ab initio frameworks depending on the desired accuracy. Computer simulations allow us to monitor the local environment of a particular molecule to see if some chemical reaction is happening for instance.

- **Computer graphics**

  Path tracing, occasionally referred to as Monte Carlo ray tracing, renders a 3D scene by randomly tracing samples of possible light paths. Repeated sampling of any given pixel will eventually cause the average of the samples to converge on the correct solution of the rendering equation, making it one of the most physically accurate 3D graphics rendering methods in existence.

- **Applied statistics**
- **Artificial intelligence for games**
- **Finance and business**

2.5 Monte Carlo Simulations of Grain Growth in Polycrystalline Materials Using Potts Model

Computer simulation techniques have been developed, which can successfully incorporate many aspects of the grain interactions and can predict the main features of the microstructure. The aim of simulation of polycrystalline grain growth is to approximate to the highest degree to the real structures. Relations between Monte Carlo simulations and real structures have been studied. A procedure for the simulation and reconstruction of real structures in crystalline solids has been presented later.

The most realistic correspondence between the evolution of real and simulated structure was achieved by Monte Carlo simulations. Monte Carlo simulation is a stochastic Markov process that generates a sequence of configurations of lattice site states. Trial states are generated from a random distribution and are either accepted or rejected with a probability given by the **Boltzmann factor**.

An area element of microstructure is represented by one lattice site and is assigned a random number Qi (1< Qi < Q) called orientation or spin. Grain boundary lies between two adjacent sites with different orientation.

The energy of a lattice site is given by the **Hamiltonian**:

$$E = J \sum_{j=1}^{n} (1 - \delta_{Q_i Q_j})$$

where J is a positive constant, Qi is the orientation of the i th lattice site, Qj is the orientation of the j -th neighbouring lattice site, $\delta_{Q_i Q_j}$ is the Kronecker delta. The sum is given over n vicinal lattice sites. During the simulation procedure the i -th lattice site orientation is generated randomly and its energy E1 is calculated according to the equation given above. Then a new random orientation is given to the i -th lattice site and energy E2 after reorientation is again calculated. The reorientation is accepted when E2 < E1 . Otherwise the reorientation is accepted with the probability

$$P \approx \exp\{-\Delta E / kT\} ,$$

where

$$\Delta E = E2 - E1,$$

k is the Boltzmann constant and T is the temperature.

If the 2D lattice consists of N×N lattice sites, N×N reorientation attempts represent a time unit called Monte Carlo step (MCS). On the other hand for 3D simulation array

N×N×N reorientation attempts represent one MCS. In all simulation types described in the contribution the lattice sites can be arranged either in square or hexagonal configuration. The type of the simulation lattice is one of the input parameters before the simulation starts. Detailed working of this procedure is shown step-by-step in the following sections.

2.6 Simulation of Vacancy annihilation and Grain Growth in Liquid Phase Sintering and the Kawasaki Model

Grain growth simulated in the previous work followed Glauber Dynamics where the the total number of sites with a particular spin is not conserved. This means that the overall system can increase in the number of different orientations instead of staying constant. While this is suitable for the grain growth described above, the one seen in sintering processes requires a conserved model – the **Kawasaki Model**.

Kawasaki Dynamics

We consider the system to have only 2 orientations here - either zeroes or ones.

Let the initial configuration be the one shown below and choose the site with the state zero

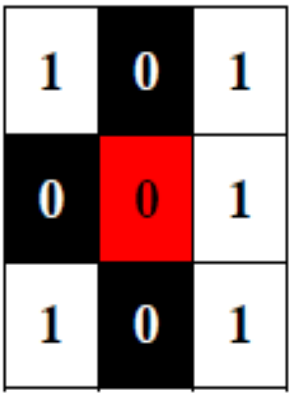

Figure 9: Kawasaki Model 1

In Kawasaki dynamics, the site is randomly swapped with a nearest neighbor. If the site is swapped with a spin of zero, then nothing happens because the energy of the system does not change and the both sites remain at zero. If the site is randomly swapped with another site containing a one, then the configuration looks like figure shown below.

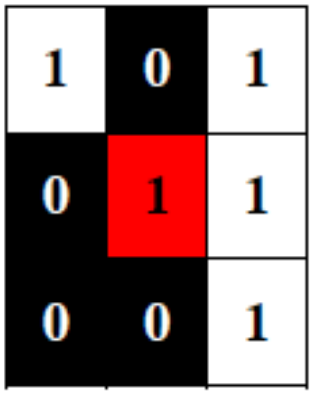

Figure 10: Kawasaki Model 2

The center site now has a value of one and the bottom left corner has a value of zero. The energy of the system (E) is given by the number of unlike neighbors for the center site and is given by:

$$E = \frac{1}{2}\sum_{i=0}^{N} * \sum_{j=0}^{n}(1 - \delta(q_i, q_2)) = \sum_{i=1}^{8} unlike\ neighbours$$

where N is the total number of sites,
n is the number of neighbors (8 for a two dimensional grid),
qi is the state of the current site,
qj is the state of the j -th neighbor site, and
δ is the Kronecker delta with $\delta(q_i = q_j) = 1$ and $\delta(q_i \neq q_j) = 0$

The above equation results in unlike neighbors contributing to the energy of the system, while like neighbors contribute no energy. The goal if KMC is to reduce the value of this expression. The change of energy in the system is calculated using the following formula:

$$\Delta E = E_{final} - E_{initial} \leq 0$$

For this example, the energy of the system went from 5 to 4. Therefore, the energy of the system is reduced and the centre site will flip to the new spin and remain a zero. The final configuration for the system will look like the last figure. The Kawasaki model conserves the total number of each kind of spin (e.g., the number of spins with values of zero and one remain constant). The new configuration has produced a larger grain size of ones on the right side of the box. The number of connected sites with values of one went from three to four. Even though the grains appeared to grow, the Kawasaki dynamics is more a method for simulating ordering processes rather than simulating grain growth.

2.6.1 Model for Vacancy Annihilation

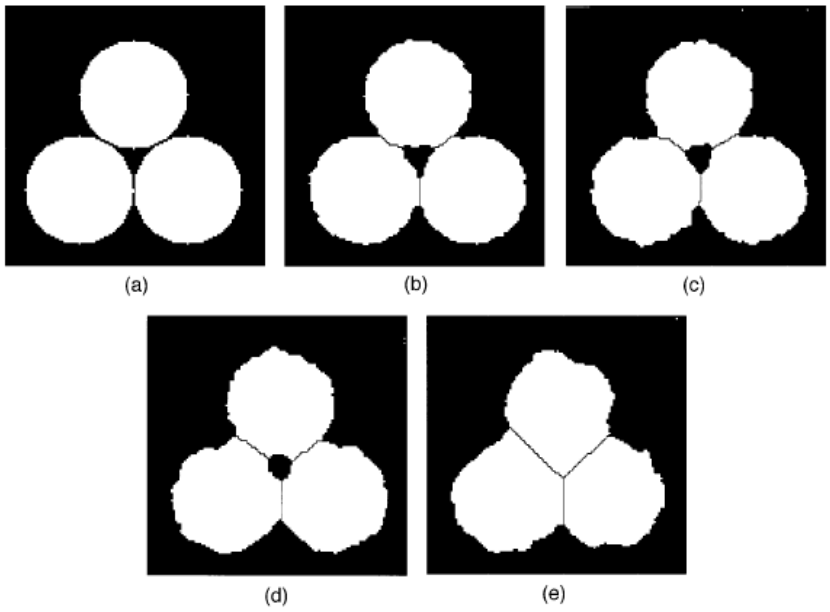

Figure 11: Microstructural evolution showing vacancy annihilation

Densification in crystalline solids occurs by uniform annihilation of vacancies at the grain boundaries. As vacancies are annihilated, the center of mass of the adjoining grain moves toward the grain boundary, thus giving densification.

In this model a vacancy is defined as a single, isolated pore site that is not connected to any other pore sites. The algorithm used for pore annihilation is the following. A pore site is chosen. If it happens to be a vacancy (an isolated pore site) on a grain boundary, it is annihilated. Annihilation is simulated as follows. A straight line is drawn from the isolated pore site through the center of mass of the adjacent grain to the outside boundary of that particle. Next, the isolated pore site and the outside grain site are exchanged with the grain site assuming the q state of the adjacent grain.

This algorithm conserves mass globally, moves the center of mass of the adjacent grain toward the annihilation site, and annihilates a vacancy.

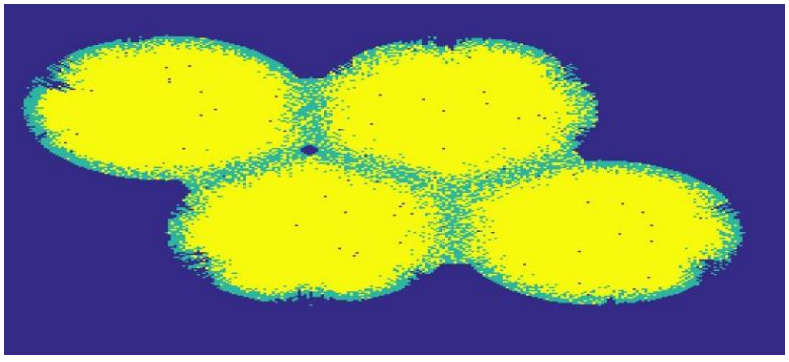

Figure 12: Pore stability

3. Algorithms adopted with relevant MATLAB program code along with the results of the simulation

3.1 Monte Carlo Simulation of Grain Growth in 2D

3.1.1 Generation of a matrix with random orientations

The function *randi* is used to generate a random integer between 1-10. The *n'th* order matrix is generated with zero padding making it an n+2 matrix.

Syntax:

```
function S = orientation_Matrix(n)
for i=1:n+2
    for j=1:n+2
        if i==1 || i==n+2 || j==1 || j==n+2
            S(i,j)=0;
        else
            S(i,j)= randi(10);
        end
    end
end
end
```

3.1.2 Calculating the energy at lattice points using the Hamiltonian

Syntax:

```
function E = energy_calculator(S)
E=0;
delta=0;
n=size(S,1);
for i=2:n-1
    for j=2:n-1
        if S(i,j)==S(i-1,j) | S(i-1,j)==0
          delta=1;
        else
                delta=0;
```

```
        end
        E=E+(1-delta);

        if S(i,j)==S(i+1,j) | S(i+1,j) ==0
            delta=1;
        else
            delta=0;
        end

        E=E+(1-delta);

        if S(i,j)==S(i,j-1) | S(i,j-1) ==0
              delta=1;
        else
            delta=0;
        end

        E=E+(1-delta);

        if S(i,j)==S(i,j+1) | S(i,j+1) ==0
            delta=1;
        else
            delta=0;
        end

        E=E+(1-delta);
    end
end
E = E/2;
end
```

3.1.3 Re-orientation matrix

A random *i*'th lattice is chosen and energy is calculated using the equation shown above. New random orientation given to the lattice site.

Energy of the lattice site calculated and compared with the previous energy.

If the new reorientation has lesser energy it is accepted, otherwise the previous orientation is retained.

Syntax:

```
function V = reorientation(V)
n=size(V,1);

i=randi([2 n-1]);
j=randi([2 n-1]);

        S=V;
        E1=energy_calculator(V);
        S(i,j)=randi(10);
        E2=energy_calculator(S);

        if E2<E1
            V=S;
        end
end
```

3.1.4 Working Example

Random generated matrix:

B =

0	**0**	**0**	**0**	**0**
0	**9**	**10**	**2**	**0**
0	**10**	**7**	**1**	**0**
0	**3**	**6**	**10**	**0**
0	**0**	**0**	**0**	**0**

Initial Energy = 12

Going through 10 iterations:

B =

0	0	0	0	0
0	9	10	2	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	2	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	2	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	2	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	10	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	10	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	10	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	10	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	0	0	0	0
0	9	10	10	0
0	10	7	1	0
0	10	6	10	0
0	0	0	0	0

B =

0	**0**	**0**	**0**	**0**
0	**9**	**10**	**10**	**0**
0	**10**	**7**	**1**	**0**
0	**10**	**6**	**10**	**0**
0	**0**	**0**	**0**	**0**

Final Energy: 10

Important observation: As the number of iterations increase the matrix reaches lower energies, i.e. the orientation at the lattice points becomes the same.

3.1.5 Implementing the Metropolis algorithm and updating the reorientation function.

The function when called returns the updated matrix V and the new value of energy at each iteration. A temperature of 1K is fixed.

The Metropolis algorithm is implemented by using *rand<exp(-(E2-E1)/temp).*

Syntax:

```
function [V,Energy] = reorientation(V)



n=size(V,1);

temp=1;



i=randi([2 n-1]);

j=randi([2 n-1]);



S=V;

E1=energy_calculator(V);

S(i,j)=randi(10);

E2=energy_calculator(S);



if E2<E1

   V=S;

else if rand<exp(-(E2-E1)/temp)

        V=S;

     end

end
```

```
Energy=energy_calculator(V);

end
```

```
for iter=1:10000

[B,E] = reorientation(A)

A=B;

X(iter,1)=E;

X(iter,2)=iter;

T=B(2:end,2:end);

if iter==1

    pcolor(T)

    hold on

    figure;

end

pcolor(T)

pause(0.005)

end

hold on

figure;



energy_iterationsPlot(X);
```

3.1.6 Plotting the energy vs Iterations graph

X axis is Energy. Y axis is iterations.

```
function energy_iterationsPlot(X)



plot(X(:,1),X(:,2),'-r','MarkerSize',10);

xlabel('Energy');

ylabel('Iteration number');



end
```

3.1.7 Working Example

Matrix Size N = 30

No. of orientations Q=2

No. of iterations iter=10000

Temperature T=1K

Initial Energy = 914

Initial Orientation

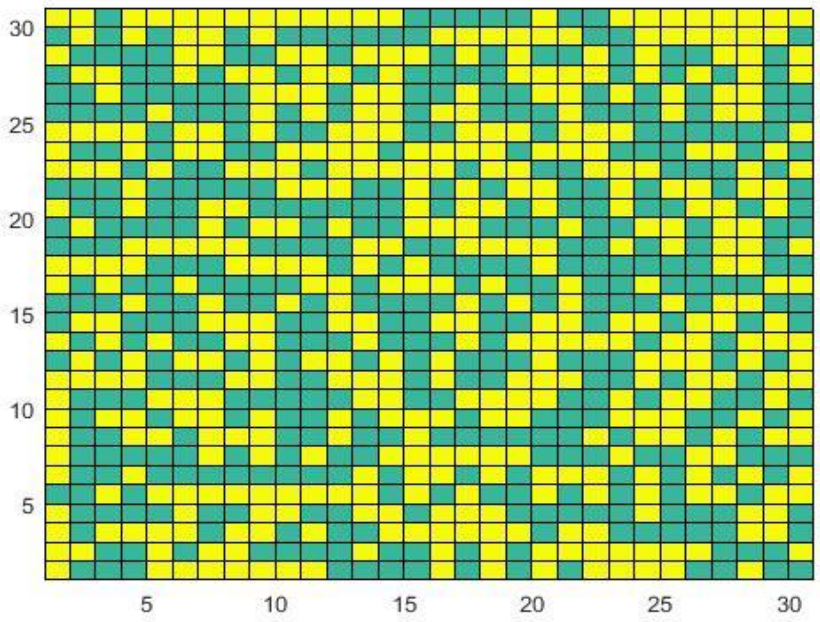

Figure 13: Initial 2D Matrix

Final Energy= 360

Final Orientation

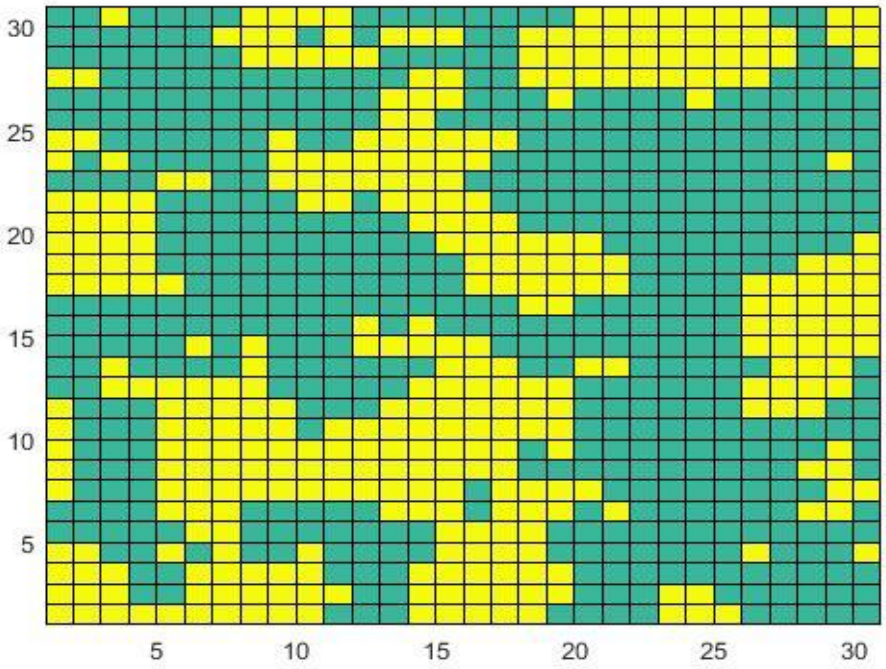

Figure 14: Final 2D Matrix

Energy vs Monte Carlo Steps

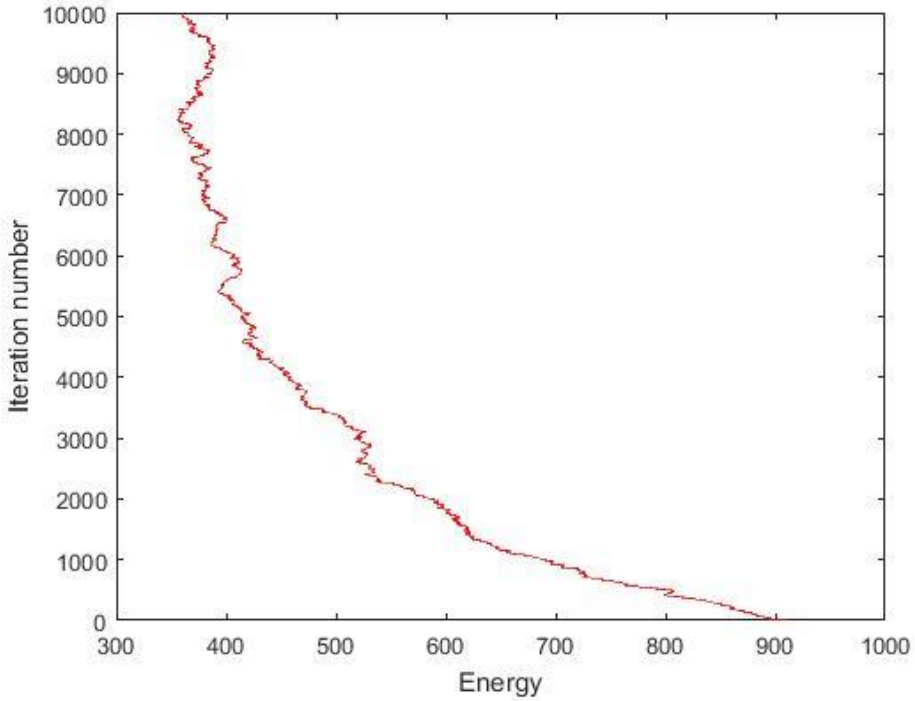

Figure 15: Energy vs MCS Steps 2D

3.2 Monte Carlo Simulation of Grain Growth in 3D

Introducing the 3rd dimension k and using the same logic everywhere.

3.2.1 Orientation Matrix for 3D

Syntax:

```
function S = orientation_3D_Matrix(n)

for i=1:n+2

    for j=1:n+2

        for k=1:n+2

        if i==1 || i==n+2 || j==1 || j==n+2 || k==1 ||k==n+2

            S(i,j,k)=0;

        else

            S(i,j,k)= randi(2);

        end

    end

end

end

end
```

3.2.2 Energy Calculation for 3D

Modifying the code to move in one direction only hence reducing computation.

Syntax:

```
function E = energy_calculator_3D(S)

E=0;

delta=0;

n=size(S,1);
```

```
for i=2:n-1

    for j=2:n-1

        for k=2:n-1



            if S(i,j,k)==S(i+1,j,k) | S(i+1,j,k) ==0

                delta=1;

            else

                delta=0;

            end



            E=E+(1-delta);



            if S(i,j,k)==S(i,j+1,k) | S(i,j+1,k) ==0

                delta=1;

            else

                delta=0;

            end



            E=E+(1-delta);



            if S(i,j,k)==S(i,j,k+1) | S(i,j,k+1) ==0

                delta=1;

            else

                delta=0;

            end
```

```
            E=E+(1-delta);

        end

    end

end

end
```

3.2.3 Reorientation for 3D

Syntax:

```
function [V,Energy] = reorientation_3D(V)



n=size(V,1);

temp=1 ;



i=randi([2 n-1]);

j=randi([2 n-1]);

k=randi([2 n-1]);



S=V;

E1=energy_calculator_3D(V);

S(i,j,k)=randi(2);

E2=energy_calculator_3D(S);



if E2<E1

   V=S;

else if rand<exp(-(E2-E1)/temp) % Implementing metropolis algorithm
```

```
        V=S;

    end

end

Energy=energy_calculator_3D(V);

end
```

3.2.4 Main Function

Syntax:

```
clear all; clc;



A = orientation_3D_Matrix(5);

C=A;

n=size(A,1);

initia_energy=energy_calculator_3D(A);



for iter=1:100

[B,E] = reorientation_3D(A)

A=B;

X(iter,1)=E;

X(iter,2)=iter;

end



energy_iterationsPlot(X);



final_energy=energy_calculator_3D(B);
```

3.2.5 Plotting in 3D

```
      function plot_3D(C,ColorMatrix,N,Q)
      %----------------------
      x=zeros(N,N,N);y=x;z=x;
      X=0:1/(N-1):1;
      Y=X;
      [xt,yt]=meshgrid(X,Y);
      x=repmat(xt,[1 1 N]);
      y=repmat(yt,[1 1 N]);
      for count=1:(N)
          z(:,:,count)=(count-1)*ones(N,N)/(N-1);
      end
      %-----------------------------------------
      [xyzQ] = Q_position(C,x,y,z,Q);

      SqSz = 1.00*10/(N/25);

          for q = 1:Q
              if prod(size(xyzQ{1,q}))~=0

plot3(xyzQ{1,q}(:,1),xyzQ{1,q}(:,2),xyzQ{1,q}(:,3),'s','MarkerFaceColo
r',...

[ColorMatrix(q,:)],'MarkerEdgeColor','k','MarkerSize',SqSz)
                      axis square,hold on;
                  end
                  xlabel('X - axis'),ylabel('Y - axis'),zlabel('Z -
axis')
      %             axesLabelsAlign3D,title('Grain Structure')

      end
      end
```

3.2.6 Working Example

Initial Matrix with 2 orientations

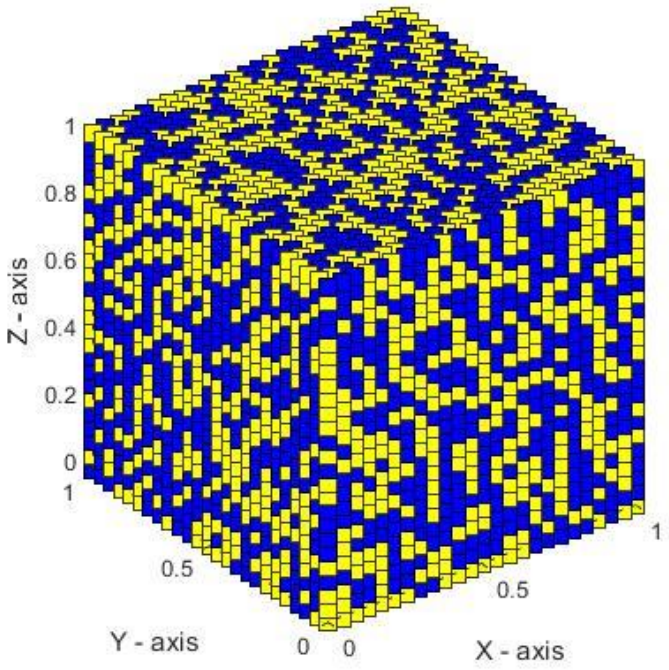

Figure 16: Initial 3D Matrix

Final Matrix after applying the algorithm

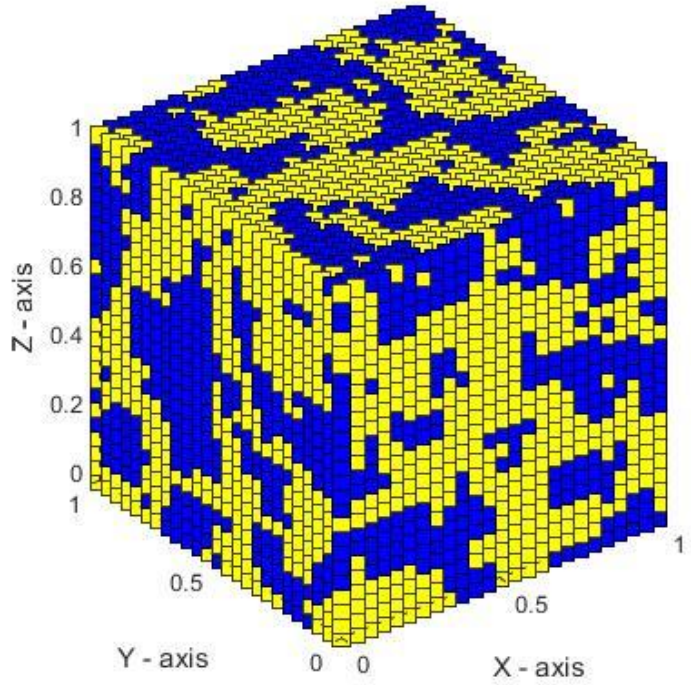

Figure 17: Final 3D Matrix

Energy vs Number of Iterations

Matrix Size N = 25

No. of orientations Q=2

No. of iterations =100,000

Temperature T=1K

J=1

Ignoring k

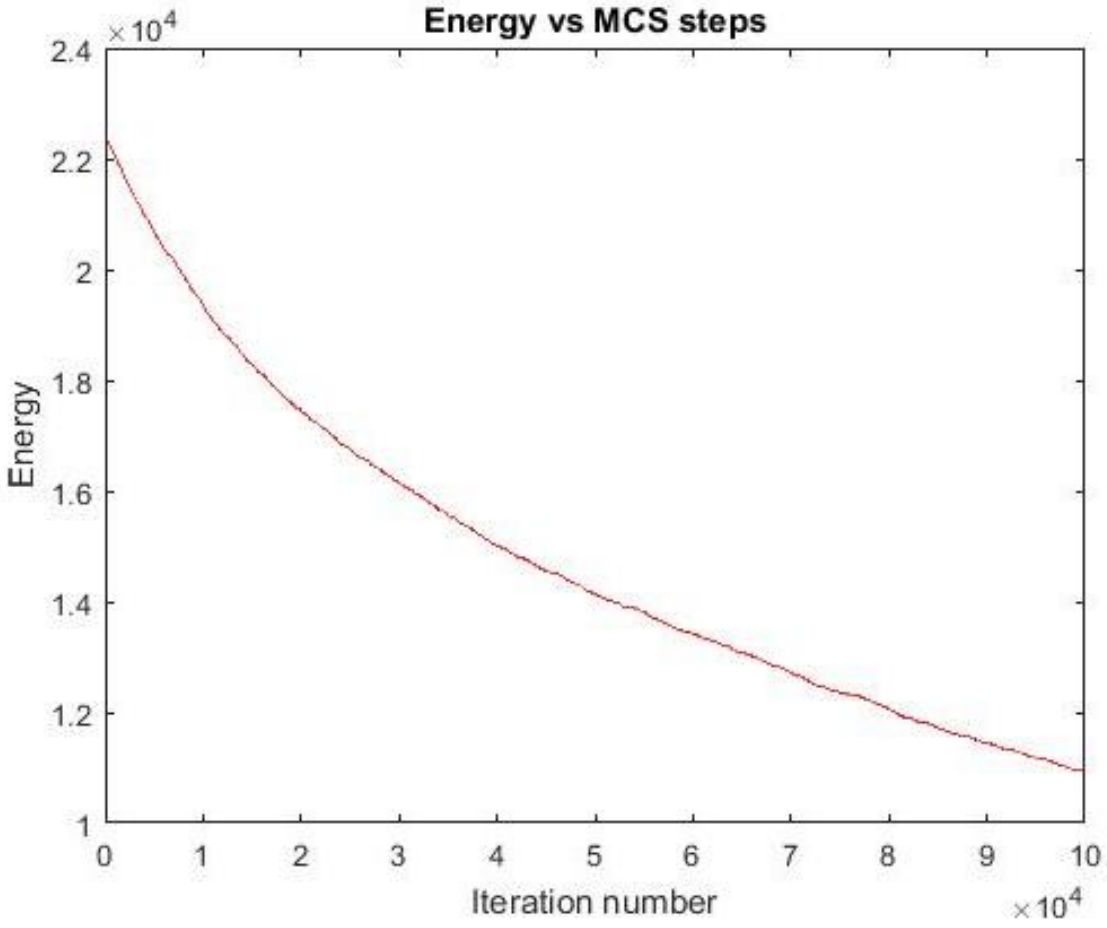

Figure 18: Energy vs MCS Steps 3D

3.3 Analytical Approach vs Monte Carlo Simulations

Analytical approaches use the following equations to calculate the average of any property of a macro system (i.e Energy).

$$< O >= \frac{\sum_{i=1}^{N} O_i e^{-E_i/T}}{\sum_{i=1}^{N} e^{-E_i/T}}$$

The Monte Carlo simulation runs the following equation to calculate the variation of energy with temperature.

$$< E > \ = \frac{\sum_{i=1}^{N} E_i}{N}$$

3.3.1 Configuration Matrix

```
N=3;

Q=2;

n=Q^(N^2);

for i=0:n-1

    temp=i;

    for j=1:(N^2)

    A((n)-i,j)=(mod(temp,Q)+1);

    temp=floor(temp/Q);

    end

end

for i=1:n

    L=1;

    for j=1:N

        for k=1:N

            S{i,1}(j,k)=A(i,j);

            L=L+1;
```

```
        end

    end

end

for i=1:n

    E(i,1)=energy_calculator_new(S{i,1});

end

 analytical_appro(E,n)
```

3.3.2 Energy Calculation

```
N=3; Q=2; n=Q^(N^2);

for i=0:n-1

    temp=i;

    for j=1:(N^2)

    A((n)-i,j)=(mod(temp,Q)+1);

    temp=floor(temp/Q);

    end

end

for i=1:n

    L=1;

    for j=1:N

        for k=1:N

            S{i,1}(j,k)=A(i,j);

            L=L+1;

        end

    end

end
```

```
for i=1:n
    E(i,1)=energy_calculator_new(S{i,1});
end
analytical_appro(E,n)
```

3.3.3 Analytical Approach Alogorithm

```
function analytical_appro(E,n)
T=(0:0.25:10)';
for i = 1:size(T,1)
    Num_total=0;
    Den_total=0;
    for j=1:n
        Num_total=Num_total+E(j,1)*exp(-E(j,1)/T(i,1));
        Den_total=Den_total+exp(-E(j,1)/T(i,1));
    end
    E_average_A(i,1)=Num_total/Den_total;
end

Num_total2 = 0;
for i = 1:n
    Num_total2 = Num_total2 + E(i,1);
end
E_average_MC = Num_total2/n;
```

```
E_average_A(1,1)=0;

plot(T,E_average_A)

xlabel('Temperature');

ylabel('Average energy');

end
```

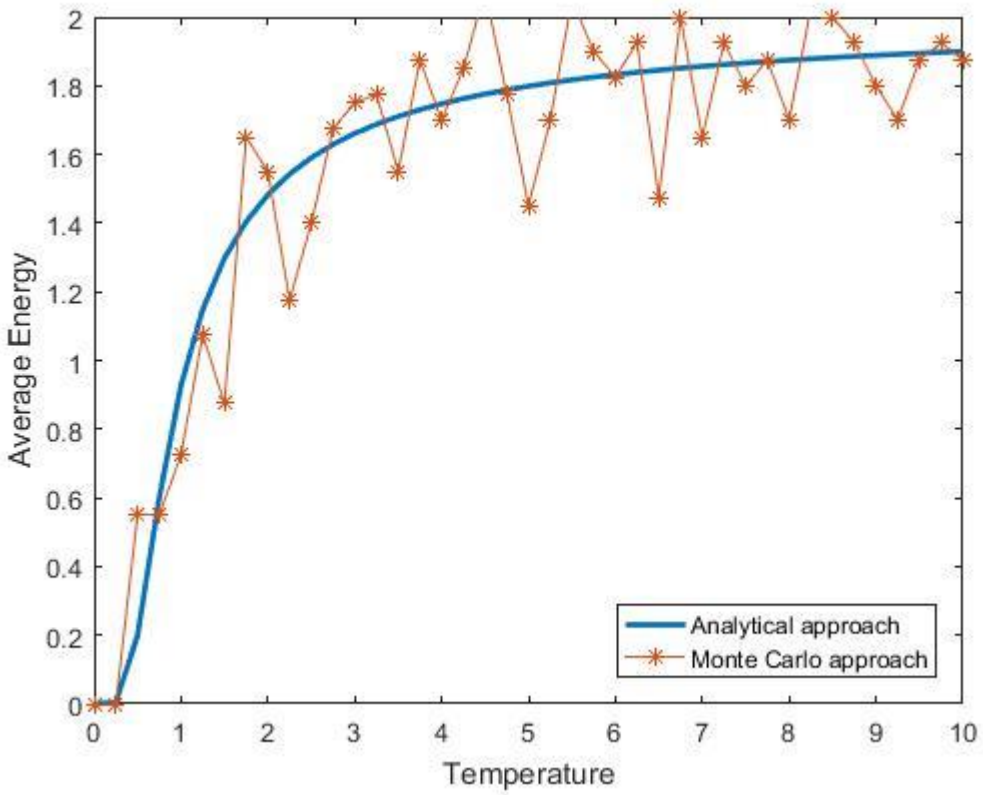

Figure 19: Analytical vs MCS 100 steps

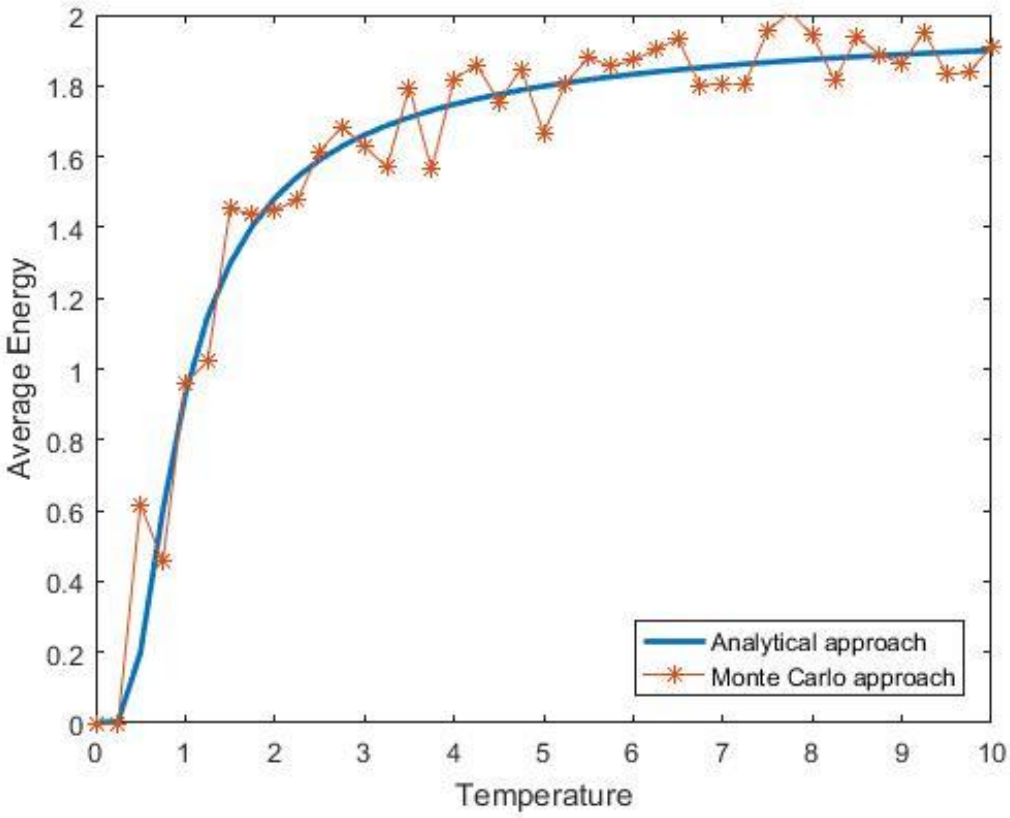

Figure 20: Analytical vs MCS 1000 steps

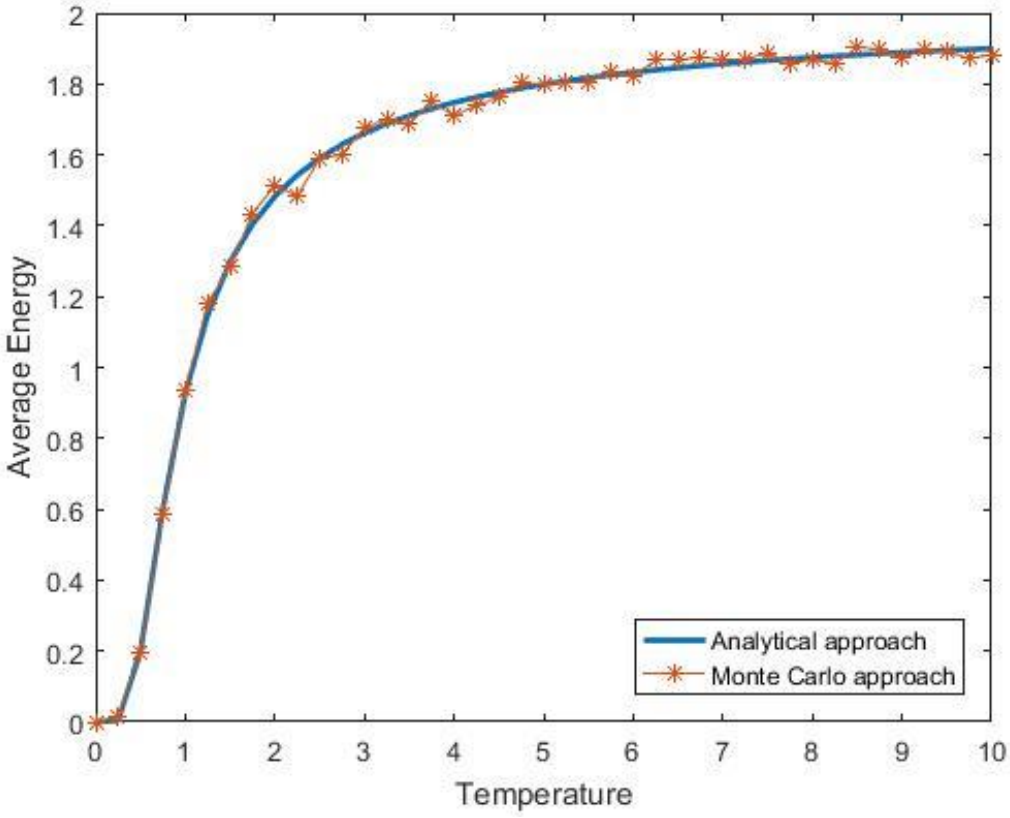

Figure 21: Analytical vs MCS 10000 steps

3.4 Monte Carlo Simulation of Sintering

3.4.1 Plotting of the sintering model

```
---- This is the MAIN function ----

clc;

clear all;

Plotting Matrix

sets=3;

r=10;



A=plotting(sets,r);

A = RectangleInitial_Matrix()

A = Initial_Matrix();



B=A;

E1=energy_calculator(A);



pcolor(A)

shading flat

figure



  Annihilation

 c=1;

[i,j]=find(A==0);

i=i';

j=j';
```

```
pores=[i;j]';

iter=size(pores,1);

for i=1:(0.9*iter)

    [vacancy,c]= Extract_vacancy(A,c);

    [vacancy]= Extract_vacancy(A);

    if ~(isempty(vacancy))

        A=annihilation(A,vacancy);

    end

           pcolor(A)

             pause(0.7)

end

C=A;

E2=energy_calculator(A);

pcolor(A)

shading flat



figure



Making liquids and solids same orientation



[i,j]=find(A>1000);%solids

core=horzcat(i,j);

for k=1:size(core)

    A(core(k,1),core(k,2))=1001;%101

end



[i,j]=find(A<1000);%liquids
```

```
liquid_free=horzcat(i,j);

[i,j]=find(A>0);

liquid_core=horzcat(i,j);

liquid=intersect(liquid_free,liquid_core,'rows');



for k=1:size(liquid)

    A(liquid(k,1),liquid(k,2))=1;

end



D=A;

E3=energy_calculator(A);

Conserved Grain Growth



 iter1=1000000;



 for i=1:iter1

    A=reorientation(A);

 end

 E=A;

 E4=energy_calculator(A);

pcolor(A)

 shading flat

solid reorietation



 [i,j]=find(A==1001);%solids

 core=horzcat(i,j);

 for k=1:size(core)
```

```
    A(core(k,1),core(k,2))=2;
end


numb=(sets*2)^2;


for k=1:size(core,1)
    A(core(k,1),core(k,2))=randi(numb)+1;
end


F=A;
E5=energy_calculator(A);
figure
pcolor(A)
shading flat
iter2=20000000;


for i=1:iter2
    point=datasample(core,1);
    A=solid_reorientation(A,point,numb);
end


E6=energy_calculator(A);
figure
pcolor(A)
shading flat
```

3.4.2 Plotting Function

```
function A=plotting(sets,r)



N=(sets*4*r)+(3*r);

A=ones(N);

A=A*-1;

[c,d]=centers(sets,N,r);




for k=1:size(d,1)

    di=d(k,1);

    dj=d(k,2);

    for i=1:N

        for j=1:N

            if ((i-di)^2+(j-dj)^2<=(r)^2)

                A(i,j)=0; % Pore

            end

        end

    end

end



for k=1:size(c,1)

    ci=c(k,1);

    cj=c(k,2);

    for i=1:N

        for j=1:N

            if ((i-ci)^2+(j-cj)^2<=(r)^2)
```

```
                A(i,j)=k; % Liquid

            end

        end

    end

end



for k=1:size(c,1)

    ci=c(k,1);

    cj=c(k,2);

    for i=1:N

        for j=1:N

            if ((i-ci)^2+(j-cj)^2<=(9*r/10)^2)

                A(i,j)=k+1000;%Solids

            end

        end

    end

end

% pcolor(A)

end
```

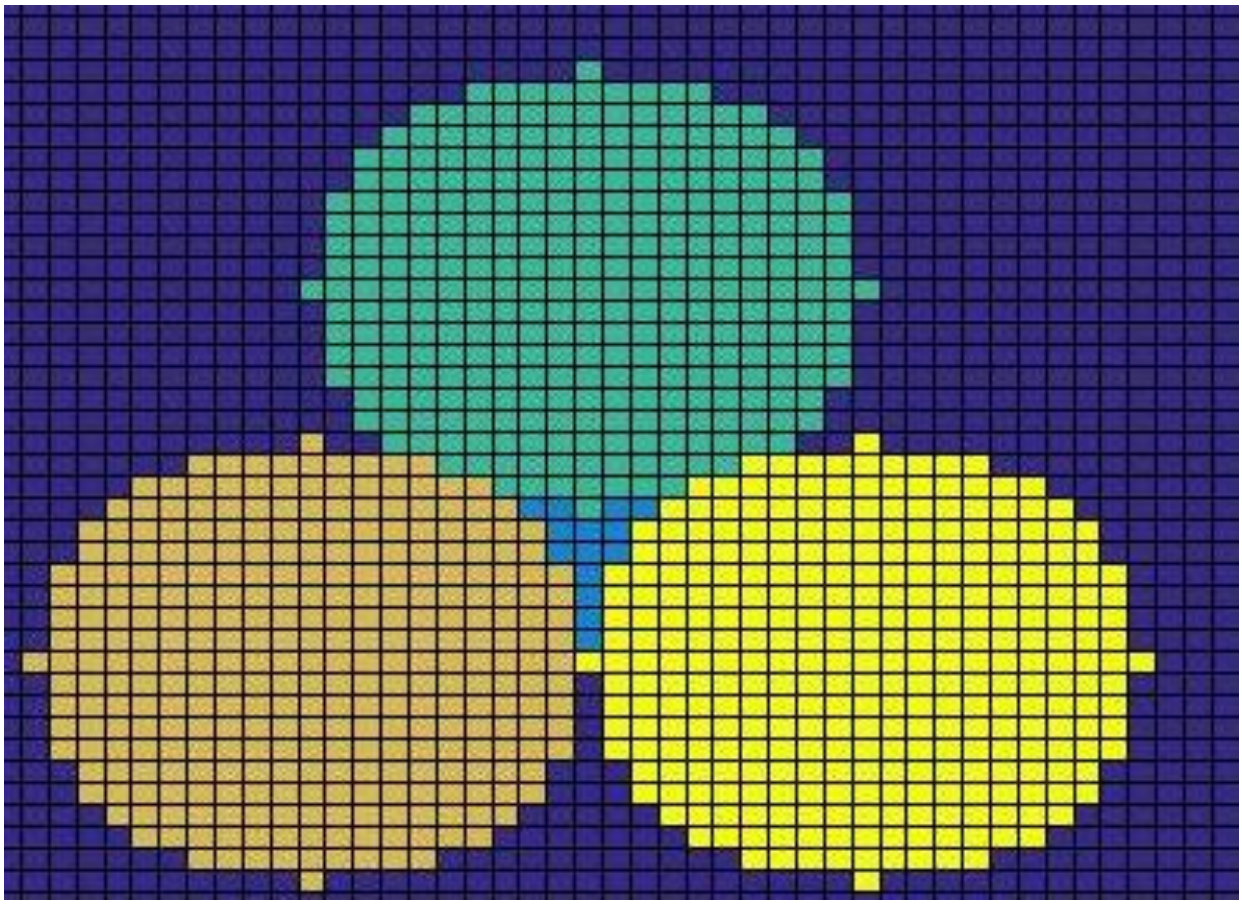

Figure 22: 3 Particle system

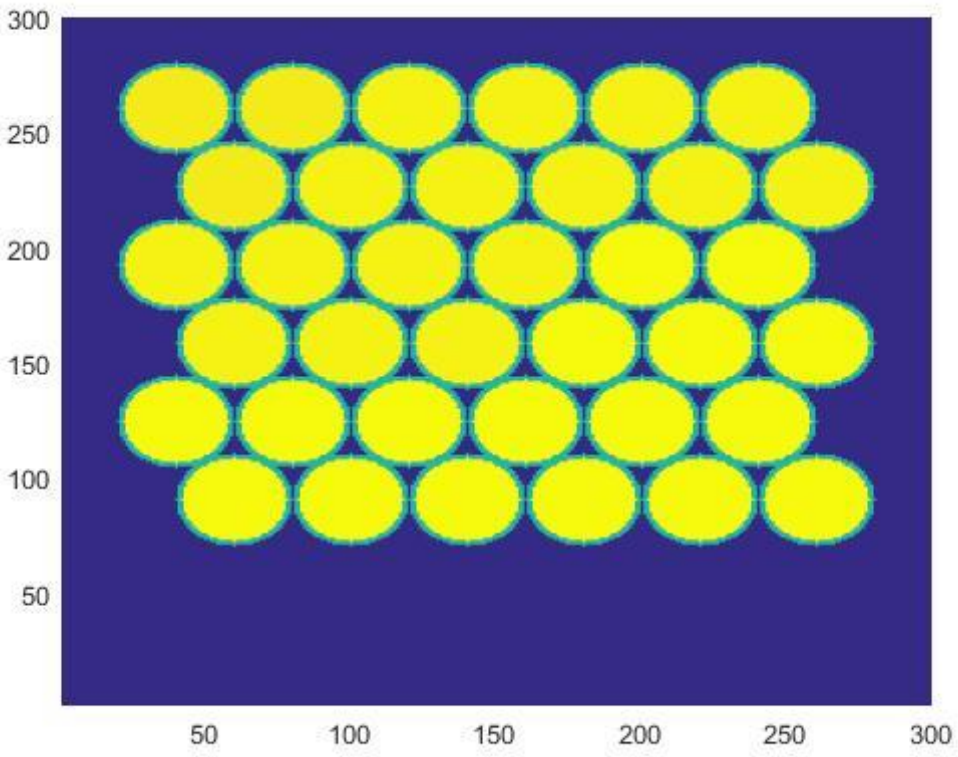

Figure 23: Hexagonal Packing

3.4.3 Simulation of annihilation of vacancies

```
function [c,d]=centers(sets,N,r)



total = (sets*2)^2;

n = total/sets;



c(1,1)=N-(2*r);

c(1,2)=2*r;



for i=2:total



    if mod(i-1,n) == 0

     c(i,1)=c(i-n+1,1)-floor(sqrt(3)*r);

     c(i,2)=c(i-n+1,2)-r;



    else

    if mod(i,2)==0

        c(i,1)=c(i-1,1)-floor(sqrt(3)*r);

        c(i,2)=c(i-1,2)+r;

    else

        c(i,1)=c(i-2,1);

        c(i,2)=c(i-2,2)+(2*r);

    end

    end



end
```

```
j=1;
for i=1:total
    if mod(i,n) ~= 0 && mod(i+1,n) ~= 0
    d(j,1)=floor((c(i,1)+c(i+1,1)+c(i+2,1))/3);
    d(j,2)=floor((c(i,2)+c(i+1,2)+c(i+2,2))/3);
    j=j+1;
    end

    if mod(i,2)==0 && mod(i,n) ~= 0 && i < (total - n)
    d(j,1)=floor((c(i,1)+c(i+(n-1),1)+c(i+(n+1),1))/3);
    d(j,2)=floor((c(i,2)+c(i+(n-1),2)+c(i+(n+1),2))/3);
    j=j+1;
    d(j,1)=floor((c(i,1)+c(i+(2),1)+c(i+(n+1),1))/3);
    d(j,2)=floor((c(i,2)+c(i+(2),2)+c(i+(n+1),2))/3);
    j=j+1;
    end
end
end
```

```
% function [vacancy,c]= Extract_vacancy(A,c)
function [vacancy]= Extract_vacancy(A)
[i,j]=find(A==0);
i=i';j=j';

pores=[i;j]'; %--pores--
vacancy_list=[];
```

```
k=1;

for count=1:size(pores,1) %-- Vacancys--

    check(1,1)=pores(count,1);

    check(1,2)=pores(count,2);

    [orientation,r]=orientation_movement(A,check);

    if r==1

        vacancy_list(k,1)=pores(count,1);

        vacancy_list(k,2)=pores(count,2);

        k=k+1;

    end

end



if size(vacancy_list,1)==0

    vacancy=[];

elseif size(vacancy_list,1)==1

    vacancy=vacancy_list;

    %     c=0;

else

    vacancy = datasample(vacancy_list,1);

end

end
```

```
function A=annihilation(A,vacancy)

orientation=orientation_movement(A,vacancy);

[i,j]=find(A==orientation);

[k,l]=find(A==(orientation+1000));

[o,p]=find(A==(orientation-1000));
```

```
m=vertcat(i,k,o);

n=vertcat(j,l,p);

centroid= [floor(mean(m)),floor(mean(n))];

c1=centroid(1);c2=centroid(2);% Centroid points

p1=vacancy(1,1);p2=vacancy(1,2);% vacancy points

N=size(A,1);

Y=line_points(N,c1,c2,p1,p2);

W=path_points(A,Y,c1,c2,p1,p2);

for e=1:(size(W,1)-1) % ANNIHILATION

    if A(W(e,1),W(e,2))~= A(W(e+1,1),W(e+1,2))

        A(W(e,1),W(e,2))= A(W(e+1,1),W(e+1,2));

    end

end

end
```

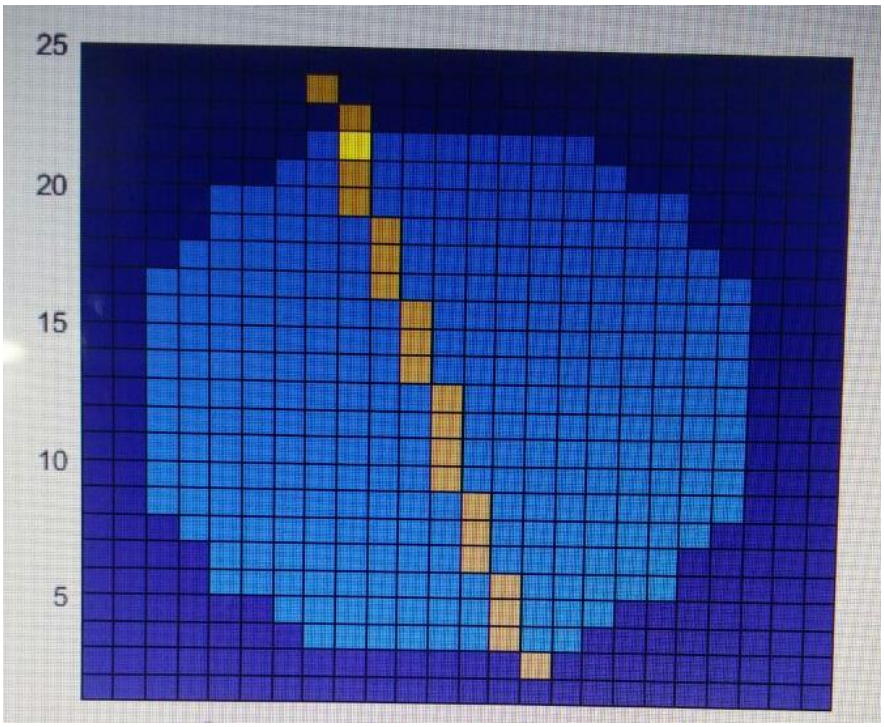

Figure 24: Annihilation

3.4.4 Simplified process of vacancy annihilation

Functions to incorporate the above in the hexagonal close packed particle system

```
function [orientation,r] = orientation_movement(A,vacancy)



v1=vacancy(1,1);

v2=vacancy(1,2);



a(1,1)=v1-1;

a(2,1)=v1-1;

a(3,1)=v1-1;

a(4,1)=v1;

a(5,1)=v1;

a(6,1)=v1+1;

a(7,1)=v1+1;

a(8,1)=v1+1;



a(1,2)=v2-1;

a(2,2)=v2;

a(3,2)=v2+1;

a(4,2)=v2-1;

a(5,2)=v2+1;

a(6,2)=v2-1;

a(7,2)=v2;

a(8,2)=v2+1;



M=[];
```

```
orientation=[];

k=1;

for i=1:8

    if A(a(i,1),a(i,2))>0

        M(k,1)=a(i,1);

        M(k,2)=a(i,2);

        k=k+1;

    end

end



if isempty(M)

    r=0;

else

    r=1;

    if size(M,1)==1

        sample=M;

    else

        sample = datasample(M,1);

    end



    orientation=A(sample(1,1),sample(1,2));

end



if size(M,1)==1

     sample=M;

 else

     sample = datasample(M,1);
```

```
 end


 orientation=A(sample(1,1),sample(1,2));


 s=size(A);
 N=length(s);
 [c1{1:N}]=ndgrid(1:3);
 c2(1:N)={2};
 offsets=sub2ind(s,c1{:}) - sub2ind(s,c2{:});
 linearInd = sub2ind(size(A),vacancy(1,1), vacancy(1,2));
 neighbors = A(linearInd+offsets);
 sample = datasample(find(neighbors~=0),1);
 orientation=neighbors(sample);


end
```

```
function W=path_points(A,Y,c1,c2,p1,p2)
j=1;
for i=1:size(Y,1)
    if Y(i,1)==p1 && Y(i,2)==p2
        a=i;
    end
end


if (p1+p2)<(c1+c2)
    for i=a:size(A,1)
        if A(Y(i,1),Y(i,2))==-1
```

```
            W(j,1)=Y(i,1);

            W(j,2)=Y(i,2);

            break;

        end

        W(j,1)=Y(i,1);

        W(j,2)=Y(i,2);

        j=j+1;

    end

else

    for i=a:-1:1

        if A(Y(i,1),Y(i,2))==-1

            W(j,1)=Y(i,1);

            W(j,2)=Y(i,2);

            break;

        end

        W(j,1)=Y(i,1);

        W(j,2)=Y(i,2);

        j=j+1;

    end

end

end
```

```
function Y=line_points(N,c1,c2,p1,p2)



i=1;

Y1=0;

for x=1:N
```

```
    y=floor(c2+((c2-p2)/(c1-p1))*(x-c1));
    if y>0&& y<N
        Y1(i,1)=x;
        Y1(i,2)=y;
        i=i+1;
    end
end


j=1;
Y2=0;
for y=1:N
    x=floor(c1+((c1-p1)/(c2-p2))*(y-c2));
    if x>0 && x<N
        Y2(j,1)=x;
        Y2(j,2)=y;
        j=j+1;
    end
end


if size(Y1,1)>size(Y2,1)
    Y=Y1;
else
    Y=Y2;
end
end
```

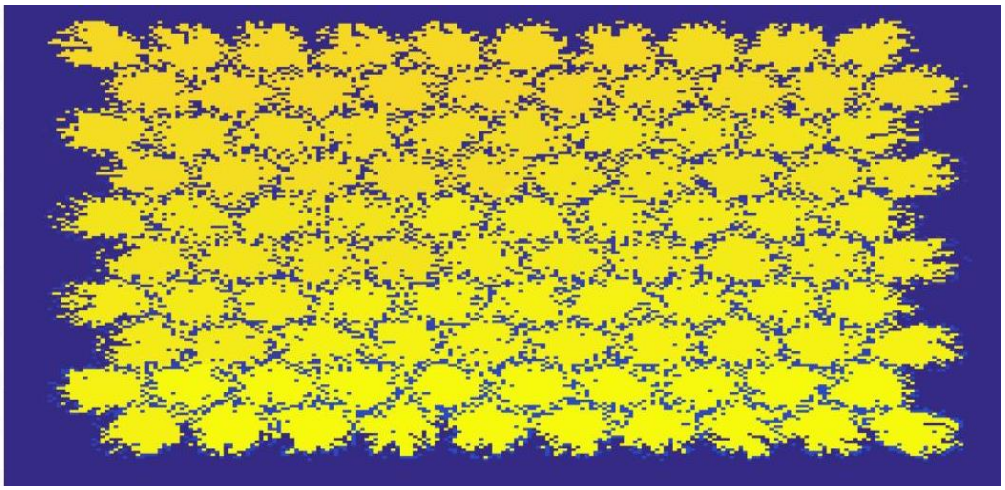

Figure 25: Post Annihilation

3.4.5 Incorporating Solution– Reprecipitation to include necking

```
function A = reorientation(A)



[i,j]=find(A==1001);

core=horzcat(i,j);

point=datasample(core,1);



E1=Pointenergy_calculator(A,point(1,1),point(1,2));

% E1=Pointenergy_cal_reori(A,point(1,1),point(1,2));

a=point(1,1);

b=point(1,2);

B= swapping(A,a,b);



E2=Pointenergy_calculator(B,a,b);

% E2=Pointenergy_cal_reori(B,a,b);



if E2<=E1
```

```
   A=B;

else if rand<exp(-(E2-E1)/0.7) % Implementing metropolis algorithm

        A=B;

    end

end



end
```

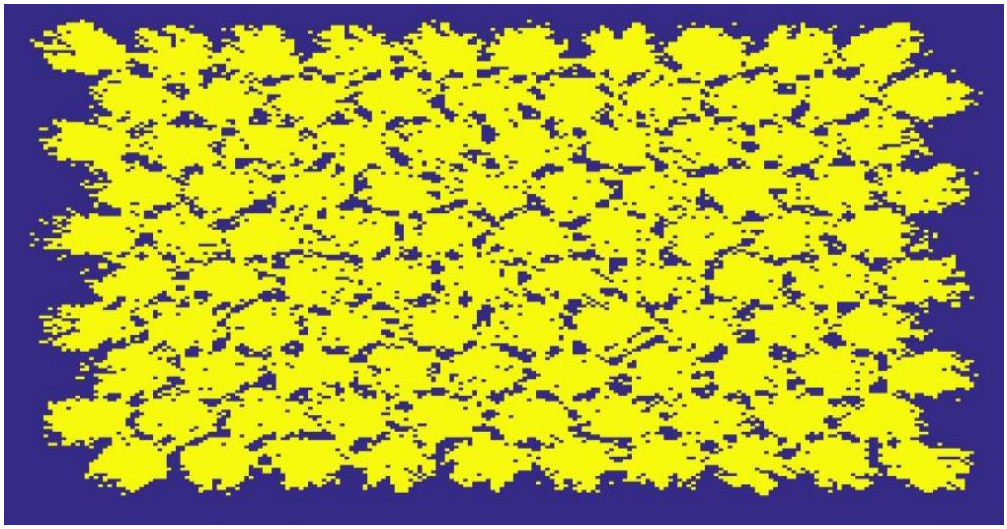

Figure 26: Post solution Reprecipitation

3.4.6 Assigning grain orientations to particles and observing the subsequent evolution

```
function A = swapping(A,pi,pj)

neighbours=[];

k=1;

for i=pi-1:pi+1

    for j=pj-1:pj+1

        if i==pi && j==pj

            continue

        end

        if A(i,j)~=-1

        neighbours(k,1)=A(i,j);

        neighbours(k,2)=i;

        neighbours(k,3)=j;

        k=k+1;

        end

    end

end

if ~(isempty(neighbours))

pos=randi(size(neighbours,1));

% if neighbours(pos,1)~=-1

    S=A(pi,pj);

    A(pi,pj)=A(neighbours(pos,2),neighbours(pos,3));

    A(neighbours(pos,2),neighbours(pos,3))=S;

% end

end
```

```
function A = solid_reorientation(A,point,numb)


E1=Pointenergy_calculator(A,point(1,1),point(1,2));


B=A;


B(point(1,1),point(1,2))=randi(numb)+1;


E2=Pointenergy_calculator(B,point(1,1),point(1,2));


if E2<=E1
   A=B;
else if rand<exp(-(E2-E1)/0.1) % Implementing metropolis algorithm
        A=B;
    end
end


end
```

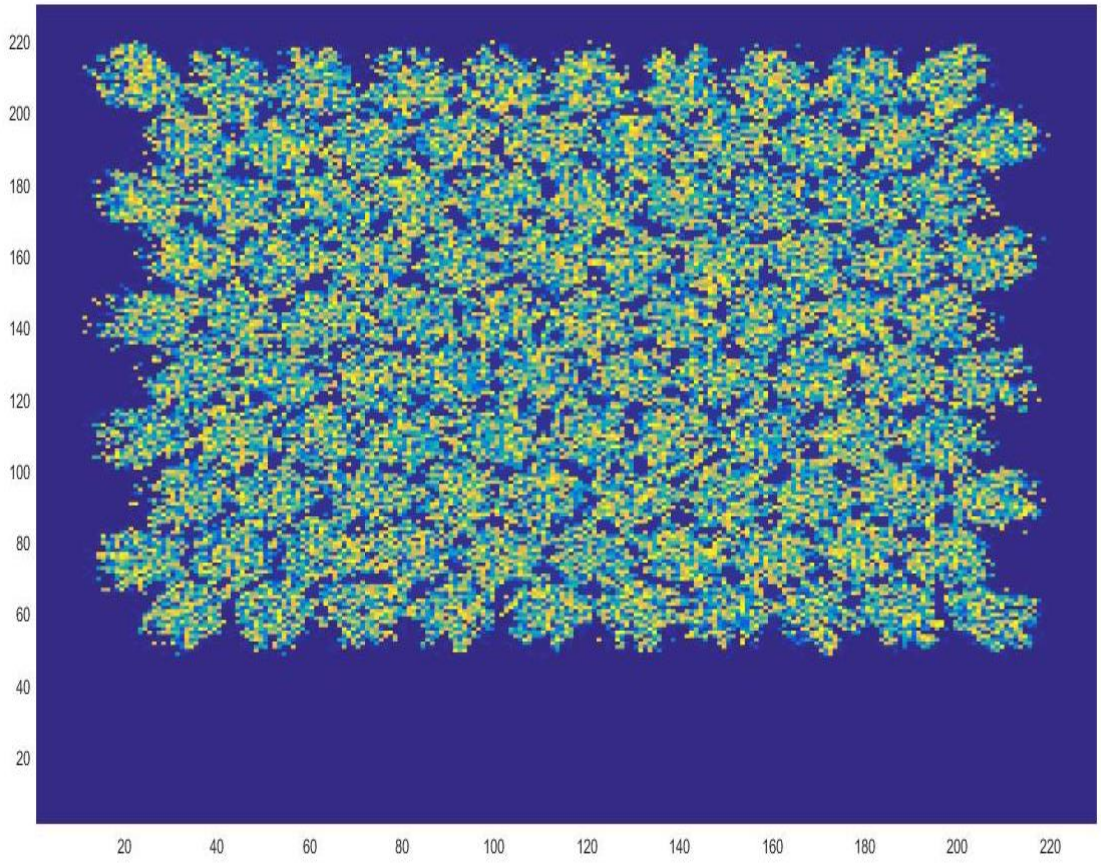

Figure 27: Prior Grain Growth

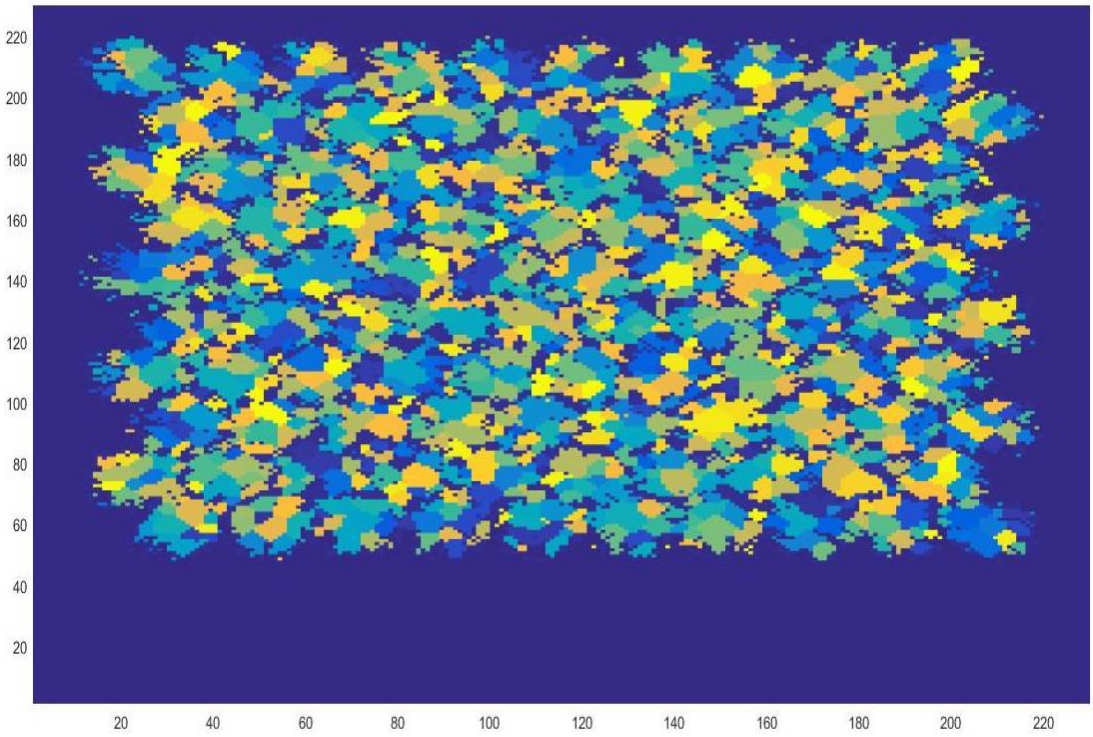

Figure 28: Post Grain Growth

4. Conclusion

The mechanisms of grain growth were first studied along with applicable Monte Carlo Metropolis algorithms using the Q state Potts Model. Algorithms for the simulation of grain growth using a 2D model were then developed and programmed on MATLAB. The program ran successfully with a suitable output that varied with time (Monte Carlo Steps) and the Energy vs Iterations graph was also obtained.

This was then extended to a 3D model and similar outputs were observed.

Verification of the results obtained from Monte Carlo Simulations against the results obtained from the systematic analytical method was done to achieve a high degree of accuracy (more so as the number of iterations and Monte Carlo Steps were increased).

Post analysis and simulation of grain growth in polycrystalline materials, the mechanisms that were at play during liquid phase sintering were studied. Appropriate algorithms were developed to ensure vacancy annihilation and pore stability for a hexagonal close packed system of coated powder particles. The previously mentioned Kawasaki Model was used to simulate Solution-Reprecipitation after this. The concepts used in the simulation of grain growth were then used to achieve a similar level of grain growth in the final system that resulted from vacancy annihilation, pore stability and solution-reprecipitation.

5. Future Work

The exercise above yields a microstructure that has only considered one pass of the laser during SLS. Successive layers would have to be added with the same processes replicated to obtain a realistic model.

This work aims at then studying the resulting microstructure to find the parameters that influence the strength of the 3D printed product. Once the necessary parameters are found out, their relationship with strength would then needed to be studied to accurately obtain the strength of the component, the final goal of this project.

6. References

1. Anderson, M.P., Srolovitz, D.J., Grest, G.S., Sahni, P.S., (1984), "Computer-simulation of grain-growth. 1. Kinetics.", *Acta Metallurgica et Materialia*, pp 783–791
2. Miroslav Morháč and Eva Morháčová (2011), "Monte Carlo Simulations of Grain Growth in Polycrystalline Materials Using Potts Model", *Applications of Monte Carlo Method in Science and Engineering*, pp 563-568
3. Hay, Aaron M. (2011), "Applying massively parallel kinetic Monte Carlo methods to simulate grain growth and sintering in powdered metals", pp 27-36
4. Veena Tikare, Michael Braginsky and Eugene A. Olevsky (2003), "Numerical Simulation of Solid-State Sintering: Sintering of Three Particles", *Acta Metall ,* pp49-51
5. Gregory N. Hassold, I-Wei Chen, and David J. Srolovitz(1990) , "Computer Simulation of Final-Stage Sintering: I, Model, Kinetics, and Microstructure" , *Journal Of the American Ceramic Society - Hassold et al. Vol. 73, No. 10,* pp2857-2861
6. I-Wei Chen, Gregory N. Hassold, and David J. Srolovitz (1990), "Computer Simulation of Final-Stage Sintering: II, Influence of Initial Pore Size" , *Journal Of the American Ceramic Society - Hassold et al. Vol. 73, No. 10,* pp2865-2869
7. Veena Tikare and J. D. Cawley (1998), "Numerical Simulation of Grain Growth in Liquid Phase Sintered Materials: I. Model", *Acta mater. Vol. 46, No. 4*. pp 1333-1342
8. Veena Tikare and J. D. Cawley(1998) ,"Numerical Simulation of Grain Growth in Liquid Phase Sintered Materials: II. Study of Isotropic Grain Growth", *Acta mater. Vol. 46, No. 4,* pp. 1343-1356
9. Nikolay Tolochko, Sregei Mozzharov, Tahar Laoui and Ludo Froyen (2003) , "Selective laser sintering of single- and two-component metal powders", *Rapid Prototyping Journal, Vol. 9 No. 2* ,pp. 68–78
10. R. M. German (1996), "Sintering Theory and Practice", New York: Wiley. pp 420-469.