The Modelling of Damage due to Diffusional Creep in High Chromium Steels

A Dissertation Submitted to the Department of Mechanical Engineering at the University of Cape Town in fulfilment of the Degree of Master of Science in Engineering

By

Royden Weyer

Centre for Materials Engineering

Mechanical Engineering

2016

Supervised by

Prof. Robert D. Knutsen

Prof. Bernhard Sonderegger

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE Reference convention for citation and referencing. Each significant contribution to, and quotation in, this dissertation from the work(s) of other people has been attributed, and has been cited and referenced.
3. This dissertation is my own work.
4. I have not allowed, and will not allow anyone to copy my work with the intention of passing it off as their own work.

Signature: Signed by candidate

Signature Removed

Date: April 2016

Abstract

Understanding the creep deformation of high chromium steels in use in modern power plants has become important in predicting the behaviour and stability of these materials over their operational lifetime. At the deformation rates and conditions recorded in modern power plants, diffusional creep by vacancy migration is seen to be the dominant creep mechanism. However, the understanding of diffusional creep in particle stabilized materials is heavily incomplete. The aim of this project was to model the damage caused by diffusional creep, while considering the microstructure of high chromium steels and the evolution of this microstructure. This problem is addressed by expanding the existing Nabarro-Herring theory on lattice diffusion into a spatially resolved FEM model using MATLAB.

This model focussed on adapting the Nabarro-Herring creep model to handle vacancy concentration changes over time. This allowed the model to produce the primary, secondary and tertiary creep stages present in experimental creep tests. As for microstructure, the focus was on adding precipitates (one of the strongest creep strengthening mechanisms) and voids (the largest cause of material damage). During creep exposure, precipitates were subject to coarsening while voids were subject to growth.

The primary creep stage was formed by the initial rapid flux of vacancies into the body of the grain, due to large chemical potential gradients. A dynamic equilibrium of vacancy concentration would form within the grain, leading to the secondary creep stage. The creep rate produced was similar to that of the existing theory and it was found to decrease with the introduction of precipitates. This was evaluated by analysing the stress gradients caused by hard particles in a softer matrix. These stress fields lowered the stress in the grain boundaries and thus resulted in fewer vacancies being generated. Coarsening led to a reduction in the stress field distribution and thus resulted in creep strength loss in the material.

The inclusion of voids was shown to decrease the initial creep rate, with void growth lessening this effect and leading to the tertiary creep stage. The initial strengthening was due to the void surface replacing the grain boundary as a source of vacancies. As the void surface is a very inefficient source, fewer vacancies were generated, resulting in lower diffusion rates. A slight steady increase in the creep rate over time was shown with the inclusion of void growth. The increase in vacancy generation was caused by the higher stress fields around voids. Initially the stress increase due to a loss in area was accounted for as a stress concentration around the void. Once this void grew too large in relation to the grain size, the stress concentration no longer accounted for all of the stress increase due to load bearing area loss. This resulted in the damage equation coming into play, causing a rapid increase in the stress throughout the grain and leading to the rapid tertiary creep stage.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors, Prof. Robert Knutsen and Prof. Bernhard Sonderegger, for their constant guidance, expertise, and patience throughout this project. I would also like to thank ESKOM, THRIP, CME, and NRF, whose funding allowed me to pursue this project.

Further thanks go to everyone at the Centre for Materials Engineering, whose friendship and assistance made sure there was never a dull moment.

I would like to thank my family, Nolan, Glenne, Lyndall and Andrea, for 25 years of unfailing support and motivation. And lastly to Caitlin, for all of her love, understanding and support, without you I would not have made it this far.

Table of Contents

List of Figures

Nomenclature

a	Cavity radius
A'	Constant for temperature dependent and microstructural parameters
A_C	Coble diffusion constant (150/π)
A_{NH}	Nabarro Herring diffusion constant (range 12-40)
A_v	Factor of value ≈ 3
b	Burger's vector
b_B	Burger's vector of boundary dislocation
c	Concentrations
C_0	Concentration of vacancies at equilibrium conditions
C_e	Concentration of precipitate particles at equilibrium
d	Grain size
D	Coefficient of diffusion
D_{GB}	Grain boundary coefficient of diffusion
D_L	Lattice coefficient of diffusion
D_V	Vacancy coefficient of diffusion
E_1	Young's Modulus of matrix
E_2	Young's Modulus of inclusion
E_V	Formation enthalpy for vacancies
G	Shear Modulus
$G_{v,mol}$	Gibbs energy of the system
J	Diffusional Flux
J_{gb}	Diffusional Flux along grain boundary
k	Boltzmann constant
k_x	Material constant in X direction
k_y	Material Constant in Y direction
K_c	Coarsening constant
L	Linear dimension of grain
M	Taylors Number
n	Stress component
N_{avo}	Avogadro's number
N_v	Mol fraction of vacancies
N_{V0}	Equilibrium concentration of vacancies sin an unstressed state
N_V^+	Mol fraction of vacancies under tension
N_V^-	Mol fraction of vacancies under compression
p	Grain size exponent
Q	Activational energy
r	Radius of precipitate

r_0	Original radius of precipitate
r_{xy}	Radial distance
R	Gas constant
R	Inclusion radius
S	Supersaturation value
t	Time
T	Absolute temperature
v_1	Poisson's ratio of matrix
v_2	Poisson's ratio of inclusion
V_m	Molar Volume of precipitate particles
x	Spatial coordinate
X_M	Concentration of matrix atoms
X_v	Concentration of vacancies
Z_l	Coordination Number
α	Constant of order unity
δ_{GB}	Effective grain boundary width
$\dot{\varepsilon}$	Strain rate
$\dot{\varepsilon}_0$	Material constant characteristic of creep
$\varepsilon_{MM}\ \varepsilon_{vv}\ \varepsilon_{Mv}$	Binding energies
σ	Applied stress
σ_{eff}	Effective stress
σ_{or}	Orowan stress
σ_{th}	Threshold stress
σ_{x}	Stress in X direction
σ_{y}	Stress in Y direction
σ_{xy}	Stress in Shear direction
Ω	Atomic volume
ρ_f	Free dislocation density
λ	Inter particle spacing
λ_s	Cavity separation
λ_{sg}	Width of subgrain
∇f	Gradient of chemical potential
$\mu_{v,mol}$	Chemical potential of vacancies
γ	Particle-matrix surface energy
ω	Damage parameter
ξ	Position of precipitate interface
$\dot{\xi}$	Velocity of precipitate interface

1. Introduction

1.1. Motivation

The long term behaviour of materials under a thermal and mechanical load has become an important topic in predicting the stability and operational lifetime of components. This time dependent deformation of the material, at elevated temperatures and stresses, is defined as creep. Creep is a very complex process and considerable effort has been made to describe and predict this deformation mechanism. Attempts have been made using simple parametric extrapolation or phenomenological approaches, but these have not taken any physical background into account, and therefore any small change in the underlying mechanism leads to very limited predictions. To overcome this limitation, statistical models have been introduced which have gained much popularity in recent times. This approach allows a model to incorporate more complex microstructures and microstructural processes which may affect the creep process over the components lifetime. However, due to the complexity of heterogeneous microstructures and the interactions between microstructural elements, these models can get incredibly complex. The key to better understanding and describing creep is, therefore, with the simple yet accurate consideration of the relationship between the creep process and the microstructure of the material.

When the lifetime of components in a power plant exceeds 25 years [1], the creep rate in these parts must be less than $10^{-11}s^{-1}$ [2] . At these very low rates of deformation, and within the stress and temperature ranges, the transport of matter has to occur by the migration of vacancies, as opposed to the glide of dislocations. This is defined as diffusional creep and has become an important topic in understanding and predicting the long term deformation of power plant steels. The existence and theoretical interpretation of diffusional creep has been well established in pure metals, however, the understanding of diffusional creep in materials which have been hardened by precipitates is incomplete.

This project offers a comprehensive approach to the modelling of diffusional creep so that it can improve upon existing theories. It addresses the existing diffusional creep model, adapting it to allow for the changes undergone by the microstructure throughout the lifetime of a component. It seeks to address the addition of second phase particles by looking at the varying stress fields that would arise in this heterogeneous microstructure. Most importantly, it looks at how these varying stresses would affect the diffusion of vacancies. Another important part of the microstructure included in this research is the presence and growth of voids. Voids are the main constituent of damage in a material which would eventually lead to failure of the material. As a result, the effect of voids in the microstructure is included in the form of their contribution to the damage parameter and how their stress fields would affect the diffusional process.

1.2. Objective

The objective of this study is to set up a spatially resolved model that represents all considered components of a heterogeneous microstructure of high chromium steels, such as X20, P91 and P92. It also includes how this microstructure interacts with the diffusional creep process. This is done in a number of steps, namely:

a) Modelling of the strain rate due to the diffusion of vacancies with locally varying concentrations, which is able to adapt to incremental changes in concentration throughout the components creep life.
b) Investigating the effect on strain rate caused by the stress fields forming around second phase particles.
c) Investigating the effect on strain rate caused by the stress fields forming around voids, including material damage.
d) Investigating the effect on stress fields caused by the microstructural evolution of these particles and voids, and subsequent effect on the strain rate.

1.3. Methodology

A Finite Difference Analysis is used to describe the diffusional creep phenomena using the basis of existing models and improving upon them by including the influence of microstructural processes. Using this method, it is possible to include the microstructural heterogeneities and the changing physical properties of a material when it is subjected to elevated temperatures and stresses. This analysis is done using the software package MATLAB as this is more versatile than any pre-existing finite element software package. This allows for new advancements in the understanding of diffusional creep in precipitate hardened materials.

2. Literature Review

2.1. Creep

2.1.1. Definition

Creep is the slow and irreversible plastic deformation of a material over an extended period of time and under a constant load [3]. Creep takes place at any temperature above absolute zero (0 Kelvin), but only becomes significant at higher temperatures, where there is a loss of material strength. This shows that creep is dependent on the temperature as well as on time and stress. Creep deformation occurs when the load applied to the material is lower than the yield stress, and when subjected to temperatures roughly over 0.4 of the melting temperature of the material.

The continuous flow of material during creep can eventually result in very large plastic deformation and a significant modification to the initial microstructure. This would eventually lead to failure of the material due to creep rupture [4].

2.1.2. Creep Rate Curves

Creep tests are carried out at either constant rate or constant force, the most common being the constant force test. In this test the specimen is kept at a constant temperature while a static load is applied. The resulting strains can then be plotted as creep curves, showing the correlation between the time dependence and the change in specimen gauge length. The theoretical ideal creep curve would be a three stage creep curve as shown in Figure 1 (a).

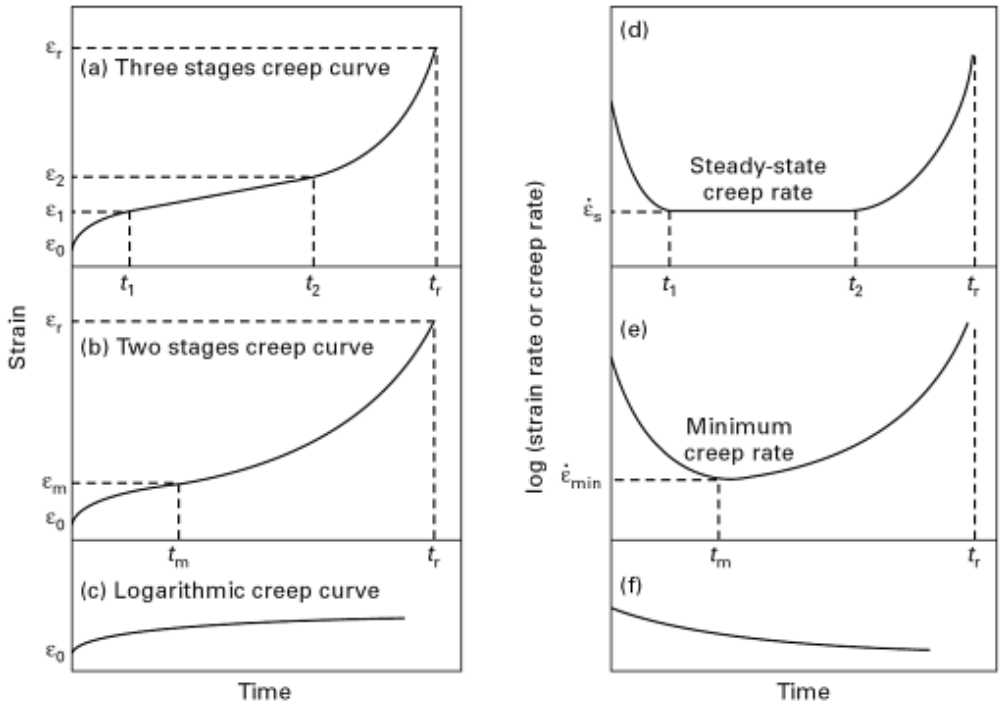


Figure 1 - Schematic of theoretical creep tests showing strain as a function of time (left) and strain rate vs time (right) [3]

The appearance of the creep stages shown in Figure 1(a) (b) and (c) are all due to the deformation of the material under the applied stress and temperature, relative to the materials melting temperature. The ideal creep curve has all three creep stages present; simply known as the primary, secondary and tertiary creep stages. Under different or non-ideal circumstance, fewer creep stages would be observed.

2.1.2.1. Primary Creep Stage

The primary creep stage, also known as the transient creep stage, has a creep rate which decreases with time. This is attributed to strain hardening, a decrease in the free dislocations, or the decreasing flux of vacancies into the body of each grain. A decrease in the number of free dislocations causes strain hardening and thus causes a decrease in the creep rate velocity [3]. Dislocation restructuring starts in this stage and thus microstructural evolution starts to occur. The increase in the subgrain size starts after the initiation of creep, while the coarsening and coalescence of precipitates does not start in the primary stage [5]. In diffusional creep this would be attributed to the initial rapid flow of vacancies into the body of the grain, where the generation of vacancies exceeds the absorption of vacancies.

This primary stage is seen in the ideal curve of Figure 1(a) as ε_0 to ε_1, in which the strain rate decreases over time as shown in Figure 1(d). In a two stage creep curve such as in Figure 1 (b) the primary creep stage is present but immediately progresses to the tertiary creep stage. In Figure 1 (c) only the primary creep stage is present, as is evident from the constant decrease in strain rate shown in 1 (f). This type of creep curve is termed logarithmic creep [3] and only occurs at low homologous temperatures, below roughly 0.3 of the melting temperature.

2.1.2.2. Secondary Creep Stage

In the secondary creep stage, also known as the steady state creep stage, the creep rate remains constant. As creep progresses, the dislocation density increases but these dislocations that are continuously produced are consumed by the formation of subgrains. This causes a balance in generation and annihilation of dislocations. Precipitates start to show a small amount of coarsening and coalescence in this stage. Towards the end of the secondary state, the dislocation density shows an increase on the boundaries while the density of internal dislocations decreases and subgrain size increases slowly [5].

In diffusion creep a steady state gradient for the vacancy concentration would have formed throughout the grain. This represents a balance in the generation and the absorption of the vacancies at the grain boundaries. At high temperatures creep mainly involves diffusion and hence the recovery rate is high enough to balance the work hardening [3].

This stage is only present in the ideal curve of Figure 1(a) as ε_1 to ε_2. It is also shown as the level gradient in 1 (d). In real world applications, a true constant strain rate stage will be absent, such as shown in Figure 1 (b). There is a point where the hardening process is immediately overwhelmed by the softening process. This is shown as the inflection point in Figure 1 (e). There is no true secondary stage in creep resistant alloys and steels and this symbolizes that there is no true dynamic equilibrium in the microstructure. Therefore to find the minimum or steady-state creep caused by an applied stress, one most commonly uses Norton's law [3], which is defined as:

$$\dot{\varepsilon} = A\sigma^n \tag{2.1}$$

$$A = A' \exp(-Q/RT) \quad (2.2)$$

Where n is the stress component.

A' is a constant that includes temperature dependence and microstructure parameters, such as grain size.

Q is the activation energy.

R is the gas constant.

T is absolute temperature.

However this law breaks down at high stresses, when n is very large, and at low stresses, where n is very small [6].

2.1.2.3. Tertiary Creep Stage

The tertiary creep stage, also known as the accelerated creep stage, is where the creep rate increases rapidly until rupture occurs. This increase in strain rate can be attributed to large amounts of microstructural evolution, mainly damage evolution. This damage evolution includes creep void formation and crack development along the grain boundaries [3]. The formation of creep voids also increases the stress in the material and thus creates higher vacancy concentrations, resulting in a higher strain rate. In this stage, the coarsening and coalescence of precipitates advances noticeably, which then causes a decrease in the solid solution and pinning points on dislocation lines. Thus, the distance between pinning points becomes large enough for the movement of dislocations to proceed. This also causes the increase in lath width and the enlargement of subgrain width is accelerated, which further increases the material softening [5].

This stage is shown in Figure 1(a) as ε_2 to ε_f and in Figure 1 (b) as ε_m to ε_f. where the total creep is shown to advance rapidly as well as an exponential increase in the creep rate shown in Figures 1 (d) and 1 (e). In Figure 1 (b) and 1 (e) this sudden transition to the tertiary stage is caused by hardening processes being overwhelmed by the softening processes caused by the microstructural evolution.

2.1.3. Impact of Temperature and Stress on the Creep Curve

Over long periods of time and low stresses, a more complex creep curve would be observed. These creep graphs can vary noticeably with a change in the testing stress and/or temperature. Basically, the total elongation increases but the rupture time decreases with an increase in temperature and the same with an increase in stress. This is illustrated in Figure 2 below.

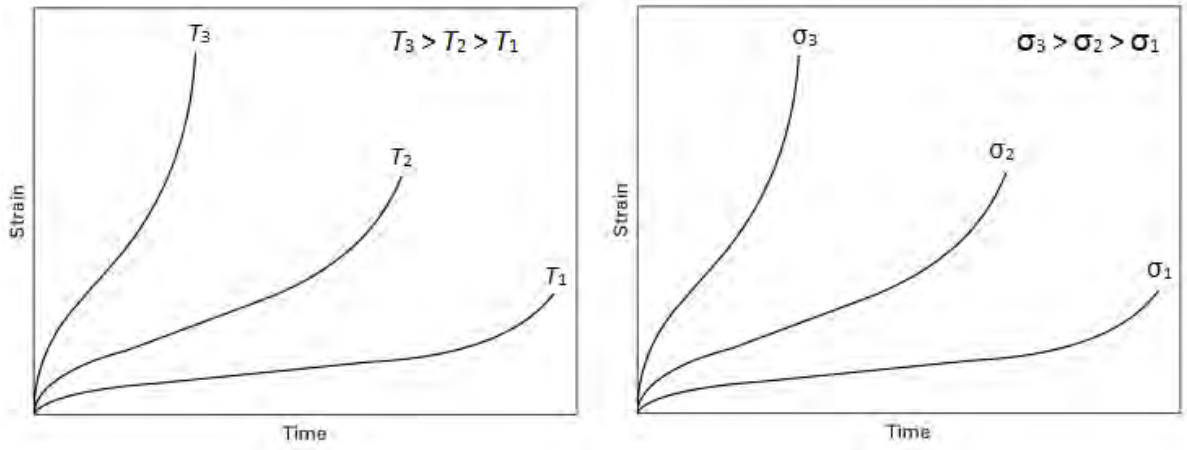


Figure 2- Impact of temperature (left) and stress (right) on the strain rate [Modified from [3]]

Interatomic spacing and shear modulus would change with a change in temperature. The material strength would decrease and the rate of microstructural evolution would increase with an increase in temperature, resulting in quicker creep rates. [4]

An increase in stress results in higher generations of vacancies or dislocations, resulting in the increased rates, however the creep curves can be a little more complex. When stresses are low, systems respond in a mostly linear fashion, such as that for diffusional creep, where the creep rate could be seen as proportional to stress

$$\dot{\varepsilon} \propto \sigma \tag{2.3}$$

At intermediate stresses, which is most often the range of practical interest, the power law creep is observed, which has already been described by Norton's Law in equation (2.1)

$$\dot{\varepsilon} \propto \sigma^n \tag{2.4}$$

At higher stresses the creep rate increases more rapidly than any power of σ and is known as the region of power-law breakdown [4].

2.1.4. Deformation Mechanisms

There are six different deformation mechanisms that are responsible for creep [3]. Each of them is considered to be mutually independent from each other and operate parallel to each other. As each mechanism is independent, each mechanism will contribute additively to the total strain rate. The mechanism that has the highest strain rate will dominate the creep deformation. This mechanism dominance would depend on the temperature and stress ranges. These mechanisms are:

- Defect-free flow
- Glide motion of dislocation
- Dislocation creep
- Volume diffusion flow (Nabarro-Herring creep)
- Grain boundary diffusion flow (Coble creep)
- Twinning

Ashby [7] then proposed the idea of the deformation mechanism map, as shown below in Figure 3. This map is based on the previous assumption that shows where each mechanism dominates at their corresponding stress and temperature. To facilitate the differences between different materials, the stress and temperature are normalised against material properties. On the Y axis the normalized stress is taken as applied stress over the shear modulus of the material. And on the X axis, as the melting temperature has major effects on creep strength of the material [4], the temperature is changed to the operational temperature over the materials melting temperature, which is also known as the homologous temperature.

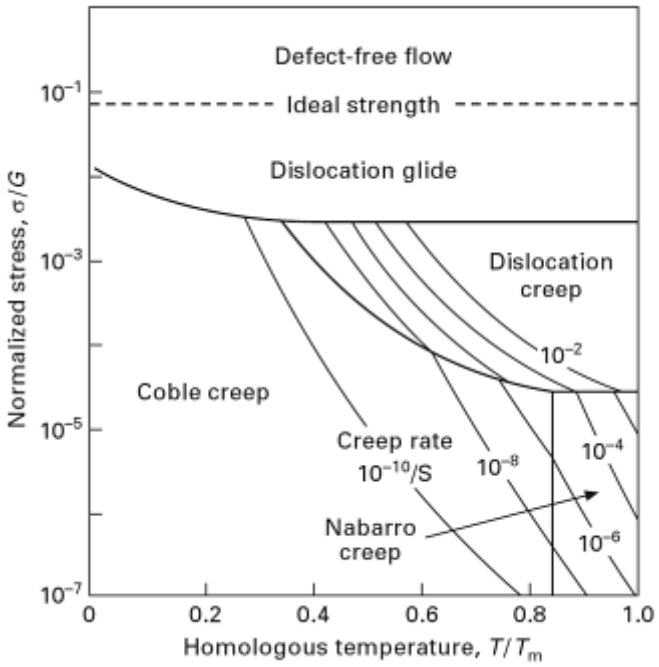


Figure 3 - Deformation mechanism map [3]

At very low stresses the movement of dislocations is very slow and thus can generally be ignored [8], so at these low stresses the deformation is caused by the transport of material through diffusion, as opposed to dislocation motion.

This leads to the study of the volume and grain boundary diffusional creep mechanisms. These mechanisms come into play when the material is subjected to low stresses and elevated temperature, as the mobility of atoms and the equilibrium concentration of vacancies both increase with an increase in temperature [4]. Volume diffusion flow is the flow of vacancies through the lattice of the grain and is defined as Nabarro-Herring creep. Grain boundary flow is the flow of vacancies along the grain boundary and is defined as Coble creep. The creep rates for all fore mentioned creep mechanisms are represented by the general creep equation: [9]

$$\dot{\varepsilon} = \dot{\varepsilon}_0 \left(\frac{\sigma}{G}\right)^n d^p D \tag{2.5}$$

As volume diffusion and grain boundary diffusion are the only diffusional mechanisms considered, the constants n, p, D are all given in a Table 1 [3].

Table 1 : Stress exponent, grain size exponent and diffusion coefficient for equation (2.5)

Deformation mechanism	n	p	D
Diffusion creep			
Coble creep	1	3	D_{GB}
Nabarro-Herring creep	1	2	D_L

Where $\dot{\varepsilon}_0$ is the material constant characteristic of type of creep

D is the grain size

G is the shear modulus

p is the grain size exponent

D_L is the lattice diffusion coefficient

D_{GB} is the grain boundary diffusion coefficient

2.2. Diffusional Creep

Diffusion is defined as the movement of atoms from areas of high chemical potential to areas of low chemical potential, as the diffusing species tries to get a uniform distribution of the atoms. In this study the chemical potential comprises mainly of the concentration of vacancies. At a later stage the influence that stress has on the chemical potential is addressed. This movement of atoms is not due to the action of a direct force but rather due to the random movement, or “walk”, of atoms. In diffusional creep specifically, it is defined as the process where material is transferred from one part of the lattice to another by the flow of vacancies [10].

The temperature dependence, or activation energy, of diffusional creep is the same as that of vacancy diffusion. This means that at temperatures higher than that of half the materials melting temperature, the creep strain rate is similar to that of the diffusion rate. At low stresses and these high temperatures the creep rate is almost completely governed by the diffusion of atoms through the crystal lattice, formally known as Nabarro-Herring creep [11].

2.2.1. Fick's Laws

Diffusion can be described on a macroscopic level (engineering level), mesoscopic level (microstructure level) and microscopic level (atoms and vacancies). At the macroscopic level the theory of diffusion is described by Fick's laws of diffusion and were formulated in 1855 by Adolf Fick [12]. The assumption is that diffusion is equal to a flux of matter proportional to a corresponding concentration gradient.

2.2.1.1. Fick's First Law

Fick's first law is based on the assumption that diffusion is equal to a flux of matter that is proportional to a corresponding concentration gradient. It is describes as: [12]

$$J = -D\frac{\delta c}{\delta x} \tag{2.6}$$

where

J is the diffusion flux,
c is the concentration,
x is the space coordinate measured normal to the section,
D is the diffusion coefficient.

The diffusion coefficient*, D,* is dependent on temperature, pressure, the diffusion mechanism and the structure [13], but can be considered independent from the concentration.

Fick's first law only holds for isotropic materials and it describes the flux of vacancies between two points of concentration which are considered to remain constant. This shows that it has no time dependence as the concentration gradient remains as a steady state and never changes, this type of concentration gradient is shown in Figure 4.

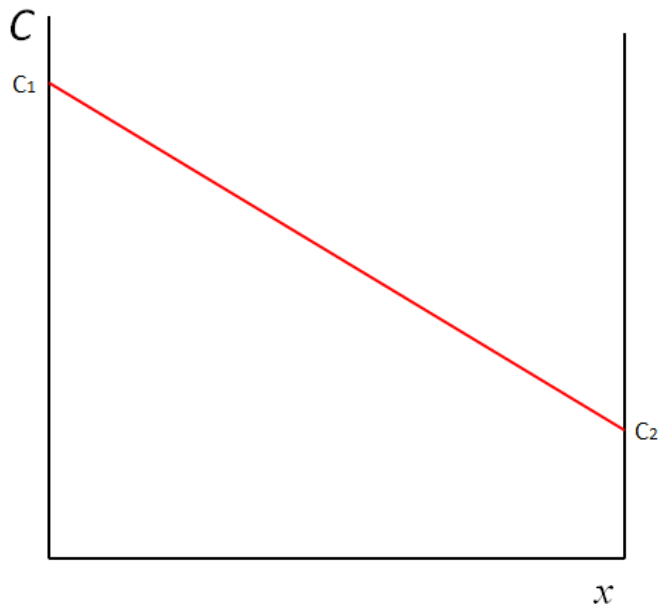


Figure 4 - Steady-state linear concentration gradient

By itself Fick's first law is not particularly useful for the diffusional measurements in solid states. This is because diffusivity rates are fairly small and the attainment of a steady state gradient in a macroscopic specimen can take a very long time. It therefore is advantageous to understand how the concentration gradient changes with time. Thus, to get the time evolution of a concentration gradient, it is usually combined with the equation of continuity: [14]

$$\frac{\delta J}{\delta x} = -\frac{\delta C}{\delta t} \tag{2.7}$$

Where the change in flux, *J,* at a position *x* is dependent on the change of concentration, *C,* over time *t*. This leads to Fick's second law by combining equations (2.6) and (2.7).

2.2.1.2. Fick's Second Law

In practise the concentration of solute atoms at any point in the material could change with time. If a base metal has an atomic concentration *C* of another metal added in solid solution. This secondary metal would diffuse within the base metal to reach a stable concentration. This concentration at time *t* would also have to be dependent on the Cartesian coordinate *x*. This leads to: [15]

$$\frac{\delta c}{\delta t} = \frac{\delta}{\delta x}\left(-D\frac{\delta c}{\delta x}\right) \tag{2.8}$$

Again the diffusion coefficient must be considered independent of concentration for this equation to form Fick's second law:

$$\frac{\delta c}{\delta t} = -D\frac{\delta^2 c}{\delta x^2} \tag{2.9}$$

This second order partial differentiation equation is also referred to as the diffusion equation [14]. This equation describes the transient stage of diffusion from when the higher concentration is first applied at the one face and has to first diffuse across the distance before it finally forms the steady state gradient shown in Figure 4. A random time step in the transient stage is shown in Figure 5, where the concentration steadily rises across the diffusion distance and would eventually converge on the steady state gradient.

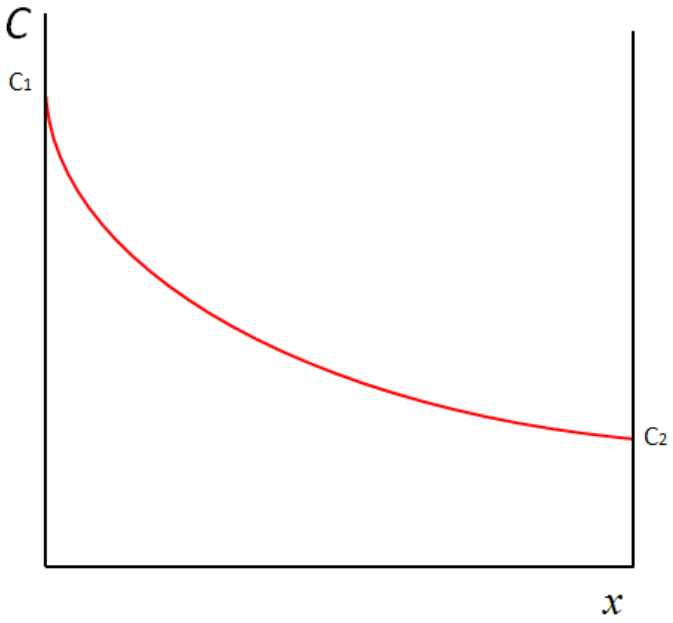


Figure 5 – Transient concentration gradient

Solutions of many types of diffusion scenarios are given by Crank [12]

2.2.2. Diffusional Creep Mechanisms

There are three possible mechanisms of how atoms can diffuse through lattice sites in a crystal such as those shown in Figure 6; the direct exchange mechanism, the ring mechanism and the vacancy mechanism.

The direct exchange method is shown in Figure 6 (a). This works by exchanging the position of two atoms, I and II. For the process to occur, the atom I has to overcome the potential barrier of the periodic lattice and squeeze into an off lattice position, shown as jump 1. It requires a significant amount of activation energy for this process, as shown by the schematic potential diagram. Atom II then moves into the empty space created by the jump of I. Atom I then moves into the original position of atom II. Thus the two atoms have switched place and a single diffusion step has taken place. [10]

Another proposed mechanism is the ring exchange mechanism, shown in Figure 6 (b). Four neighbouring atoms, I, II, III and IV, rotate simultaneously around their central position in a single step. Thus they have exchanged places and a single diffusion step has taken place. This is advantageous over the former mechanism as it requires less activation energy, as shown by the lower peak in the interatomic potential diagram. [10]

The vacancy mechanism, shown in Figure 6 (c), takes advantage of the fact that real crystals are never prefect and that they always contain lattice imperfections. If for instance an atom is located next to a substitutional vacancy, a vacancy in the site of a lattice atom site, it is possible for the atom to fall into this empty position. This mechanism is favourable as the potential barrier for the atom to move into the empty lattice site is much lower than for other mechanisms. Thermal agitation will cause the atom to squeeze past its neighbours and fall into the lattice position of the vacancy [15]. The vacancy is now present in the original lattice position of the moving atom. Effectively the vacancy has moved one lattice position across. This process repeats until the vacancy moves through the entire lattice. The

movement of atoms along the track of a moving vacancy gives rise to diffusional creep. As there is a lower activation energy needed for the vacancy mechanism it makes this the most probable and dominant diffusion mechanism acting in solid state diffusion [10].

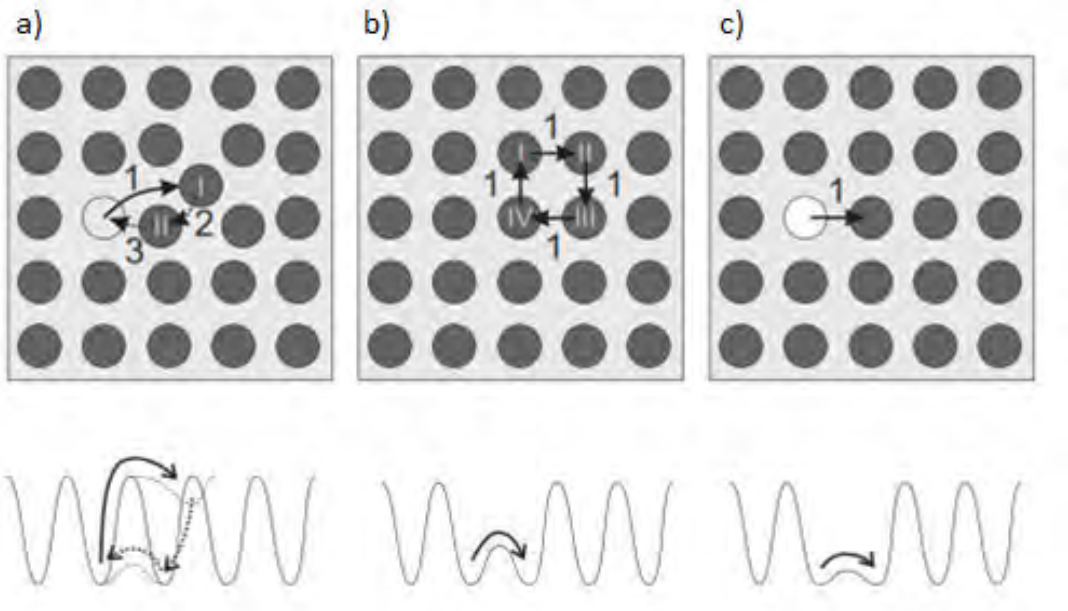


Figure 6 - Mechanisms of solid state diffusion with corresponding interatomic potentials a) Direct exchange mechanism. b) Ring mechanism. c) Vacancy exchange mechanism [10]

2.2.3. Crystal Defects

The physical basis for diffusional creep is established from defects within the materials. The two most common defects are vacancies, shown in Figure 7, and self-interstitials. A vacancy is an atomic lattice position that is not occupied and a self-interstitial is an interstitial defect that consists only of the same atoms already making up the material [16]. Self-interstitials cause a large structural distortion in their surroundings and therefore are much less common [17].

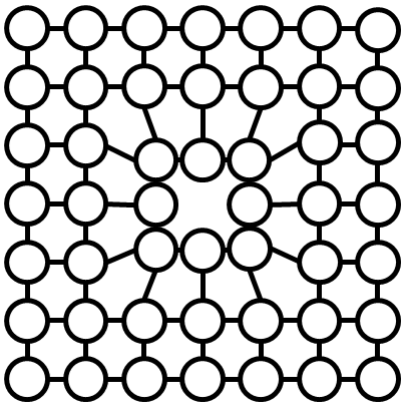


Figure 7 - Vacancy present in a lattice [17]

Vacancies are produced randomly and are necessary due to the second law of thermodynamics. This law states that the entropy of a closed system strives to reach a maximum, and that by adding vacancies to the system, the number of possibilities for atoms to arrange is greatly increased. Therefore it is advantageous to the system to provide the energy for the creation of vacancies [18].

Vacancies need activation energy of about 1 eV to be formed at the grain boundaries. To get the mole fraction of vacancies in equilibrium, one uses [18]:

$$N_v = A_v exp(\frac{-E_v}{kT}) \tag{2.10}$$

Where:

E_V is the formation enthalpy for vacancies
A_V is a factor, which has the value of about 3,
k is Boltzmann's constant

The formation enthalpy, E_V, for vacancy formation is dependent on the material. For iron with a body-centred cubic (bcc) structure, it has a value of about E_V = 154 kJ/mol, whereas for the face-centred cubic (fcc) form a value of about E_V = 135 kJ/mol [19] [10]. The higher activational energy shows that creep will be slower in a close packed lattice structure, such as fcc or hcp structures, as fewer vacancies will be formed, as opposed to a less close packed structure like bcc. [11]

It can be seen from the exponential in equation (2.10) that with decreasing temperature, diffusion becomes rapidly sluggish due to the rapid decrease in the equilibrium vacancy density. It is possible for vacancies which travel with diffusing atoms to be generated or to disappear within the body of the grain when they reach discontinuities in the grain; such as subgrain boundaries, by recrystallization or even due to the formation of precipitates with a lattice orientation different than that of the matrix [15].

2.2.4. Vacancy Diffusion

There are three different mechanisms of vacancy diffusion affecting a material: Firstly the migration of vacancies over the surface of the grain, which has small activation energy value, Q. Secondly the diffusion of vacancies along grain boundaries, this has a relatively medium Q value. And lastly the diffusion of the vacancies through the bulk of the material, which has a much larger Q value than the other mechanisms. This bulk diffusion mechanism has much evidence as being the effective mechanism of creep, especially at high temperatures which compensate for the high Q value [15].

Stress affects the diffusion process when it is applied to a polycrystalline material. When a force is applied, the equilibrium concentration of vacancies will change within the material. The grain boundaries are considered to be the the sources and sinks of vacancies [17]. Vacancies would be generated at the grain boundaries experiencing a tensile stress, or would be annihilated at boundaries that experience a compressive stress. This is because it requires less energy for a vacancy to form at a grain boundary than for it to form on the inside of the body of the grain. Or in the case of compressive stress, it is easier to annihilate a vacancy by squeezing a single boundary atom into a vacant lattice position than to shift the whole row of atoms to annihilate an interior vacancy.

A vacancy is created by pulling an atom out of its lattice position, forming a vacancy-interstitial atom pair. The interstitial atom forms a new layer, effectively moving the grain boundary [4]. To visualize how this is done, consider a square mono crystal under tension. Looking at only the right hand side of the mono crystal, the surface may be thought of as a set of normal forces applied to each individual surface atom, shown in Figure 8 (a). Under this stress and due to thermal fluctuations, some of the atoms of the column can jump to a new equilibrium position, Figure 8 (b). This creates the vacancy just under the surface layer, resulting in an excess concentration of vacancies close to those grain boundaries. This results in the non-homogeneous distribution of vacancies and thus diffusion creep results from the flow of these vacancies in order to restore an equilibrium condition [2]. This

observation was first outlined by Nabarro. Herring later conducted an analysis using chemical potentials and fluxes within the poly crystal [17].

The vacancy is able to flow when an atom jumps into the new vacant position and the vacancy now enters the crystal body, as seen in Figure 8 (c) [20]. This process repeats itself to allow the vacancies to flow from these grain boundaries high in concentration to grain boundaries that have a lower concentration. In this way the vacancy is able to migrate along a random path inside the body of the grain and leave at a free surface not experiencing any stress, as shown in Figure 8 (d). Here the excess vacancies are absorbed by the grain boundary [2]. The motion of a single vacancy through the lattice results in a single movement step for each atom. The overall result is the movement of material as shown in shaded in Figure 8 (e) [20]. The key idea here is that boundaries allow deformation to occur solely by diffusional mass transport. The resulting change in grain dimensions after diffusion of material and vacancy flux is equivalent to the creep strain [8].

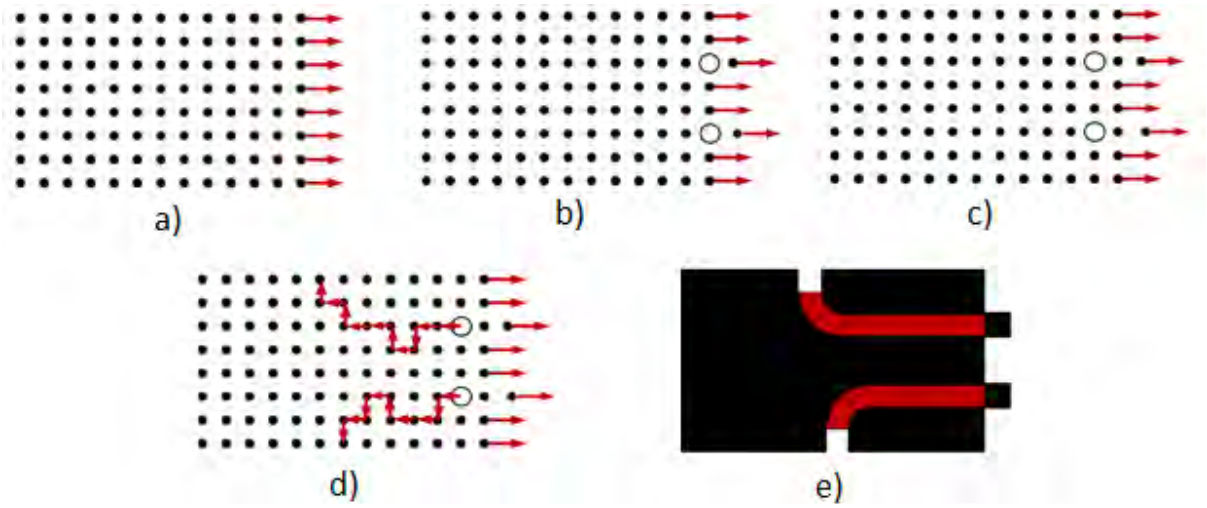


Figure 8 – Mechanism of diffusional creep [Modified from [20]]

In real life situations the grain boundaries have different orientations to the direction of the applied stress; therefore these grain boundaries will experience different normal stresses. The difference in concentration is dependent on the angle between the applied stress and grain boundary. Hence the thermal concentration of vacancies is altered in the direct neighbourhood of the stressed grain boundary. The vacancy concentration in the lattice at a grain boundary is dependent on only the normal stress component of an applied stress [8][21]. New concentrations can be calculated using the equilibrium concentration of vacancies and considering the work expended by the applied stress. This is calculated using equations (2.11) and (2.12).

Work, $\sigma\Omega$, is expended for the movement of the atom, where Ω is the atomic volume. On faces where tensile stresses are acting, the enthalpy for vacancy creation will be lowered by $\sigma\Omega$. Therefore, the equilibrium fraction of vacancy concentration will be changed. A local imbalance in vacancy concentration occurs, where the concentration of vacancies in areas with a tensile stress will increase according to:

$$N_V^+ = N_{V0}\exp(\frac{\sigma\Omega}{kT}) \tag{2.11}$$

And in boundaries subject to compressive stresses it decreases according to:

$$N_V^- = N_{V0}\exp(-\frac{\sigma\Omega}{kT}) \tag{2.12}$$

Where N_{V0} is the equilibrium fraction of vacancies in the unstressed state, given by equation (2.10) [17].

In this example, the difference in vacancy concentration is the driving force for the flux of the vacancies from the areas with high vacancy concentrations to area of low vacancy concentration. This driving force relates to the chemical potential in the case of Coble creep.

Diffusional creep in materials hardened by precipitates or by transformation strains is very incomplete [2].

2.2.5. Diffusion Coefficient

The diffusion coefficient is the proportionality between the concentration gradient and the resulting flux of vacancies [17]. The diffusion coefficient changes for different processes, such as the diffusion of interstitials, self-diffusion and the diffusion of vacancies. The lattice self-diffusion coefficient, D_L, can be calculated using the diffusion coefficient for vacancies, D_V.

$$D_L = D_V N_{V0} = D_V C_0 \Omega \tag{2.13}$$

Where C_0 is the concentration of vacancies in equilibrium conditions, shown as:

$$C_0 = \frac{N_{V0}}{\Omega} \tag{2.14}$$

The diffusional creep rate increases linearly with applied stress [22] and the temperature dependence is determined practically via the coefficient of lattice self-diffusion. This dependence can be described for a thermally activated process whereby the diffusion rate increases with temperature. The change to the self-diffusion coefficient is given as:

$$D_L = D_0 \exp\left(-\frac{Q}{RT}\right) \tag{2.15}$$

Where

D_0 is a pre-exponential factor and Q is the activation energy. Values can be found in [23]and [21]

In the grain boundaries the diffusion coefficient is calculated separately. Here the diffusion can be several orders faster along the grain boundary than through the lattice, thus the resulting diffusion coefficients would be different [17].

With the use of radioactive isotopes it is possible to measure the self-diffusion coefficient, D_L. This coefficient is important because even if the process of diffusion required zero activational energy to occur, the diffusion of the vacancies could not occur faster than the self-diffusion rate of matrix atoms [15].

2.2.6. Grain Boundary Sliding

Diffusional creep and grain boundary sliding are very closely connected. As the grain changes shape due to the flux of vacancies and atoms, the changes would have to be accommodated with a further process of mass transfer at the grain boundaries, such as grain boundary sliding. If this change was not accommodated then the formation of voids and cracks would occur and the coherency of the material would not be maintained. This means that grain boundary sliding is the generally accepted, necessary accommodating process for diffusional creep to maintain structural integrity of the material [24]. The contribution of sliding to the total strain would be of the same order as the contribution of atomic diffusion, meaning both would contribute to the total strain [2].

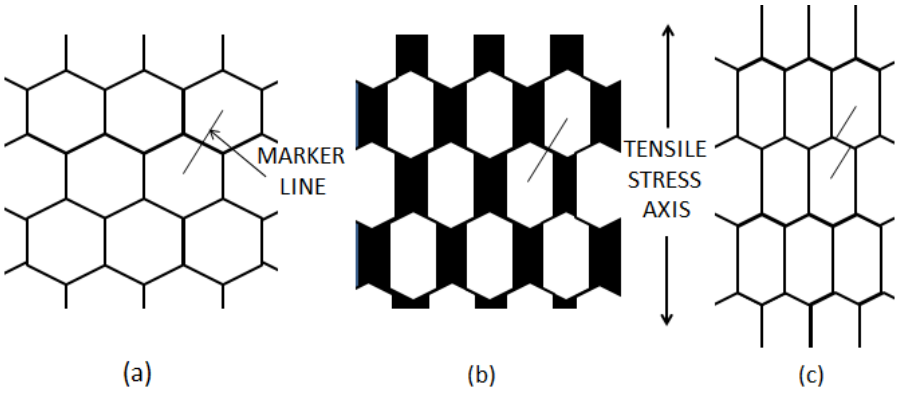


Figure 9 - Illustration of Grain Boundary Sliding where (a) shows an microstructure before creep with a score line across touching grains, (b) shows creep in the absence of grain boundary sliding, causing voids to form, and (c) shows the microstructure after both creep and grain boundary sliding [17]

This grain boundary sliding was first pointed out by Lifshitz as a necessary occurrence in continuum mechanics and is thus termed Lifshitz sliding. It can be illustrated as shown above in Figure 9, where a marker line is scored between two adjacent lines such as in 9 (a). Diffusional Creep then occurs and causes the elongation of the grains, if there was no grain boundary sliding voids would be present such as in 9 (b) and material coherency is lost. Grain boundary sliding then occurs and this can be measured by the displacement of the score lines in 9 (c). As seen in Figure 9, when the grains deform they keep the same neighbouring grains, this is a characteristic of diffusion creep [8]. There is another form of grain boundary sliding not considered in this study as it does not involve diffusion.

The experimental measurements of sliding usually involves examining offsets in maker lines on the specimen surface, or vertical offsets in surface grains, such as shown in Figure 10. These can be used to evaluate the fraction of strain contributed by grain boundary sliding. [25]

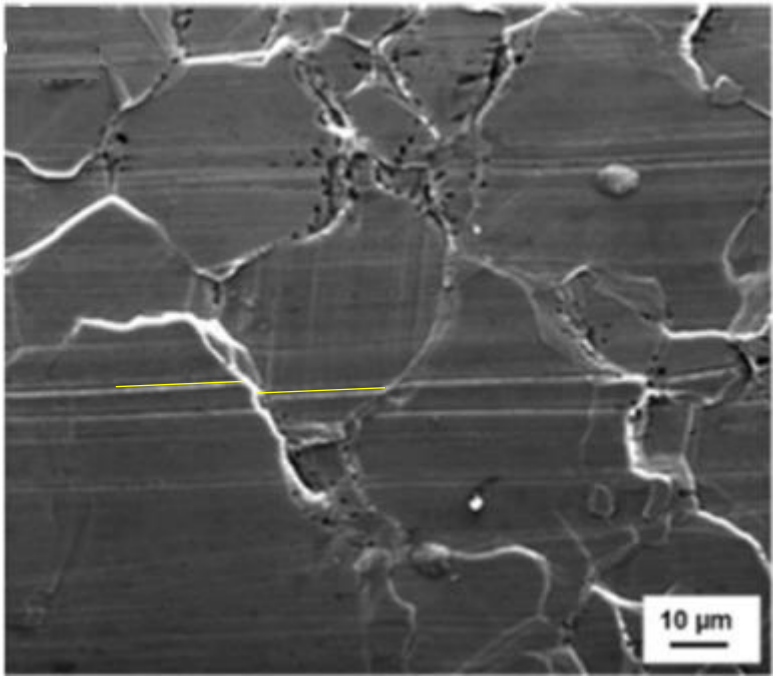


Figure 10 – SEM image of grain boundary offsets in deformed sample [25]

Plastic strain can only take place if both diffusional creep and grain boundary sliding take place simultaneously [17]. When two processes occur simultaneously, one or the other may determine the rate of the combined processes, but the rate determining process may not be the major component of the strain [26]. The contribution of grain boundary sliding can range from >50 % under superplastic flow to a low of <10% under conditions of high stress and coarse grained size [25]. At high temperatures, where grain boundary sliding is an important creep mechanism, samples with a larger grain size usually exhibit a lower creep rate than do the finer grained samples [4] while large contributions of strain were observed in fine grain samples [25]. This shows that grain boundary sliding is heavily grain size dependent. For small grain sizes the grain boundary sliding could represent the main strain mechanism, while in very large grain sized materials grain boundary sliding could be considered negligible.

Figure 11 shows a model done by [27] to investigate the behaviour of polycrystalline aggregates at different grain sizes. For different grain sizes the total grain boundary sliding differs significantly. From the figure it further supports that grain boundary sliding decreases as the grain size increases. When the grain size is larger than 50nm, the grain boundary sliding is less than 0.4% of the grain size and is therefore considered negligible compared to the total plastic deformation of the material. [27]

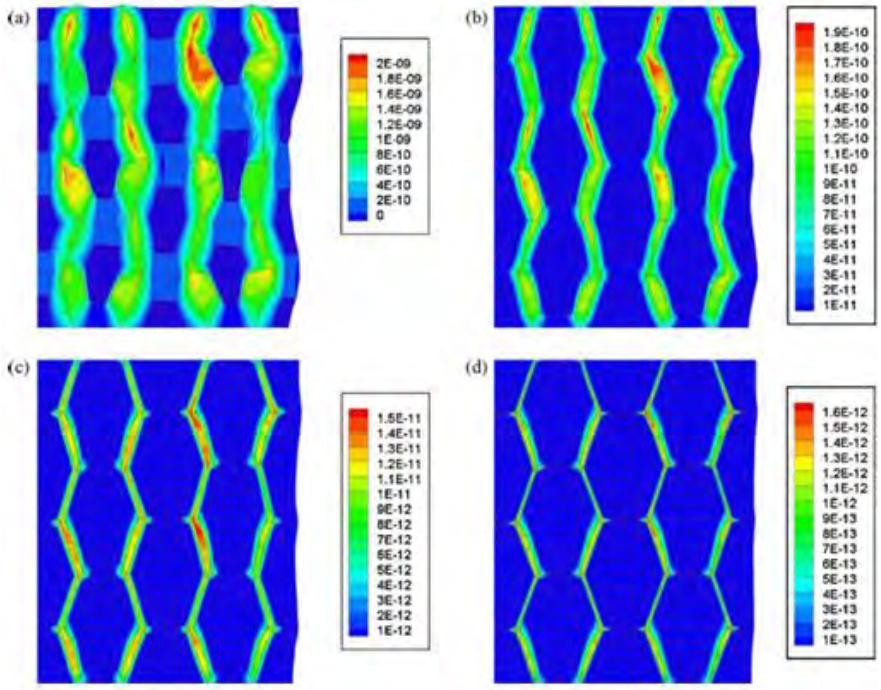


Figure 11 - Grain boundary sliding, depicted in meters, for different grain sizes. (a) 25nm, (b) 50 nm, (c) 100 nm, (d) 200 nm [27]

2.2.7. Denuded Zones

A denuded zone is an area of the grain that is void of any precipitate particles. The formation of these zones has been cited as unique and compelling microstructural evidence for the occurrence of diffusional creep [28]. The reason being that during diffusional creep matrix atoms will flow from grain boundaries parallel to a tensile axis towards the grain boundaries that are perpendicular to the tensile axis, where the new layer of matrix atoms is formed [28]. These denuded zones would represent the amount of matrix material deposited at the grain boundaries by diffusional creep. However, as diffusional creep is usually accompanied by other mechanisms of deformation like grain boundary sliding, denuded zones do not always represent the total strain [2]. It follows that in an alloy that contains a dispersion of second phase particles, the denuded zones would be expected to build up only on the grain boundaries perpendicular to the applied stress, as shown in Figure 12. Additionally precipitates would be expected to accumulate only on the grain boundaries parallel to the tensile axis, as matrix particles are removed from these regions. [28][2]. Denuded zones are not fully accepted as definite evidence for diffusional creep [28].

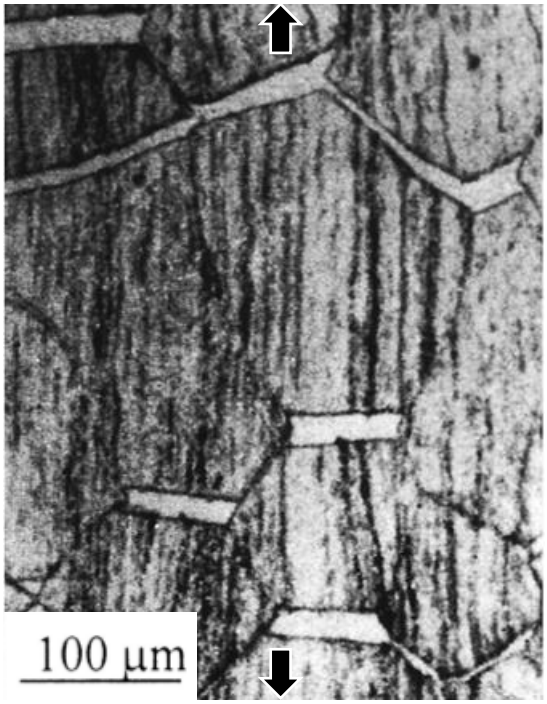


Figure 12 - Denuded zones at grain boundaries experiencing tensile stress [Modified from [28]]

The build-up of precipitates on parallel boundaries gives a scenario of how precipitates could greatly reduce the rate of diffusional creep. Two mechanisms are involved, first; the presence of precipitates on parallel boundaries inhibits grain boundary sliding. Second, the phase boundary, or interface, between a precipitates and matrix is a much less effective source and sink of vacancies than the grain boundary. This is because at a grain boundary the configuration can remain at a low energy state as the two sets of dislocations climb into their own grains. Whereas when the interface is between a matrix and high melting temperature precipitate, which has low vacancy mobility, the climb of dislocations in the matrix alone destroys the low energy structure of the interface [2]. This is possible because the emission and absorption of vacancies in a grain boundary can only occur at grain boundary dislocations. These grain boundary dislocations are confined to grain boundaries and their motion within the boundary can require both glide and climb [29].

2.2.8. Grain Boundaries and Threshold Stress

At very low stresses there seems to be threshold stress behaviour in diffusional creep. In pure metals this value is about 1 MPa but in materials containing precipitate particles on the grain boundaries, this threshold stress can be larger, but not greater than a few MPa [30]. A possible explanation for this is discussed in literature [31][32][33][34][35][36][37] where the grain boundaries act as fast diffusion paths because of their highly disordered atomic structures. The vacancies diffusing through the grain boundaries will partly leak into the surrounding grains by means of volume diffusion.

The generation of vacancies must therefore be fast enough to compensate for the drain of vacancies caused by this leak into the grain. It is assumed that the process is fast and efficient enough that it can easily provide enough vacancies to maintain the equilibrium concentration. This however does not fully explain the threshold stress for diffusion and can possibly be explained as that the absorption of these vacancies is absorbed at discrete points along the grain boundaries and not along the whole grain boundary. Dislocations on the grain boundaries provide an appropriate source and sink for these

vacancies. While emitting vacancies, the dislocations will climb in the boundary plane, thus the diffusion along the grain boundary will be limited by this process. [17] [37]

The applied stress is then replaced by an effective stress.

$$\sigma_{\text{eff}} = \sigma - \sigma_{\text{th}} \tag{2.16}$$

Where σ_{th} is the threshold stress for diffusional creep modelled as: [22]

$$\sigma_{\text{th}} = \frac{1}{2}\frac{Gb_B}{d} \tag{2.17}$$

Where

b_B is the length of the burgers vector of a boundary dislocation.

d the grain size

As the threshold stress is only a few MPa at maximum, under larger stresses this could be neglected.

2.3. Models for Diffusional Creep

There are a number of existing models that describe diffusion creep. Nabarro and Herring independently derived expressions for diffusional creep [22][38][39] and consider diffusion to occur within the grains, and type of creep is referred to as Nabarro-Herring creep. Coble [40] however regarded this diffusional process as taking place along the grain boundaries, and therefore termed Coble creep. These theories of diffusional flux imply that matter is transported directly from transverse to longitudinal grain boundaries with no distortion of the interior of each grain [4]. The main difference between Nabarro-Herring creep and Coble creep arises in grain size dependence [40]. A more recent and not fully verified model is the Harper Dorn Creep model.

2.3.1. Nabarro-Herring Creep Model

The theory of diffusional creep is one of the first creep mechanisms to be analysed theoretically, resulting in one of the few theories which does not make use of arbitrary constants [4]. It is best explained as creep that is controlled by the generation and annihilation of vacancies at low stresses and at high temperatures [41], rather than dislocation creep resulting from formation and annihilation of dislocations. This type of creep occurs from atom movement through the interchange of an atom in a crystal lattice with a vacant site inside the grain. This is more formally referred to as lattice diffusion. The diffusion coefficients relating to Nabarro-Herring creep depend closely on the melting temperature value of the metal. The higher the melting temperature, the lower the diffusion coefficient will be, and vice versa [11].

Nabarro considered the change in concentration of defects near surfaces by acting forces in a cubic crystal [39]. To derive this theory of diffusion he considered a material made up completely of cubical grains and subject to a uniaxial tension σ. At a zero stress state there are an equilibrium number of vacancies already present in the material, which is a function of the temperature. When a uniaxial stress is applied along the cubes axis, vacancies and interstitial atoms are formed in pairs on the grain boundaries normal to this tensile axis. The vacancies diffuse into the bulk of the material, while interstitials form additional layers of the crystal lattice on top of the grain boundaries normal to the tensile axis. The specimen elongates to accommodate these changes and the external forces do work, thus reducing the free energy. These vacancies diffuse to the grain boundaries parallel to the tensile

axis, where they are absorbed. The main driving force for this movement is the concentration gradient of vacancies [4], however it may be linked to a chemical potential in other scenarios.

The situation is modelled as follows: The volume fraction of vacant lattice sites in thermal equilibrium in the unstressed crystal is C_0. When a vacancy is formed at the grain boundary normal to the tensile stress, work σb^3 is done by the external force. According to the Maxwell Boltzmann distribution, the equilibrium local concentration of vacancies increases by a factor of $e^{\left(\frac{\sigma b^3}{kT}\right)}$ and the excess atomic concentration of vacancies is equal to:

$$C_0\left[e^{\left(\frac{\sigma b^3}{kT}\right)} - 1\right] \tag{2.18}$$

Since usually $\sigma b^3 \ll kT$ the expression above can be approximated by:

$$\frac{C_0 \sigma b^3}{kT} \tag{2.19}$$

The excess concentration of vacancies per unit volume is then $\frac{C_0\sigma}{kT}$. If the linear dimension of the grain is *L*, the gradient of the volume concentration of vacancies is then $\frac{C_0\sigma}{kTL}$. The flux of vacancies from transverse to longitudinal is then equal to:

$$\frac{D_v C_0 \sigma}{kTL} \tag{2.20}$$

Where D_v is the diffusion coefficient of vacancies in the material. $D_v C_0$ is equal to D_L, the self-diffusion coefficient for the material by the vacancy migration mechanism. The flux of vacancies is therefore written as:

$$\frac{D_L \sigma}{kTL} \tag{2.21}$$

This corresponds to an equal and opposite flux of atoms. Looking rather at the atom movement, the arrival of an atom adds volume b^3 to unit area, This gives the rate of motion of the boundary wall as equal to $\frac{D_L \sigma b^3}{kTL}$. The velocity of the elongation of a grain of length L is then represented as the strain rate given by

$$\dot{\varepsilon} = \frac{D_L \sigma b^3}{kTL^2} \tag{2.22}$$

This equation gives the secondary creep stage according to Nabarro-Herring. A more precise experimental analysis gives a strain rate larger by a factor of approximately 12 [4] but in some cases has ranged up to 40 [17].

The final Nabarro-Herring creep equation is given below as:

$$\dot{\varepsilon} = A_{NH} \frac{D_L \sigma \Omega}{d^2 kT} \tag{2.23}$$

Where

A_{NH} Constant depending on the geometry of the grain, grain size and load distribution (range 12-40)

D_L is the lattice self-diffusion coefficient

Ω is the atomic volume

This shows that the strain rate is linearly dependent on the applied stress and inversely dependent on the inverse of the grain size squared, d^2.

2.3.2. Coble Creep Model

Coble creep is the movement of vacancies along the grain boundary as opposed to through the grain structure. Coble [40] argued that at lower temperatures the main diffusion paths would be the grain boundaries, thus at these lower temperatures grain boundary diffusion dominates over lattice diffusion [42]. As this occurs at a lower temperature than Nabarro-Herring creep it has a lower activation energy value. The diffusion coefficients for Coble creep are much higher than the diffusion coefficients for Nabarro-Herring creep [11] and the strain rate mainly depends on this grain boundary diffusion coefficient. Herring also used a gradient of chemical potential as opposed to just concentration gradient to express the diffusional flux of atoms and vacancies within a poly crystal [38].

Creep by lattice diffusion is the predominant mechanism of creep only when stress is low and the temperature is very close to the melting point. Coble pointed out that the activation energy for self-diffusion along grain boundaries was considerably less than for self-diffusion through the grain. At temperatures over 100 °C below the melting point or more, the transport of matter along the grain boundaries would start to occur more rapidly than through the bulk of the grain, even though the area available for diffusion is much smaller [4].

The lattice diffusion coefficient, *D*, is replaced by a grain boundary diffusion coefficient D_{GB}, which has roughly half the activation energy of *D*. In addition the area of the channel through which diffusion can occur is reduced from approximately L^2 to *αbL*, where *α* is a constant of order unity. The rate of diffusional creep by the Coble mechanism is obtained by multiplying the Nabarro Herring model by the ratio of the diffusional areas, $\frac{\alpha bL}{L^2}$, and then replacing *D*, therefore the model becomes [4]:

$$\dot{\varepsilon} = \frac{\alpha D_{GB} \sigma b^4}{kTL^3} \tag{2.24}$$

The final Coble model given in [42] is then given as:

$$\dot{\varepsilon} = A_C \frac{D_{GB} \delta_{GB} \sigma \Omega}{d^3 kT} \tag{2.25}$$

Where

D_{GB} is the grain boundary diffusion coefficient

δ_{GB} is the effective grain boundary width

A_C is a constant with the value of approx. value 150/π [42][43]

This shows that the strain rate is also linearly dependent on the applied stress and inversely dependent on the grain size cubed, d^3. This shows that Coble creep has much higher grain size dependence than Nabarro-Herring creep.

2.3.3. Harper-Dorn Model

The basic mechanism of Harper-Dorn creep is the migration of vacancies from the edges of dislocation with Burgers vector parallel to the tensile axis to those with Burgers vectors perpendicular to this axis [2].

This gives a strain rate that is proportional to the stress, independent of grain size and proportional to the dislocation density. The whole process is described to occur within the grain, and has no accumulation or depletion of the material at grain boundaries. It is characterized by the initial dislocation density that is independent on stress. It is only observed in samples with low initial dislocation density and not in those of high dislocation density [44].

Harper-Dorn seems to be exhibited at temperatures very close to melting temperature of the material, but where the strain rates are significantly faster than those predicted by the theory of Nabarro-Herring creep [6]. Here the creep is a dislocation mechanism occurring in the grain interior where grains are elongated without the need for any accommodating sliding. It cannot be identified from measurements of the activation energy for creep, Q, or from the value of the inverse grain size exponent, as these match dislocation creep, it can only be identified by the stress exponent n =1 [6]. The understanding of Harper-Dorn is much less satisfactory than the previous models and will be avoided in this study.

2.4. Stress Fields Around Hard or Soft Particles

Stress fields are formed as stress concentrations around hard precipitates and voids. These stress fields are caused by the difference in Young's modulus between the matrix and inclusion when stress is applied to the material.

The analytical representation of these stress fields in the X, Y and shear directions was addressed by Yevgeny [45]. The method used to derive these stress fields was the super positional method, which gives an approximation of the linear theory of elasticity. Using this principle, the total solution of the boundary problem can be represented in the form of a superposition of more simple solutions, under the condition where the boundary conditions remain the same.

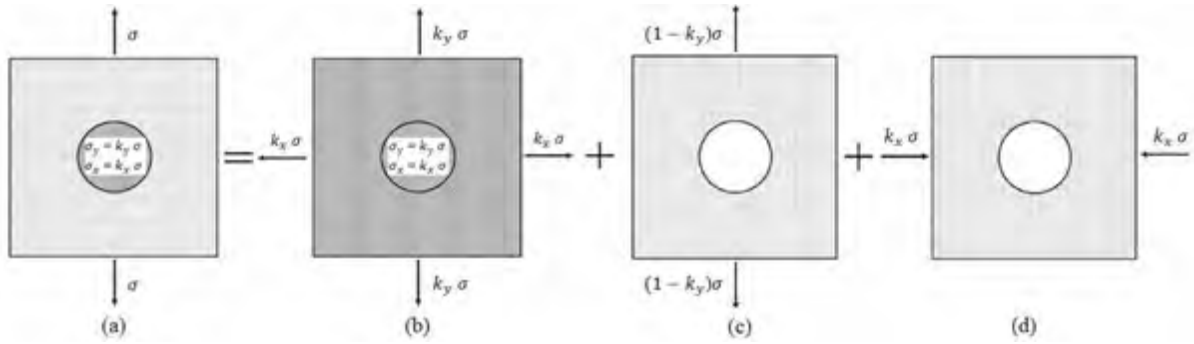


Figure 13 – Schematic representation of the stress fields around an inclusion split into three definable scenarios [45]

Yevgeny [45] derives these equations by taking the situation where a hard particle rests in a softer matrix with a tensile stress directed along the Y axis, as illustrated in Figure 13 (a) above. Within the bounds of the precipitate radius, Eshelby [46] showed that the X, Y and shear stress fields inside the inclusion will be homogeneous, defined as $\sigma_y = k_y\sigma$, $\sigma_x = k_x\sigma$, $\sigma_{xy} = 0$ respectively. k_y and k_x are

constants that have been introduced to account for differences in material properties between the matrix and precipitate.

This scenario is then split into three simpler scenarios that are easily definable. Figure 13 (b) illustrates a homologous field under biaxial stresses, where the whole field has the properties of the precipitate. Figure 13 (c) and (d) illustrate a field with a hole located at the position the precipitate would have occupied, (c) is under tension along the Y axis and (d) is under compression along the X axis. The stress experienced in each scenario is also a function of the constants k_y and k_x.

The resulting stress fields in the Y, X and shear stresses are summed up from the three different scenarios:

$$\sigma_y = \sigma_y^b + \sigma_y^c + \sigma_y^d \tag{2.26}$$

$$\sigma_x = \sigma_x^b + \sigma_x^c + \sigma_x^d \tag{2.27}$$

$$\sigma_{xy} = \sigma_{xy}^c + \sigma_{xy}^d \tag{2.28}$$

These resulting equations for the stress fields outside of the precipitate are given below. Equation 2.29 gives the stress in the Y direction, Equation 2.30 gives the stresses in the X direction and Equation 2.31 gives the shear stresses. Note that all stresses have been divided by the applied stress to give a normalized stress.

$$\frac{\sigma_y}{\sigma} = 1 + \frac{(1 - k_y + k_x)R^2}{2{r_{xy}}^2}\left[1 + \frac{3R^2 + 10y^2}{{r_{xy}}^2} - \frac{8y^2(3a^2 + 2y^2)}{{r_{xy}}^4} + \frac{24R^2y^4}{{r_{xy}}^6}\right] - \frac{k_xR^2}{{r_{xy}}^2}\left(1 - \frac{2y^2}{{r_{xy}}^2}\right) \tag{2.29}$$

$$\frac{\sigma_x}{\sigma} = \frac{(1 - k_y + k_x)R^2}{2{r_{xy}}^2}\left[3 - \frac{3R^2 + 18y^2}{{r_{xy}}^2} - \frac{8y^2(3a^2 + 2y^2)}{{r_{xy}}^4} + \frac{24R^2y^4}{{r_{xy}}^6}\right] + \frac{k_xR^2}{{r_{xy}}^2}\left(1 - \frac{2y^2}{{r_{xy}}^2}\right) \tag{2.30}$$

$$\frac{\sigma_{xy}}{\sigma} = \frac{(1 - k_y - k_x)R^2xy}{{r_{xy}}^4}\left[3 - \frac{2(3R^2 + 4y^2)}{{r_{xy}}^2} + \frac{12R^2y^2}{{r_{xy}}^4}\right] - \frac{2k_xR^2xy}{{r_{xy}}^4} \tag{2.31}$$

As stated before, the normalized stresses inside the inclusion are constant and given as:

$$\frac{\sigma_y}{\sigma} = k_y \tag{2.32}$$

$$\frac{\sigma_x}{\sigma} = k_x \tag{2.33}$$

$$\frac{\sigma_{xy}}{\sigma} = 0 \tag{2.34}$$

The constants k_y and k_x are then given as equation 2.35 and 2.36 respectively, which accounts for the differences between the Young's modulus and Poisson's ratio of the matrix and the precipitates.

$$k_y = \frac{E_2[(3 - v_2)E_1 + (5 - v_1)E_2]}{(E_1 + 2E_2)^2 - [v_2E_1 + (1 - v_1)E_2]^2} \tag{2.35}$$

$$k_x = \frac{E_2[(3v_2 - 1)E_1 + (1 - 3v_1)E_2]}{(E_1 + 2E_2)^2 - [v_2E_1 + (1 - v_1)E_2]^2} \tag{2.36}$$

The resulting stress fields from these given equations are shown below. Figure 14 (a) represents the stresses in the Y direction, where there are high tensile stresses present in the precipitate and along the tensile axis, with lower tensile stresses occurring at the sides of the precipitate. Figure 14 (c) shows the stresses in the X direction, with low stresses inside the precipitate, and higher tensile stresses along the tensile axis. It also shows that there are now compressive stresses at the sides of the precipitate. Figure 14 (e) then shows the shear stresses in the system. There are no shear stresses inside the precipitate and low tensile and compressive shear stresses are offset from the tensile axis.

These equations are also able to represent the stresses around a circular void. By taking the Young's modulus and Poisson's ratio as zero, $E_2 \to 0$ and $v_2 \to 0$, it is equivalent of only considering the stresses in scenario (c) and (d) from Figure 13. The resulting X, Y and shear stress fields in the matrix around a void are shown in Figure 14 (b) (d) and (f) respectively. Immediately obvious is that the stresses are an inverse of stresses around a hard precipitate, which is expected. The stress concentrations around a void are shown to be much higher than that around a hard precipitate. The resulting stress at the transverse void surface is a factor of three times larger than the applied stress, which fits in with stress levels reported in [47].

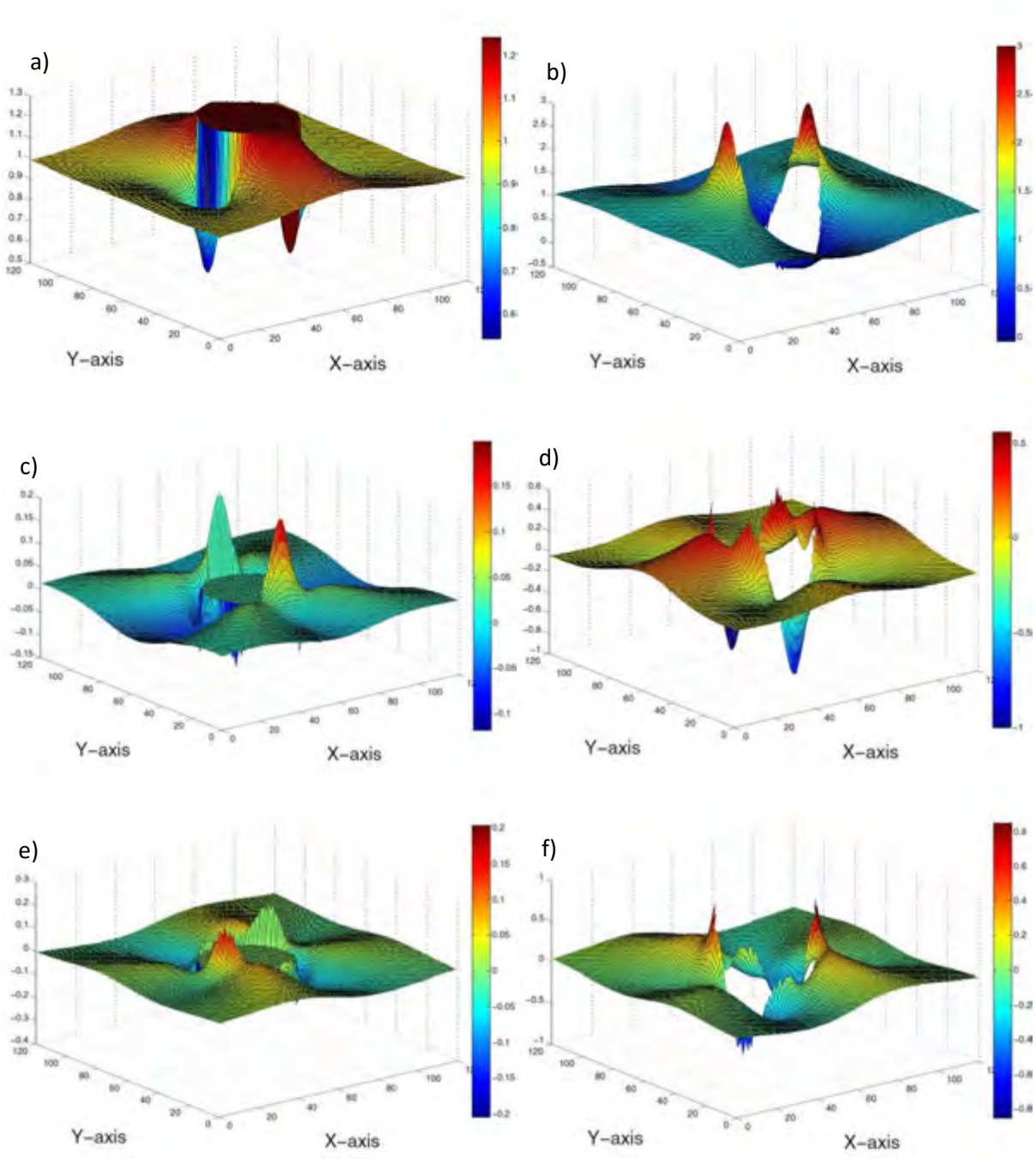


Figure 14– Stress fields around a hard precipitate given as normalized stresses: (a) y axis stresses, (b) x axis stresses, (c) shear stresses. Stress fields around a void: (d) y axis stresses, (e) x axis stresses, (f) shear stresses [modified from [45]]

2.5. Creep Resistant Steels

2.5.1. Material Selection for Creep Resistant Steels

Under the high temperature and elevated pressure environment in use at a power plant the materials have a historical problem of thermal fatigue. Under these circumstances a material with a low coefficient of thermal expansion and low elastic coefficient are favourable. It is for these reasons that ferritic and martensitic steels, such as P91 and P92, are chosen over austenitic steels in a power plant. Their higher thermal conductivity provides better structural stability by providing quicker transfer rates to temperature changes, causing lower thermal stresses to be experienced [2]. The chemical composition of P91 and P92 are given in Table 2. The composition of these materials, high temperature resistance, oxidation resistance, good thermal conductivity and low thermal coefficient of thermal expansion make these materials highly creep resistant, as well as the strengthening mechanism which form in the heat treatment of these steels [48]. The most important strengthening mechanisms in steels operating during high temperature creep exposure are precipitation strengthening and solid solution strengthening [49]

Table 2 - Limits of chemical composition of P91 and P92 creep resistant steels (wt %) [50]

Element	C	Mn	Si	S	P	Cr	Ni	Mo	W	V	Nb	N	B	Al
P91 Minimum	0.08	0.30	0.20	-	-	8.00	-	0.85	-	0.18	0.06	0.03	-	-
P91 Maximum	0.12	0.60	0.50	0.01	0.02	9.50	0.40	1.05	-	0.25	0.10	0.07	-	0.04
P92 Minimum	0.07	0.30	-	-	-	8.50	-	0.30	1.5	0.15	0.04	0.03	10 ppm	-
P92 Maximum	0.13	0.60	0.50	0.01	0.02	9.50	0.40	0.60	2.0	0.25	0.09	0.07	60 ppm	0.04

2.5.2. Microstructure

Creep is closely related to the microstructure of the material. It describes the totality of all thermodynamic non-equilibrium lattice defects and phases [17]. The microstructure will influence the creep behaviour and the creep process will change this microstructure, resulting in a complex relationship. The microstructure of 9-12% Cr ferrite or martensitic steels typically contain laths with a high dislocation density and precipitates distributed along the grain and lath boundaries as shown in Figure 15.

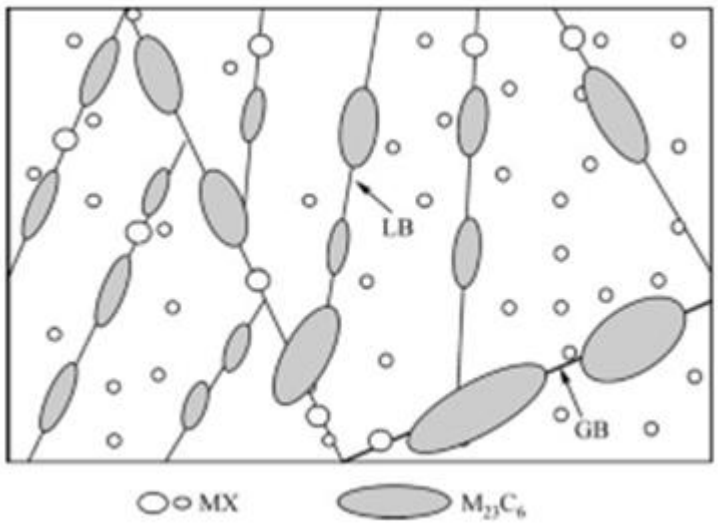


Figure 15 - Typical microstructure of tempered high Cr steel [51]

2.5.2.1. Grain Boundaries

The interface between two adjacent grains is termed as the grain boundary, and they play a very large role in diffusional creep. They are considered to be between one atom and 10 nm thick and are defined by taking the relative misorientation between the two grains. This angle is used to separate grain boundaries into high angle boundaries and low angle boundaries [52].

High angle grain boundaries are defined as interfaces with large differences in orientation between adjacent grains. Many boundary models are based on the idea that a boundary consists of an array of dislocations. These grain boundary dislocations cannot be assumed to have the same structure and Burgers vector as lattice dislocations [53]. If these dislocations are uniformly spaced, a low energy boundary is assumed to form and the boundary is assumed to be in equilibrium. This only occurs if the dislocation spacing matches the number of lattice planes terminating at an interface. This is shown in Figure 16. When a grain boundary does not have an equilibrium structure, such as during deformation, the structure differs and defects such as vacancies can be included.

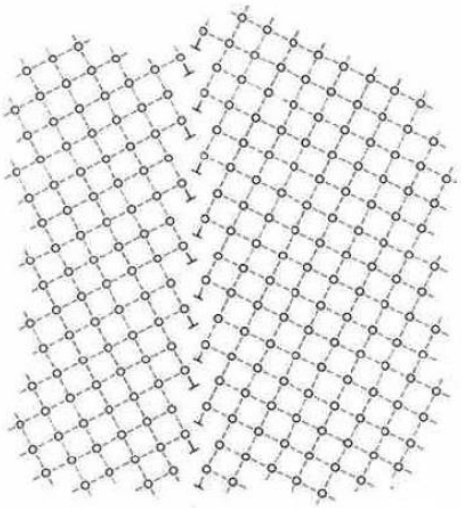


Figure 16 - High angle brain boundary [53]

Small angle grain boundaries are defined as interfaces between lattices that have only a slightly different misorientation. This type of boundary can be view as periodic networks of dislocations and can be quite complex, this is shown in Figure 17 [17]. Their formation is usually due to dislocations forming modulated structures within a grain. These low angle boundaries form subgrain boundaries where their formation and size is dependent on the dislocation density [22].

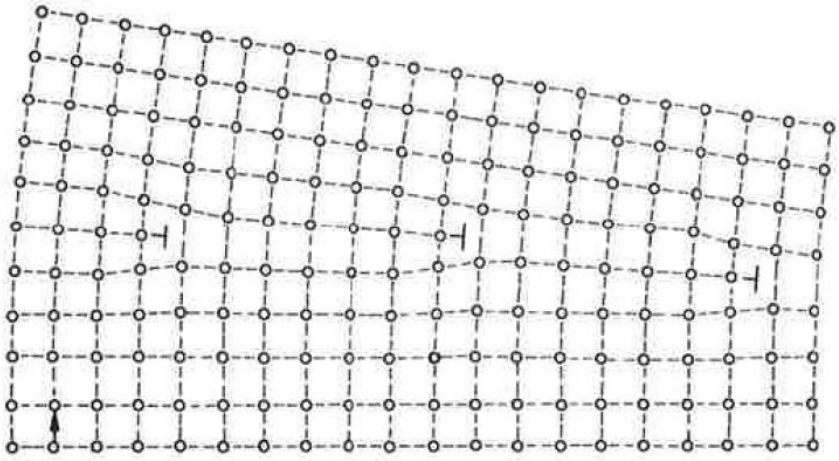


Figure 17 - Small angle grain boundary [53]

2.5.2.2. Precipitates

Precipitates have a different chemical composition and different physical properties than that of the surrounding matrix and can affect the strength of the material in two ways. They can have a larger or smaller bulk modulus, different lattice spacing or have different crystallographic orientations. They are mostly considered as obstacles for the movement of dislocations or preventing the movement of subgrain boundaries. [17]

Precipitates form when a group of substitutional atoms concentrate on the lattice sites within a small spherical region, creating a small coherent precipitate. The first case of increasing material strength is in the early stages of precipitate formation. Considering the case that the substitutional atoms are larger than the matrix atoms, the material inside the sphere will then be under pressure. To relieve this pressure the precipitate expands until the elastic energy is minimized. In this state, the precipitate is still compressed, just less than previously and the matrix now suffers an elastic strain. These strains interact with and impede the motion of dislocations. The resulting hardening would be large, but the stresses in the matrix would anneal out rather readily meaning this is not effective at suppressing creep for long periods at high temperatures [4].

During the process of annealing, lattice vacancies move from the matrix into the interface between the matrix and the precipitate particle [17]. Each arriving vacancy destroys one lattice site, expanding the space available to the precipitate until the precipitate and matrix are free of elastic strains. While this could be seen as a sink of vacancies, possibly effecting diffusional creep, it is done in the tempering stage so does not affect the creep resistance of the material. Due to the arrival of vacancies the number of lattice sites in the precipitate is now less than the number required to fill void in the matrix,

and the precipitate is now incoherent. The orientations could even be completely different to the matrix altogether. This forms an incoherent precipitate that no longer produces long range stresses in the matrix. However these precipitates still impede the motion of dislocations because the dislocations cannot penetrate into the harder, stronger bonded material that forms the precipitate. [4]

In the book the Physic of Creep by Nabarro [4], he states that there is no reason to expect precipitates to prevent diffusional creep, whether it occurs by Nabarro-Herring or Coble creep mechanisms. The only reason given is that they may cause the grain boundaries to become inefficient sources or sinks of vacancies. This study hopes to give the stress concentrations formed around precipitates when under stress as a reason for preventing diffusional creep.

The most common precipitates affecting creep strength are the dispersions of carbide, $M_{23}C_6$, and carbonitrides, MX, as well as Laves and Z phase which precipitate during creep [54][55]. Other, less common precipitates present in high chromium steels which will not be covered are AlN and MnS. Aluminium is detrimental as it precipitates in the form of coarse AlN particles, thus reducing nitrogen, and as such, the availability of N for fine VN particles [56]. MnS is however good as it removes sulphur, which causes embrittlement of the steel.

$M_{23}C_6$

$M_{23}C_6$ is a more general term for $Cr_{23}C_6$ or the $Cr_{21}Mo_2C_6$ that forms in steels containing significant amounts of Mo. $M_{23}C_6$ carbides are the most abundant precipitate in creep resistant steels and therefore the strength and creep properties of the steel is highly dependent on this precipitate [49]. They are large precipitates of 50-300 nm and have an fcc structure. These precipitates are found at the early stages of tempering because they nucleate easily on internal interfaces. They coarsen during creep exposure, weakening their pinning effect. It has been shown that they coarsen much faster at temperatures of 600°C and higher [48].

MX

MX forms when strong carbide/nitride metals (Ti, Nb, V, Zr or Ta) are combined with C or N. They are a smaller sized particle of around 10-15 nm and also have an fcc structure like that of $M_{23}C_6$. They usually form during the tempering process on dislocations within the matrix, on stacking faults and on grain boundaries, where they increase the creep strength by the pinning of dislocations [48]. MX precipitates are found to be stable and do not coarsen heavily during creep exposure; however it is found that they dissolve and form complex Z-phase particles under long term creep exposure [49]. Some precipitation of fine MX particles is still observed during long term creep which would further reduce the microstructural instability with the obstruction the dislocation mobility [57].

Laves Phase

Laves phase are large intermetallic precipitates of composition $(Fe,Si)_2(Mo,W)$ that nucleate and coarsen during creep exposure on the grain boundaries close to $M_{23}C_6$. It forms due to the Mo and W that is added for solid solution strengthening. The precipitation of this phase then leads to the depletion of these solid solutions, resulting in a loss of this hardening mechanism, this lead to the belief that it had a detrimental effect on the creep strength. Recently however there has been ambiguity on the effect of laves phase on creep strength. It has been said that the precipitation of

laves phase can increase the creep strength by precipitation hardening, before it coarsens obviously [1]. Therefore it is believed it only has a negative effect once it coarsens to one of the largest precipitates in the alloy, roughly >200 nm [49]. This phase only precipitates after around 10000h of creep exposure. Si and Cu have been shown to hasten the formation of laves phase [48].

Z Phase

Z phase precipitates form very large particles at the expense of depleting the small and stable MX precipitates, specifically VN, but they have a very slow coarsening rate. This can have a crucial influence on the breakdown in creep strength of high Cr ferritic steels. The formation of Z phase is much faster in steels with >10% Cr content. Thus the Cr content can be kept low to retard formation considerably [58]. Very little is known about Z phase as it precipitates only after long term creep exposure. It has been recognised to decrease long term creep strength by consuming the fine MX particles. They precipitate as large particles, which do not contribute to precipitate strengthening, thus the creep strength is considerably lowered by their formation [49].

2.5.3. Strengthening Mechanisms

Creep resistant steels need to be reliable over long periods of time and at elevated temperatures; therefore they need to be strengthened against long term creep. The basic methods for hardening against creep are solid solution hardening, precipitation, dislocation hardening and boundary or sub-boundary hardening. It is possible to combine these mechanisms but difficult to quantify each contribution to the overall hardening [58].

2.5.3.1. Solid solution Strengthening

This type of strengthening mechanism uses atoms with different atomic sizes than that of the solvent material, this creates lattice distortions that would impede dislocation movement and increase the yield strength of the material. These atoms can either be substitutional or interstitial. Substitutional atoms would impart a compressive or tensile stress, depending on the solute size, which would then interfere with the stress field around a dislocation, forming a barrier against their movement. Substitutional atoms such as Molybdenum (Mo) and Tungsten (W), are used effectively in high chromium steels due to their larger size than the iron matrix [58]. Interstitial atoms will typically have a stronger effect than substitution atoms. When the atom occupies more space than the space between two atoms it can produce a much higher distortion in the lattice and a subsequent higher stress field that can interact with dislocations.

This type of strengthening is superimposed over other mechanisms such as precipitation hardening. The addition of solid solution hardening usually leads to formation and fine distributions of precipitates during elevated temperature exposure. It also may have the effect of inhibiting grain growth and suppressing grain boundary sliding [4].

2.5.3.2. Precipitate Hardening

This is one of the most important strengthening mechanisms in creep resistant steels. Creep resistant steels usually contain several kinds of precipitate particles in the matrix, namely; MX carbonitrides [(V, Nb) (C, N)], $M_{23}C_6$ Carbides [(Cr, Fe, Mo, W)(C,B)], Laves phase [$(Fe, Si)_2$(Mo, W)], Z-phase [(Cr)(V,

Nb)(N)], Aluminium nitride [(AlN)] and Manganese sulphide [(MnS)]. A dispersion of fine precipitates is highly effective in stabilizing dislocation motion in the crystal matrix [59].

Large precipitates will normally have a different crystal structure from the matrix. This prevents the dislocations from the matrix passing directly into the precipitate [58]. This can however be overcome when a higher stress is applied to the steel. The threshold stress needed for a dislocation to pass through a precipitate is represented by the Orowan stress: [59]

$$\sigma_{or} = \frac{0.8MGb}{\lambda} \tag{2.37}$$

Where *M* is the taylor factor

b is the magnitude of Burgers vector

λ is the inter particle spacing .

As precipitates tend to coarsen there is an increase in λ which leads to a decrease in Orowan stress overtime. The coarsening and dissolution of fine precipitates usually occurs at grain boundaries, which promotes the formation of localized weak zones and promotes localized creep deformation at these grain boundaries [59] [60].

2.5.3.3. Dislocation Hardening

Dislocation hardening is an important means of strengthening steels at an ambient temperature. Tempered martensitic steels usually contain a high density of dislocations even after tempering. This density can be controlled by the temperature of the tempering process. A high number of dislocations mean they interact with one another, with their stress fields having a repulsive force against one another [59].

This dislocation hardening, given as $\sigma_{dislocation}$ is :

$$\sigma_{dislocation} = 0.5MGb\sqrt{\rho_f} \tag{2.38}$$

Where ρ_f is the free dislocation density in the steel after tempering [58]

At elevated temperatures however excess dislocations accelerate the recovery and recrystallization during creep, with the aid of applied stress. If tempered at a higher temperature the number of dislocations is too low to promote recovery and recrystallization during creep. This suppresses the rapid decrease in creep rupture strength. Thus dislocation strengthening is useful for creep strengthening for only a short time and has no impact on long term creep at high temperatures [59].

2.5.3.4. Sub-Boundary Hardening

Tempered martensitic steels with high Cr content usually have lath martensitic microstructure consisting of lath and block sub-structures. These boundaries provide hardening with respect to the width of these subgrains, again by impeding the movement of dislocations. This generally gives a larger stress than the Orowan stress; therefore sub-boundary hardening gives an important means of strengthening the creep resistance of high Cr ferritic steels [59].

The lath and block boundaries provide the sub-boundary hardening given by:

$$\sigma_{sg} = \frac{10Gb}{\lambda_{sg}} \quad (2.39)$$

Where λ_{sg} is the short width of an elongated subgrain. [58]

2.5.4. Microstructural Evolution

During creep, the microstructure of the material does not remain constant. The microstructural evolution and changes are the result of the interactions between structural defects and the result of phase transformations [53]. These processes are temperature, time and stress dependent and can lead to a complex relationship between the microstructural evolution and the creep process.

In the first stages of creep, a change in dislocation configuration occurs, followed by slow changes in dislocation structures. This microstructural evolution is exhibited in the forms of; lath widening, dissolution of prior austenite grain boundaries, the emergence of subgrains, the change in the type, size, shape, and amount of precipitates [11] and precipitation of new phases [51]. Most of these phenomena lead to a degradation of creep strength while some exhibit strengthening factors. Microstructural evolution and degradation would eventually lead to the formation of micro voids at creep boundaries and result in the failure of the material [49].

2.5.4.1. Martensitic Lath Widening

In the first stages of creep, a change in dislocation configuration occurs, followed by slow changes in dislocation structures. Laths would widen over creep exposure until they finally evolved into subgrains. The mechanism of lath widening occurs with the bulging of local parts of the lath boundary, then boundary migration. Repeated bulging and migration eventually results in the migration of the whole lath boundary. It is then obvious that lath widening can be prevented by decorating the lath boundaries in precipitates [61]. The precipitates found on lath boundaries are the large $M_{23}C_6$ precipitates. When bulging, the boundary will enter into the inside of the adjacent lath, where its only resistance is met by small MX particles, this growth of lath boundaries will only stop when it hits another boundary [51] which results in large laths tending to grow and narrow laths tending to shrink. The driving force for lath widening has been identified as the strain accumulated by the martensitic transformation, or due to creep deformation [61].

2.5.4.2. Dissolution of Prior Austenite Grain Boundary

After long term creep the prior austenite boundary would gradually become invisible and could only be detected by tracing the precipitates that had previously formed on them. These boundaries are in fact walls made of dislocations that are able to trap and eject other dislocations. Hence the prior austenite grain boundary will disappear by the movement of dislocations out of this boundary [51]. Laves phase is found to form on these boundaries due to the presence of Mo and W found on the prior austenite grain boundary.

2.5.4.3. Emergence of the Subgrain

The initial microstructure is a very regular martensitic lath structure but after creep this lath microstructure turns into subgrains. Subgrain boundaries are major obstacles against the glide of

mobile dislocations and hence, subgrain hardening is an important creep strengthening mechanism against dislocation creep [62].The formation of low angle grain boundaries increase during the creep lifetime, causing a continuous decrease in the dislocation density [48]. Over long term creep, deformation and high temperatures will increase the number of mobile dislocations. This means higher interactions between dislocations and thus an acceleration in the formation of subgrains [51].

2.5.4.4. Growth of the Subgrain

It has been observed that subgrain size increases with long term creep exposure [63]. This growth in the subgrain removes the pinning effect against dislocation movement, causing a loss in creep strength. Maruyama [62] stipulates that in long term creep the subgrain coarsening is controlled by the stability of precipitates rather than from creep deformation. It was observed that the precipitates in contact with the subgrain grow faster than those that are not in contact with a subgrain [64]. This results in the loss of stabilizing precipitates and further results in subgrain coarsening. There will then be a migration of the subgrain boundary away from the precipitates, resulting in the loss of correlation between the precipitates and the subgrain boundaries [48].

2.5.4.5. Solid Solution Depletion

Solute atoms in solid solution gather to make precipitates, this formation reduces the drag resistance against dislocations and as a result the material is weakened. It has been clear that during creep exposure, at temperatures around 600–650°C, most of the Mo and W in the steels will precipitate as Laves phase. The dominating opinion is that this would cause creep instability in the steels because the solid solution strengthening effect from Mo and W would be lost; however this is counteracted in the hardening effect of the precipitation of laves phase. As a result there is no loss of overall creep strength [65].

2.5.4.6. Precipitation

Precipitation is where elements which are solute in the base material precipitate out of the solution. The driving force for the process comes from the favourable atomic bonding of the cluster atoms and is expressed in terms of the supersaturating of the original solution. A number of different types of precipitates will occur; precipitates at subgrain boundaries will stabilize the structure against microstructure coarsening, whereas precipitates within the bulk will influence the deformation resistance of the grain itself. During creep deformation, precipitates may nucleate, grow, coarsen and dissolve [17]. When looking at the life cycle of a precipitate, a traditional categorization is often made into three stages of precipitation: Nucleation, growth and coarsening. [10]

Nucleation

This is the first step in the formation of a new precipitate. Nucleation is a stochastic process driven by the reduction of the Gibbs free energy [53], in which a new phase will form within an existing phase. The precipitation of second phase particles in multiple component material is due to the partitioning of the individual components in the different phases of this material. These precipitates can nucleate in the grain or at defects such as grain boundaries, dislocations of stacking faults [17].

Growth

Growth is the controlled attachment of atoms onto an existing precipitate. It is a deterministic process that is driven by the chemical and/or mechanical forces. After nucleation a stable nucleus is embedded in a still supersaturated matrix. It will then be surrounded by a concentration gradient of solute atoms, which provides the driving force for solute diffusion. This concentration gradient gives rise to its growth. When the precipitate grows, solute atoms must be supplied from the matrix, which makes this a diffusion controlled process [10]. Figure 18 shows the rate of movement of the phase interface, *v*, due to the diffusional flux of solute atoms towards and into the growing precipitate [10]. Once the precipitates have grown to a certain size, there will be a depletion of solute atoms from the matrix, and as a result there is a reduction of the rate of growth [14].

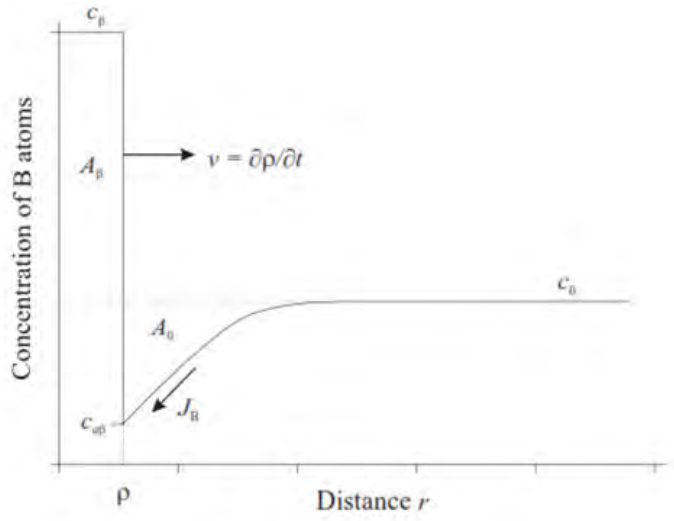


Figure 18 - Concentration profile of solute atoms around the moving interface of a growing precipitate [10]

Zener proposed a simple approximation for calculating the precipitate growth, the basic assumption is that the concentration profile can be approximated by a linear profile. It is defined that the mass increase in the precipitate is balanced by the decrease of mass in the matrix, i.e. the two areas are identical. Thus the velocity of the interface could be calculated if the concentration of precipitate atoms was known. He gave the position and velocity of the interface as: [10]

$$\xi = \sqrt{{\xi_0}^2 + S^2 D t} \tag{2.40}$$

$$\dot{\xi} = \frac{D\,S^2}{2\,\xi} \tag{2.41}$$

Where *S* is the dimensionless supersaturation:

$$S = \frac{c_0 - c_{\alpha\beta}}{c_\beta - c_{\alpha\beta}} \tag{2.42}$$

In the quasistatic approach that deals with a spherical precipitate the interface is assumed to be stationery and the precipitate itself is assumed to be a point source and sink for solute atoms. This

solute profile is given by Fick's second law in spherical precipitates. This gives the position and velocity of the interface as [10]:

$$\rho = \sqrt{\rho_0{}^2 + 2SDt} \tag{2.43}$$

$$\dot{\rho} = \frac{DS}{2\,\rho} \tag{2.44}$$

Coarsening

Coarsening refers to the evolution of a system of precipitate particles embedded in a matrix. It is the dissolution of smaller particles in favour of larger ones. This coarsening is driven by a curvature induced pressure where small particles have a higher surface to volume ratio than larger particles, and therefore are less stable than larger particles. An increase in the mean particle size will therefore reduce the total free surface energy of the system and thus is the driving force for coarsening [48].

Coarsening of the precipitates is unavoidable at high temperatures and leads to an increase in the mean spacing between the particles, allowing for easier flow of dislocations through the material [66]. It therefore has a profound effect on the properties of materials and will be a main factor of creep strength degradation. It also has a high contribution to the creep rate increase in the tertiary stage. It involves or initiates other creep loss factors [10] in the form of microstructural evolution, such as the coarsening of lath and block structures and a loss of sub-boundary hardening [59].

Figure 19 clearly represents coarsening where it can be seen that the small precipitates are shrinking while the large precipitates are growing and absorbing the small precipitates as a) progresses to d). This also shows how the total volume fraction of the precipitates does not change. [67]

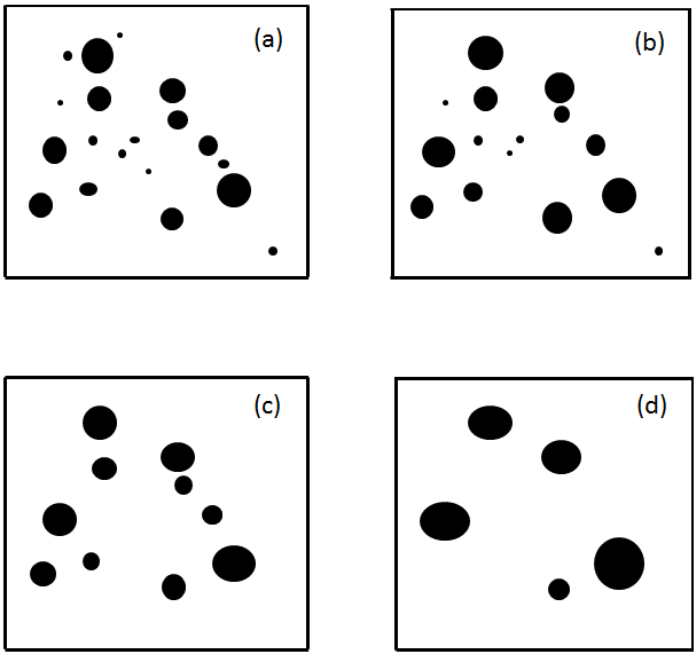


Figure 19 - Ostwald ripening as time increases from (a)-(d) [67]

The theory of diffusion controlled precipitate coarsening was developed by Lifshitz and Slyozov [68] and is given as:

$$r^{-3} - {r_0}^{-3} = \frac{8DC_e\gamma {V_m}^2 t}{9\mathrm{R}T} \tag{2.45}$$

Where:

r is precipitate radius

r_0 is original precipitate radius

C_e is equilibrium concentration of the precipitate

γ is particle-matrix interface energy

V_m is molar volume of precipitate particles

The most striking aspect of the equation is that the mean radius cubed varies with time, as opposed to the radius squared in the growth calculations. Coarsening is thus shown to be a much slower process than precipitate growth.

Coarsening theory also predicts that particles will take on a time independent size distribution as shown in Figure 20, with limiting particle size of r_{max} = 1.5 r_{mean} [69]

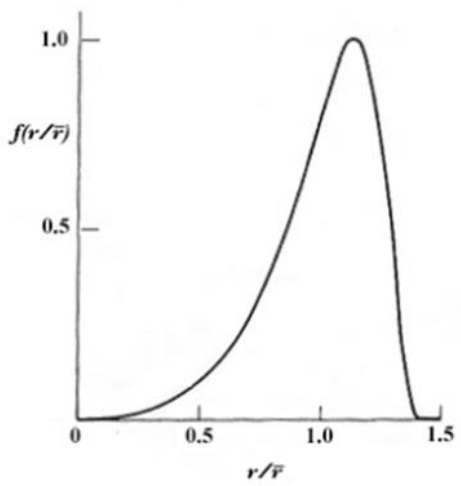


Figure 20 - Steady state particle distribution for diffusion controlled coarsening [69]

The average growth or shrinkage rate, *dR/dt*, of a typical particle depends on its size relative to the average radius particle. If R < R_{ave} then the particle will shrink or when R > R_{ave} the particle will grow. Where R/R_{ave} = 1 is known as the critical particle radius, R_{crit}. In more complex systems and differing situations R_{crit} will tend to deviate from 1 [69].

The various parameters and constants in equation (2.45) above can be combines to give:

$$r^{-3} - {r_0}^{-3} = K_c t \tag{2.46}$$

Where K_c is the rate constant for coarsening. [48]

2.5.4.7. Evolution of Known Precipitates

The evolution of the previously discussed precipitates results in a loss of creep strength over long periods of time; their evolution is summarized in Table 3:

Table 3- summary of basic precipitates in 9-12% Cr Steels [48]

Precipitate	**Formed during**	**Preferred precipitation**	**Characterized by**	**Main contribution to**
$M_{23}C_6$	Tempering	Former austenite grain and martensitic lath boundaries	Medium growth rate during creep	Precipitation strengthening
MX	Solidification and tempering	Austenite grain boundaries, inside subgrains and martensitic laths	Restriction of grain growth	Precipitation strengthening
Z Phase	Long-term creep exposure	On $M_{23}C_6$ and MX precipitates	Dissolution of MX precipitates	Lowering precipitation strengthening
Laves Phase	Long-term creep exposure	On $M_{23}C_6$ at grain and subgrain boundaries	Fast coarsening above 600 C	Lowering precipitation strengthening and decrease in solid solution strengthening

$M_{23}C_6$ carbides are formed right after the tempering process. These particles stabilise the microstructure by pinning subgrains and so reduce the effectiveness of the recovery mechanism. [57]. They have been found to coarsen at a relatively large rate. Thus the growth of the precipitates should be controlled by adding Co and B to retard the growth, so that they can still pin subgrain boundaries. $M_{23}C_6$ also change chemical composition throughout long term creep. The Cr content was found to increase with increasing creep time [51]. This effect is more prevalent in low nitrogen steels [41]. The area fraction of $M_{23}C_6$ stays constant throughout creep at about 9-10%, with the highest volume fraction this is the most important precipitate [48].

MX carbonitrides, in contrast to $M_{23}C_6$, are much more stable. They show little change , even under creep loading their volume fraction was reported to stay constant, with no change in size or chemical composition also reported [70]. High creep strengths are due to the high stability of these precipitates, such as VX hardening [51]. However there is still some debate on coarsening, a paper by [57] states that MX precipitates still coarsen, while the study done by [48] states that they barely coarsen at all. The area fraction of MX stays constant throughout creep at about 0.5% [48].

Z-phase, or modified Z-phase in martensitic steels, precipitate very slowly as finely distributed rod like particles during the creep process. It leads to the partial or complete disappearance of MX particles and suppression of creep hardening. It was proposed by [71] that the nucleation mechanism was the diffusion controlled transformation of MX with chromium atoms into cubic or tetragonal Z phase, with diffusion explaining the low precipitation rate. Once it has nucleated it can grow fast, and large Z phase

particles can be observed after relatively short exposures. It grows by the dissolution of VN precipitates by picking up Cr from the matrix. This formation causes a vanadium free zone to form around the Z-phase which further causes a drastic loss in creep strength. Observations of large and small Z phase particles over long exposure time show that the nucleation of Z phase is both slow and continuous [71]. The nucleation sites are lath boundaries and prior austenite grain boundaries [71]. As Cr is the most influential element in Z phase growth, steels above 10.5% Cr content show accelerated growth rates.

Laves phase particles nucleate and grow during the creep process. They grow into a square shape with very large sizes. Their favoured nucleation sites are lath boundaries [72] and are unequally distributed along these boundaries. Precipitates facilitate the formation of laves phase by providing nucleation sites so they are then observed to grow on the $M_{23}C_6$ precipitates. It is a cause of solid solution depletion in the formation of laves phase, which decreases creep resistance due to depletion of Molybdenum (Mo) and Tungsten (W) [41]. They quickly grow to large sizes during creep where they offer little creep resistance and can act as cavity triggers [51]. The area fraction of laves phase linearly increases from 0.25% at 12500 h to 1.3% at about 140000 h. [48]

2.5.4.8. Void Nucleation and Crack Nucleation

The synergistic effects of all microstructural evolution will lead to the weakening of the material and direct it towards cavitation [73]. Cavities will gradually form micro cracks by interlinkage and thus, the rupture of materials under service conditions is heavily dependent on the formation and growth of cavities. The emergence of micro voids means that the material in service is damaged seriously. As voids grow they elongate perpendicular to the tensile axis and develop into cracks, mainly along the grain boundaries. However in a uniform medium the cavities would extend more rapidly in the tensile axis direction than in the transverse direction [4].

There are two groups of cavities that can be observed in a creep damaged material. The first is large cavities (>1 μm) that were present from the manufacturing process. The second type is small cavities (<0.6 μm) that are produced during creep [73]. The cavities present in the steel from the manufacture process grow with creep exposure over time. A recent study [73] has stated that the growth rates of these initial cavities is higher than the formation of new cavities, and as a result they have a much higher volume fraction than creep cavities. This result indicates that final fracture may actually be attributed the cavities produced during the manufacture process [73].

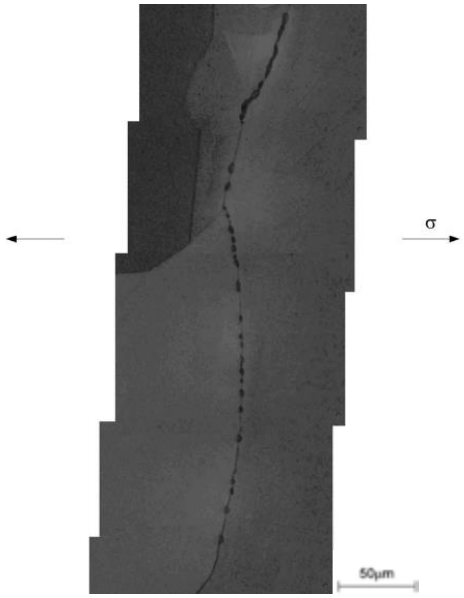


Figure 21- Creep cavities forming on boundaries perpendicular to the applied stress [47]

It is still not well established by which mechanisms cavities nucleate [47]. It has generally been observed that cavities frequently nucleate on grain boundaries, particularly on those transverse to tensile stress, as illustrated in Figure 21. The nucleation theories for cavity formation fall into several categories that are illustrated in Figure 22. Figure 22 (a) theory is that grain boundary sliding can open up a void at triple points at boundaries [73] [47]. Figure 22 (b) shows a void forming from vacancies condensing in highly stressed regions, usually at the transverse grain boundaries [73] [47]. Figure 22 (c) shows cavity formation at the head of a dislocation pileup [47] and there is the possibility of a combination of all three mechanisms occurring.

In commercial alloys these cavities appear to be associated with second phase particles along the grain boundary as shown in Figure 22 (d) [47]. This could be because the interface between hard particles and the matrix act as void nucleation sites due to the lack of cohesion between the two materials [73] and because second phase particles can result in stress concentrations upon application of a stress.

Another nucleation mechanism that may be important, as damage progresses in a material, is the stress concentration that arises around existing cavities. Cavity nucleation by vacancy accumulation appears to require significant stress concentration around the hard particles [47]. The initial stress concentration at the cavity 'tip' is a factor of three larger than the applied stress, and even after relaxation by diffusion, the stress may still be elevated [74], which leads to increased nucleation of existing cavities.[47]

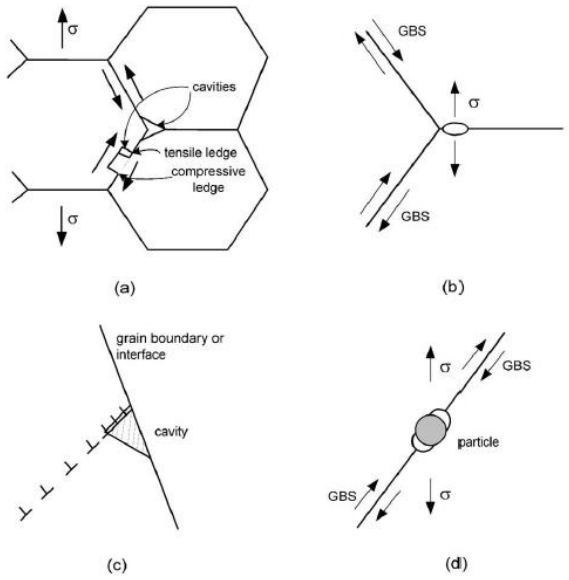


Figure 22 - Mechanisms of cavity nucleation: (a) cavitation from boundary sliding at triple points (b) Cavity nucleation from vacancy condensation (c) cavity formation from Zener-Stroh Mechanism (d) Cavity formation around a hard particle in conjunction with other mechanisms. [47]

Once these voids have nucleated they grow in many different ways: diffusion, grain boundary sliding, constrained diffusional growth, plasticity and a coupled effect of diffusion and plasticity [73]. While adding precipitates to boundaries slows down creep, the growth rate of a cavity is unaffected, this suggests at diffusion rather than plasticity as the main controlling mechanism for cavity growth [47].

Growth of Cavities by Grain Boundary Diffusion

The cavity growth process at grain boundaries, at elevated temperatures, has long been suggested to mainly involve vacancy diffusion. Diffusion to the void occurs by the transport of vacancies along a grain boundary and then by surface migration along the cavity wall [47]. The activation energy was found to be similar to that of grain boundary diffusion, resulting in it being the rate controlling mechanism [4]. Hull and Rimmer were one of the first to propose a mechanism by which diffusion leads to cavity growth of an isolated cavity in a material under an external stress. A stress concentration is established just ahead of the cavity. This leads to an initial negative stress gradient, meaning stress decreases away from cavity; however, a positive stress gradient is suggested to be established due to relaxation by plasticity. The equations derived were:

$$J_{gb} = -\frac{D_{gb}}{\Omega kT}\nabla f \tag{2.47}$$

Where J is the flux along the grain boundary, Ω is the atomic volume and ∇f is the gradient of chemical potential. The gradient of chemical potential is then given as:

$$\nabla f \sim \frac{\Omega}{\lambda_s}\left(\sigma - \frac{2\gamma_m}{a}\right) \tag{2.48}$$

Where a is the cavity radius, σ is the applied stress and λ_s is the cavity separation. $\Omega\sigma$ would correspond to the chemical potential at the grain boundary, while $\frac{\Omega 2\gamma_m}{a}$ corresponds to the chemical

potential along the cavity surface. In large voids the chemical potential along the surface can be neglected [4].

Combining equations (2.46) and (2.47) above gives the growth rate of voids as:

$$\frac{da}{dt} \cong \frac{D_{gb}\delta\left(\sigma - \frac{2\gamma_m}{a}\right)\Omega}{2kT\lambda_s a} \tag{2.49}$$

Where δ is the grain boundary width.

This theory assumes that the surface diffusion of a vacancy would be rapid enough for the void to maintain an almost spherical shape [4]. The most basic concept of applying this equation is illustrated by the Figure 23 below, where the separation distance between voids is now the grain size when there is a single void in a grain.

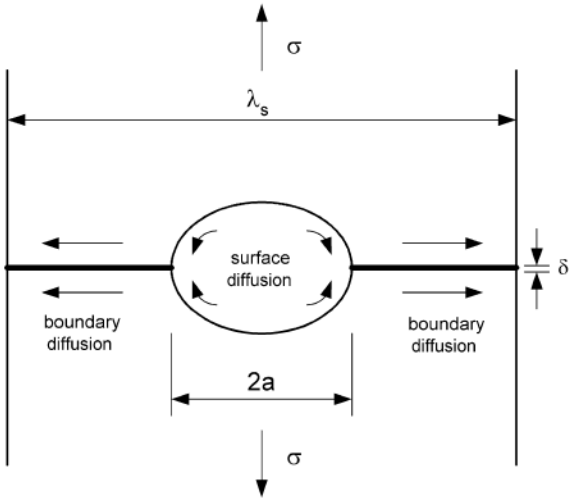


Figure 23– Cavity growth from diffusion along the grain boundaries [47]

Plasticity

For completion sake, cavities can grow exclusively by superplasticity. Initially it was proposed that the creep controlled cavity growth model was based on the idea that cavity growth during creep should be analogous to the model for a cavity growing in a plastic field. Cavity growth, according to this model, occurs as a result of creep deformation of the material surrounding the grain boundary cavities in the absence of vacancy flux. This mechanism becomes important under high strain rate conditions where significant stress is realized. The cavity growth according to this model is given as

$$\frac{da}{dt} = a\dot{\varepsilon} - \frac{\gamma}{2G} \tag{2.50}$$

It has been suggested on occasion that the observed creep cavity growth rates are consistent with plasticity growth but not always obvious that constrained diffusional cavity growth is not occurring, which is also controlled by the plastic deformation. [47]

Damage

As cavities appear and grow they cause a loss in the load bearing area. This is referred to as damage and is the most common example of creep damage. The tertiary creep stage is due mostly to this increase in damage. The creep damage theory assumes that the extent of creep damage can be represented by a single parameter, ω. The creep rate is then said to a function of the stress and this damage parameter:

$$\dot{\varepsilon} = f(\sigma, \omega) \tag{2.51}$$

It is then assumed that the accumulation of damage is represented only by the growth of voids and other changes such as size and growth of precipitates are not considered able to affect the damage parameter. Therefore ω is only a representation of the fraction of area occupied by voids, cavities and cracks along the grain boundaries. This then leads to the concentration of stress in the equation below.

$$\sigma = \frac{\sigma_0}{1 - \omega} \tag{2.52}$$

This equation shows that creep cavitation increases the applied stress when the damage parameter evolves from 0 to 1 as these cavities nucleate and grow [4] [22]. This is however not the only form of stress concentration considered in this study, but also the varying stress fields forming around the void as a stress is applied; this is more related to the shape of the void rather than the loss of load bearing area.

2.6. Dislocations

One of the other important creep mechanisms previously mentioned is dislocation creep. This is where the material deformation is caused by the motion of dislocations through the crystal lattice. The study of their generation and motion becomes very important in predicting the creep lifetime of a material when the stress is high enough to cause the flow of dislocations. The movement of these dislocations is one of the most important causes of plastic deformation of crystals at stresses lower than the shear strength [53].

Dislocations are line defects that are found within the crystal structure and are quantified with a Burgers vector. The presence of these dislocations produces distortion in the crystal lattice, which then has a strong influence on the strength, plasticity and creep deformation of the material [18]. This deformation becomes dominant over diffusional creep when the applied stress is high enough to cause dislocation movement to be faster than that of vacancy movement.

There are two distinguishable types of dislocations present in the crystal structure, edge dislocations and screw dislocations, shown in Figure 24. Edge dislocations, Figure 24 (a), are line dislocations within the crystal lattice where a line of atoms terminate. The stress field produced by these dislocations comprises of tensile, compressive and shear stresses [18]. Screw dislocations, Figure 24 (b) form a spiral ramp of crystal planes and have no distinct slip plane. These types of dislocations only produce shear stresses [75].

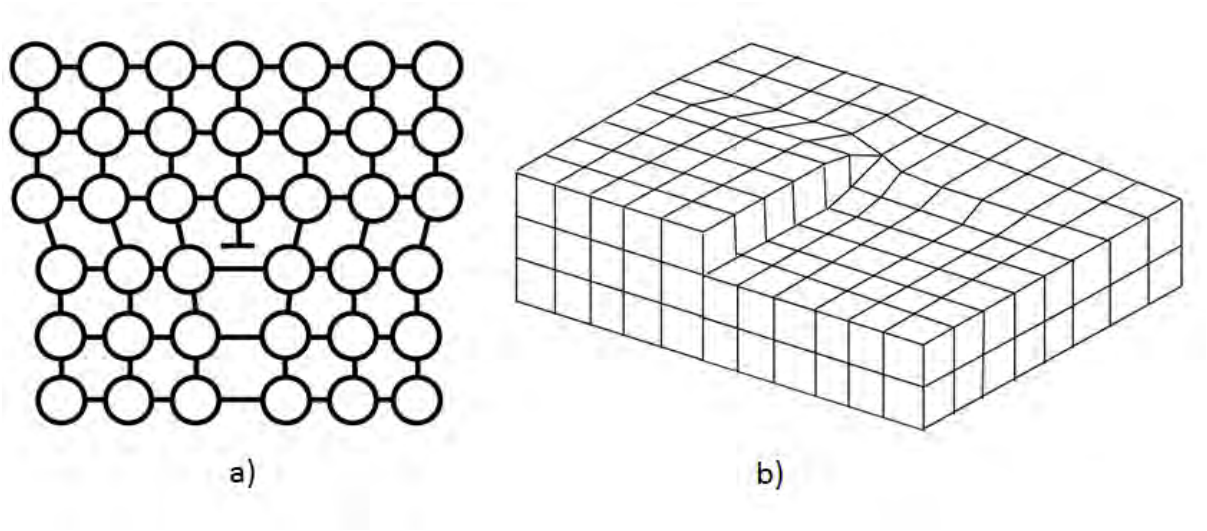


Figure 24 – Edge dislocation (a) and screw dislocation (b) [17]

2.6.1. Movement of dislocations

Dislocations are able to move by both climb and glide (or slip)[18]. For glide motion, the atoms can be rearranged under an applied stress and the dislocation line can move through the crystal. Edge dislocations have a distinct slip plane within which they can move, this slip plane is normal with the Burgers vector for the edge dislocation [76]. In a screw dislocation the Burgers vector and dislocation line is parallel and therefore is not restricted to a unique plane [76]. The second mechanism of dislocation movement is climb. Here the atoms are removed from or added to the bottom of the extra plane. This accomplished due to the emission or absorption of vacancies from the dislocation core. The vacancy either arrives at the slip plane and is absorbed, or is removed from the slip plane. In either instance the dislocation climbs one atomic place to accommodate this diffusion [22].

Not all dislocations within a material can move freely. Those that are not bound with boundaries are termed free dislocations and only a fraction of these have the potential of moving when a stress is applied, these are referred to as the mobile dislocation density. At any instant all free dislocations are considered to be part of a dislocation network and the density of moving dislocations represents only a very small fraction of the total free dislocation density. The density of dislocations is calculated using the dislocation link lengths, where only links above a certain size are considered [22] [77].

2.6.2. Dislocation movement during creep

Dislocation density varies during the creep process. In the early stages of creep there is an increase in the dislocation density as a result of the stress increase in the crystal. The increase in dislocation density translates to a reduction in the mobility of the free dislocations as dislocations act as obstacle to other dislocations movement. This lack of dislocation movement ultimately leads to the hardening effect on the material in the primary creep stage. After the initial decrease in creep rate, a dynamic balance forms between the generation and annihilation of dislocations. This means there is a balance between the hardening and softening effects in the microstructure, leading to the secondary stage of creep [66].

Microstructural evolution leads to the greater dislocation numbers, mobility and a greater creep rate, ultimately forming the tertiary stage of creep. The microstructural evolution that affects dislocation

creep is identified as subgrain evolution, precipitation and precipitate coarsening. Subgrain boundaries are dislocations that have arranged into a network. This result is the tendency of homogeneously distributed dislocations to form modulated structures. The configuration of these structures is heavily influenced by grain size and grain misorientation [22]. The boundary can be thought of as a hard region, while the interior of the subgrain can be thought of as soft. The strengthening effect of subgrain boundaries is that they act as obstacles to mobile dislocation movement [65]. Thus as sub grains coarsen, the distance between subgrains increases and the material is softened.

Precipitates contribute to the overall creep strength by stabilising the initial microstructure through the pinning of grain and sub-grain boundaries, as well as by acting as obstacles to dislocation movement within the grain [1]. Therefore with coarsening leading to larger precipitate particles and greater inter-particle spacing, there is a resulting loss of creep strength. Precipitation like that of Modified Z-phase is detrimental to the pinning effect of mobile dislocations by means of forming large precipitates that offer little resistance to dislocation movement as well as dissolving the highly beneficial MX precipitates to form and grow [78].

3. Model Overview

The model to be introduced in this study is a spatially resolved model that will draw influence from the existing Nabarro-Herring model, specifically the derivation of the model in section [2.3.1] in the literature review. This model is run using MATLAB as opposed to any existing FEM software packages as they are too rigid and do not easily allow changes to the existing code. By using MATLAB a complex microstructure can be introduced that can easily be adapted later to suit specific situations. At this stage it is used to introduce an idealistic heterogeneous microstructure similar to that of high chromium steels and their respective microstructural changes. At later stages this can be changed to suit other real life scenarios.

The microstructure introduced is complex in nature. The entire microstructure will consist of a number of different elements, namely:

- Grains
- Grain boundaries
- Vacancies
- Precipitates
- Voids

Note that dislocations are left out in this study. These elements would be represented on an elemental matrix with a corresponding numerical value. Each of these elements would have their own set of properties that would be superimposed over the value at each respective node of the matrix. Vacancies are rather represented as a property of each element as the concentration of vacancies at each node. The characteristics of each of the represented elements in the material are controlled by their respective properties. The properties covered in this study are:

- Young's modulus
- Poisson's ratio
- Diffusion coefficient
- Vacancy concentration
- Stress state

The level of interaction of all the microstructural elements is dependent on their properties. Young's modulus and Possion's ratio between differing elements would cause stress discrepancies between interacting elements. Diffusion coefficients would alter the creep rate through a particular element, and the stress state and resulting vacancy concentration would heavily influence both the generation rate and the movement of vacancies

The first major deviation from any standard model is to set up a vacancy concentration field in two dimensions. By applying Fick's second law, presented in section [2.2.1.2], the formation of concentration gradients within a material can be tracked the moment a stress is applied. This tracking is a further improvement upon the Nabarro-Herring equation, which made the assumption that a steady state concentration gradient had already been formed within the material. The resulting vacancy concentration field in two dimensions provides a more complex and spatially differing concentration gradient that would affect the diffusion rates of vacancies through the lattice.

Fick's second law is used to determine both the changes in concentration and the actual number of diffusing vacancies at each time step. Fick's first law, described in section [2.2.1.1], is used only at the boundary layer to calculate all vacancies arriving or leaving these boundaries. By applying Fick's laws

in this manner the total strain can subsequently be calculated by using these vacancies and the same derivation path used in the Nabarro-Herring model (which was described fully in section [2.3.1]).

At this level the Nabarro-Herring model has been replicated and expanded upon by using Fick's second law to form a more complex, yet realistic, vacancy concentration field to describe the diffusion. But this is all done for a homogenous microstructure where the stress field is uniform. When the heterogeneous microstructure is introduced, the stress fields throughout the material would no longer be uniform, which would show further deviation from the Nabarro-Herring model.

Stress fields form as a concentration around particles that are harder or softer than the matrix material. In this study these are the precipitates and voids respectively, which are present in high chromium steels. These two microstructural elements are introduced in separate systems to evaluate their influences on diffusional creep. The values for the generated stress fields are taken from the article discussed in section [2.4]. This gives the stress gradients that would form inside the lattice of the material. The resulting gradients are also resolved spatially to give a stress state at every nodal point in the matrix. The introduction of these stress fields shows how the inclusion of precipitates is actually able to affect the diffusion rates. This is to give a definitive reason that precipitates do, in fact, slow down diffusional creep rates with the long range effects of their stress fields. In terms of voids, their stress concentration was already considered in terms of the damage equation taken from section [2.5.4.8]. However, when using the stress field equation from section [2.4] the damage equation is not as influential. This is due to the stress field equations showing that the increased stress due to the loss of load bearing area is carried by high stresses next to the void.

The types of precipitates and their location is placed according to where they would most likely be found in a high chromium steel (described throughout section [2.5]); where the large precipitates are most likely found at the grain boundaries, and where voids most likely found at the meeting points of grains. The stress at grain boundaries is most responsible for the generation of vacancies, thus the interaction of their stress fields with the grain boundaries is most important to this study. The influences of this stress would alter vacancy generation in accordance with the stress dependent equations described in section [2.2.4], and would thus affect the vacancy concentration fields.

Including a stress gradient throughout the material raises an issue that has to be addressed; stress gradients should affect the diffusion rates within the body of the grain. Thus the concentration change due to stress gradients has to be included into Fick's second law. This requires the derivation of the stress' effect on the chemical potential of the system and is addressed in the next section.

Lastly, the evolution of the microstructure also has an effect on the creep life of high chromium steels. In this study it is introduced as the coarsening of precipitates, described in section [2.5.4.6], and the growth of voids, described in section [2.5.4.8]. In both cases the microstructural evolution will constantly alter the stress fields and thus affect diffusion throughout the materials creep lifetime. The secondary effect of damage due to void growth is brought in at a much later stage, when the voids are large enough to induce stresses greater than that supported by the stress field equations from section [2.4].

The final step of the project is to incorporate precipitates and voids, as well as their respective evolutions, into the same system. This is to give a clearer representation of their combined effect, as well as providing the first steps to a predictive model of diffusion creep deformation in service material.

4. Diffusional Flux of Vacancies in a System with Stress Gradient

To add the stress gradients into the model, the stresses at the grain boundaries are not the only consideration. How these stress gradients might affect diffusion within the body of the grain also has to be considered. As vacancy generation is heavily dependent on the stress state at the grain boundaries, the main focus was on the stresses in these regions. However, stress gradients are also induced within the lattice of the grain and their effect on diffusion should not be neglected. These internal stress gradients would not alter the generation or annihilation of vacancies, but would rather affect the rate of diffusion by means of altering the chemical potential. Where higher stresses would cause a higher concentration of vacancies at the grain boundaries, in the body of the grain, they would increase the diffusion rate. Conversely, lower stresses would decrease the diffusion rate.

The effect of stress in a uniform field can be easily analysed by including the stress changes in the stress dependent equations (2.11) and (2.12). However, with the inclusion of second phase particles and the stress fields they produce, the effect these gradients have within the body of the grain would have to be derived.

These stress gradients are theorized to have little effect when compared to temperature or stress state at the boundaries, but it would none the less have an effect on the local chemical potential. Thus, the diffusional flux of vacancies in a system with stress gradients has to be derived and added to Fick's second law for changing concentrations. The equation is derived by combining two different scenarios; firstly, from the change in chemical potential due to stress and then from the change in concentration due to a chemical potential change.

The chemical potential is derived from the regular solution model for Gibbs energy of a two part system, found in [10], and adding the mechanical energy of the vacancies under stress. The mechanical energy on the stress is the work per vacancy, $\sigma\Omega$, times the total volume of all the vacancies from section [2.2.4]. The total resulting equation is given as:

$$G_{v,mol} = N_{avo}.Z_l\left[\frac{1}{2}\varepsilon_{MM}{X_M}^2 + \frac{1}{2}\varepsilon_{vv}{X_v}^2 - \varepsilon_{Mv}X_M X_v\right] + RT[X_v \ln X_v + X_M \ln X_M] + N_{avo}.\Omega_{at}.\sigma.X_v \quad (4.1)$$

Where

$G_{v,mol}$ is the Gibbs energy of the system
N_{avo} is Avogadro's number
Z_l is a coordination number of nearest neighbour bonds of a single atom
M represents matrix atoms
v represents vacancies
$\varepsilon_{MM}, \varepsilon_{vv}\ and\ \varepsilon_{Mv}$ are the binding energies between matrix atoms, between vacancies and matrix and vacancies respectively
X_M is the concentration of matrix atoms
X_v is the concentration of vacant atoms
R gas constant
T Temperature
Ω_{at} is the atomic volume
σ is the applied stress

From here only the influence on the vacancy constituent of the equation is needed. Therefore, the Gibb's energy equation is reduced to only be expressed in terms of the concentration of vacancies, X_v:

$$G_{v,mol} = N_{avo}.Z_l\left[\frac{1}{2}\varepsilon_{MM}(1-X_v)^2+\frac{1}{2}\varepsilon_{vv}{X_v}^2-\varepsilon_{Mv}(1-X_v)X_v\right] \\ +R.T[X_v\ln X_v+(1-X_v)\ln(1-X_v)] \\ +N_{avo}.\Omega_{at}.\sigma.X_v \tag{4.2}$$

It is now possible to derive the chemical potential of a vacancy, where the chemical potential of a system is calculated from the change in Gibb's energy due to a change in the concentration of the constituent. In this case it is the change in Gibbs energy of the vacancies over the change in concentration of the vacancies:

$$\mu_{v,mol} = \frac{\partial G_{mol}}{\partial X_v} \tag{4.3}$$

The result for the chemical potential of vacancies is given below as equation (4.4):

$$\mu_{v,mol} = N_{avo}.Z_l\left[\varepsilon_{MM}(1-X_V)(-1)+\varepsilon_{vv}X_v-\varepsilon_{Mv}[(1-X_v)+X_v(-1)]\right] \\ +R.T[1+\ln X_v+(1-X_v).\tfrac{1}{1-X_v}.(-1)-\ln(1-X_v)] \\ +N_{avo}.\Omega_{at}.\sigma \tag{4.4}$$

This equation is split into four distinct parts that each affects the chemical potential of the vacancies in the system:

$\mu_{v,mol} = N_{avo}.Z_l[\varepsilon_{MM}-\varepsilon_{Mv}]$	describes a constant
$+N_{avo}.Z_l[\varepsilon_{MM}+\varepsilon_{vv}-2\varepsilon_{Mv}]X_v$	describes the effect of the concentration gradient
$+R.T\,[\ln X_v-\ln(1-X_v)]$	describes the entropy of mixing of the system
$+N_{avo}.\Omega_{at}.\sigma$	describes the effect of stress

This chemical potential can then be worked back into Fick's laws of diffusion.

Firstly, the flux of vacancies is seen to not only be influenced by the concentration gradients as previously described but rather by a number of factors represented as the chemical potential. The most basic representation of the chemical potential is affected by the stress, temperature and concentration. At equilibrium the chemical potential is given as zero:

$$\mu_{v,mol}(\sigma,T,X_v) = 0 \tag{4.5}$$

Having this applied to Fick's first law of diffusion, equation (4.6), with respect to the chemical potential gives:

$$J = -D_\mu\left(\frac{\partial \mu_v}{\partial X_v}.\frac{\partial X_v}{\partial s}+\frac{\partial \mu_v}{\partial \sigma}.\frac{\partial \sigma}{\partial s}+\frac{\partial \mu_v}{\partial T}.\frac{\partial T}{\partial s}\right) \tag{4.6}$$

Due to the small sizes and operational conditions of the material, it is assumed that there is no temperature gradient throughout the specimen; as a result the effect of temperature change can be removed.

$$J = -D_\mu \left(\frac{\partial \mu_v}{\partial X_v} . \frac{\partial X_v}{\partial s} + \frac{\partial \mu_v}{\partial \sigma} . \frac{\partial \sigma}{\partial s} \right) \tag{4.7}$$

As the vacancy concentration is the main cause of vacancy diffusion in this model the equation is altered to represent the changes in terms of a change in concentration instead of change in the chemical potential. The diffusion coefficient with respect to concentration is given as:

$$D_c = D_\mu \left(\frac{\partial \mu_v}{\partial X_v} \right) \tag{4.8}$$

Which can be rearranged as:

$$D_\mu = D_c \left(\frac{\partial \mu_v}{\partial X_v} \right)^{-1} \tag{4.9}$$

Placing equation (4.9) into equation (4.7) gives:

$$J = -D_c \left(\frac{\partial \mu_v}{\partial X_v} \right)^{-1} \left(\frac{\partial \mu_v}{\partial X_v} . \frac{\partial X_v}{\partial s} + \frac{\partial \mu_v}{\partial \sigma} . \frac{\partial \sigma}{\partial s} \right) \tag{4.10}$$

By changing to concentration this simplifies the change in the concentration portion of the equation to be in line with that of Fick's first law. The additional part of the equation is then purely the effect of stress on the flux of vacancies. The resulting equation is:

$$J = -D_c \left(\frac{\partial X_v}{\partial s} + \left(\frac{\partial \mu_v}{\partial X_v} \right)^{-1} . \frac{\partial \mu_v}{\partial \sigma} . \frac{\partial \sigma}{\partial s} \right) \tag{4.11}$$

Going back to equation (4.4) and deriving the change in chemical potential with respect to changes in stress gives:

$$\frac{\partial \mu_v}{\partial \sigma} = N_{avo} . \Omega_{at} \tag{4.12}$$

Thus Fick's 1st law with the consideration of stress gradients is given as equation (4.13)

$$J = -D_c \left(\frac{\partial X_v}{\partial s} + \frac{N_{avo}}{E_{M,mol}} . \Omega_{at} . \frac{\partial \sigma}{\partial s} \right) \tag{4.13}$$

Where $E_{M,mol}$ is the molar enthalpy of the matrix $\left(\frac{\partial \mu_v}{\partial X_v} \right)$.

Altering Fick's 1st law into Fick's 2nd law is easily done by taking the equation of continuity as explained in section [2.2.1]. This gives the change in concentration with respect to time as equal to the change in flux with respect to position:

$$\frac{\partial c}{\partial t} = D_c \frac{\partial^2 c}{\partial s^2} = D_c \frac{\partial}{\partial s} \left(\frac{\partial c}{\partial s} \right) = -\frac{\partial}{\partial s} J \tag{4.14}$$

$$\frac{\partial c}{\partial t} = D_c \left(\frac{\partial^2 c}{\partial s^2} + \frac{N_{avo}}{E_{M,mol}} . \Omega_{at} . \frac{\partial^2 \sigma}{\partial s^2} \right) \tag{4.15}$$

Thus equation (4.15) gives the final equation that derives the diffusional movement of vacancies in a system with stress gradients. The output of the equation specifically gives the movement of vacancies only within the body of the grain.

5. Applied Creep Model

5.1. Scope of the model

The first step of creating the model is to specify the spatial range covered by the project. There are three spatial levels that could be used; the first level is the macroscopic level or engineering level. At this level there is a high enough volume of material being considered that it could be considered homogeneous. Here any deformation is described at scales ranging down to the millimetre, without explicitly taking underlying physical mechanisms into account. The second level is the mesoscopic level. This scale takes the internal structure into account, such as grains, grain boundaries and precipitates. Here each microstructural element is described by, and assigned, a set of physical properties. The length scale is micrometres to nanometres. The third level is the microscopic level and this consists of atomic structures and their interactions. Creep modelling is concerned with the microstructural constituents such as vacancies and dislocations, and their reactions and interactions within the material.

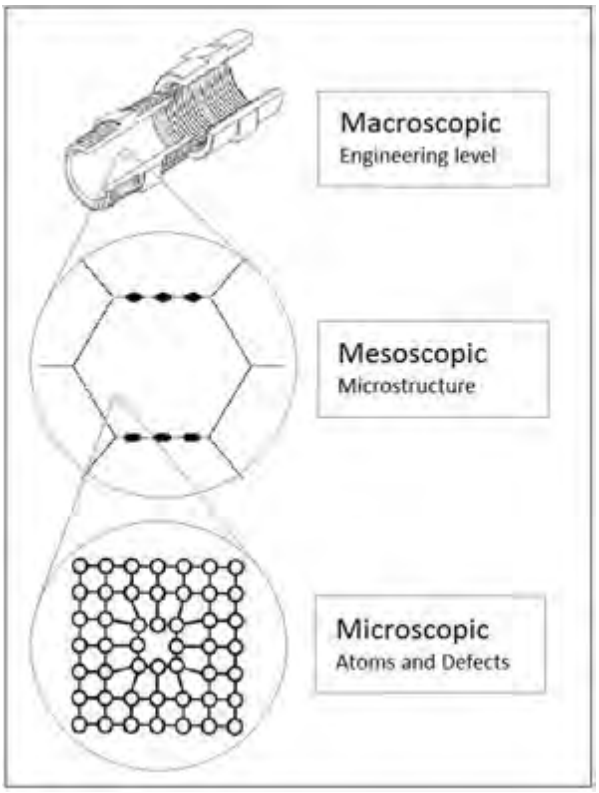


Figure 25 - Macroscopic, Mesoscopic and Microscopic levels [17]

The diffusional processes and resulting mass fluxes are caused by a migration of structural defects at the microscopic level, which describes, and determines, the deformation of the material; yet the resultant creep is calculated by taking these microstructural processes and averaging them over a larger volume, and thus this model is able to describe creep at the mesoscopic level.

By performing the simulations on small, ideal microstructures or situations it can satisfactorily provide a general view of the phenomenon. This was first accomplished with the original Nabarro-Herring creep model. That model only considered creep in a material consisting of perfectly cubic-shaped grains with no second phase particles. This current model is heavily influenced by the original Nabarro-Herring model with the inclusion of cubic grains. There are three reasons for this; first, cubic grains

eliminate the potential for any grain boundary sliding that would contribute to the overall creep deformation. Secondly, as only normal forces affect the generation of vacancies, for a cubic grain the forces are normal along the entire grain boundary, making the entire boundary an effective source of vacancies. Additionally the side grain boundaries have no normal force acting on them, making them an effective sink for vacancies. Lastly, the use of cubic grains is a way to validate this model against the original and accepted Nabarro-Herring theory by using the same initial conditions and achieving a similar result.

This project then addresses the effect that a heterogeneous microstructure would have on the diffusional creep rate of a material. This is done by including ideal situations of precipitates and voids into the system instead of the previous models that only considered a homogeneous microstructure in their approximations. The effect the heterogeneous microstructure has is approached by including the stress fields forming due to the difference in mechanical properties of all the included elements. Any changes in the stress distribution in the system would play a critical role in vacancy generation and diffusion. This would lead to changes in void nucleation, crack propagation and ultimately the overall creep rates. The initial stress variation would be caused by the presence of hard precipitates found within creep resistant steels. Only the $M_{23}C_6$ precipitates were considered as they are the largest and most prevalent precipitates at the grain boundaries. These stress fields would then be subject to change as the microstructure evolves, namely precipitate coarsening and the formation and growth of voids. The geometry of the precipitates and voids are taken as being ideally spherical during the entire creep process.

The code is then implemented in stages so that the effect each stage would have on the diffusional creep process would be individually analysed. Each stage is then written in such a way to allow further sections to be added over time. This results in six stages:

1. Diffusion in a pure metal
2. Diffusion in a material containing precipitates
3. Diffusion in a material containing precipitates subject to coarsening
4. Diffusion in a material containing voids
5. Diffusion in a material containing voids subject to growth
6. Diffusion in a material containing both precipitates and voids and subsequent evolution

The layout of the complete code is described in Figure 26 in the next section. This code is presented in the final stage representing all microstructural elements and microstructural evolution considered. The code can be altered retroactively to remove the effects of the unwanted scenarios.

Different grain geometry; precipitates and voids outside of the grain boundary and different types of precipitates are not considered at this stage of the project. Also, as the focus of this project is on material deformation due to vacancy movement, any deformation caused by any dislocation movement is beyond the scope of this project. However as diffusional creep and dislocation creep use different mechanisms they are mutually independent. This allows for the possibility to expand the model to include dislocation creep as a parallel deformation mechanism at a later stage.

5.2. Layout of the code

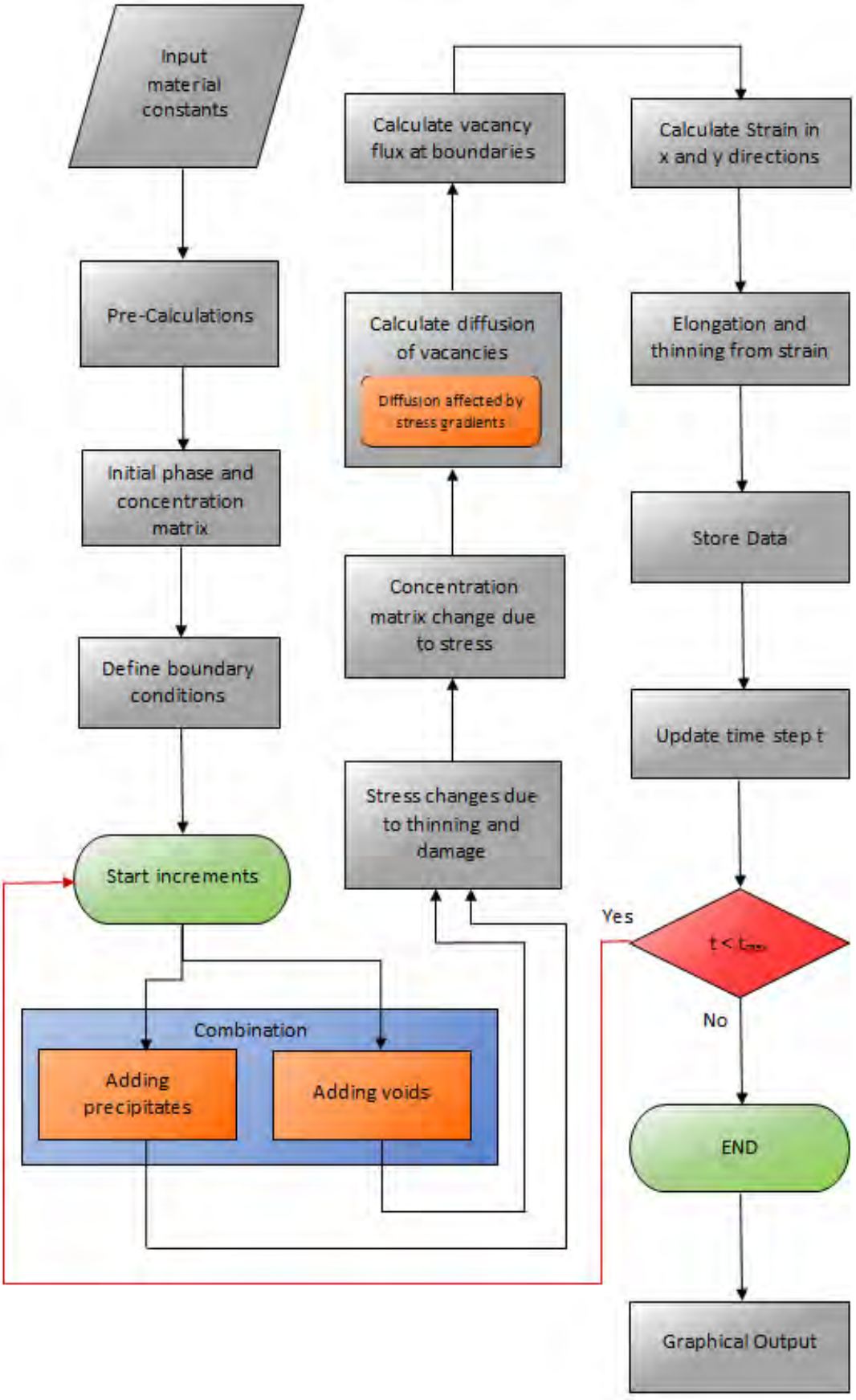


Figure 26 – Process followed by the model

This section describes the workings of the code in MATLAB, laid out in Figure 26. Initially, all necessary input data for the creation of the finite element grid has to be specified. These inputs would set the initial conditions of the material before any calculations or iterations are performed. This data relates to material constants, physical inputs and external inputs at every nodal point. The material constants describe the properties of the matrix and precipitates, such as Young's modulus, Poisson's ratio and formation enthalpy of vacancies. The physical constants relate to the chosen physical dimensions such as grain size, grain boundary width, and precipitate and void size and number. Finally, external inputs such as operating temperature and stress are needed.

These inputs are then used to calculate all necessary pre-calculations required before iterations can commence, or calculations that will not vary over time. Here the equilibrium concentration of vacancies in an unstressed state is calculated with equation 2.10 from section 2.2.3,

$$N_v = A_v exp(\frac{-E_v}{kT}) \tag{2.10}$$

The material correlation factors are also calculated first as they are dependent only on the material properties constants. These are given from equations 2.35 and 2.36 from section 2.4. They describe the interaction level between hard and soft materials,

$$k_y = \frac{E_2[(3-v_2)E_1 + (5-v_1)E_2]}{(E_1+2E_2)^2 - [v_2E_1 + (1-v_1)E_2]^2} \tag{2.35}$$

$$k_x = \frac{E_2[(3v_2-1)E_1 + (1-3v_1)E_2]}{(E_1+2E_2)^2 - [v_2E_1 + (1-v_1)E_2]^2} \tag{2.36}$$

The spatially resolved model of a heterogeneous microstructure is created using a number of matrices that overlay one another. Each matrix is responsible for a different property of the material. The first matrix is the phase matrix, which is created to represent the position of each microstructural element in the system. Each node represents a microstructural element which is considered to be homogeneous within its matrix element boundary. The microstructural elements considered within the phase boundary are:

- Matrix material of the grain
- Grain boundary
- Phase boundary
- Precipitates
- Voids

The phase boundary between the matrix and precipitates would usually have some intermediate properties, causing it to be a non-ideal source or sink. In this study the phase boundary is considered to have ideal grain boundary properties for the ideal situation. The code allows for this to be changed at any time.

Each material element would be represented as a numerical value in the phase matrix, as illustrated in Figure 27. This phase matrix would remain constant if there was no microstructural evolution included, but would have to be updated every iteration if microstructural evolution is included. The internal matrix material would be denoted at every node with a (0). The grain boundaries, and phase boundaries as they share the same properties, would be denoted as a (1). Voids and precipitates both

denoted with a (3). Initially, voids were separately denoted as a (2), but this caused errors when the void boundary came into contact with the precipitate at later stages of the code. An increase in mesh size would refine the microstructural shape. This would give a more accurate representation of the microstructure but would give a squared increase in the computational time.

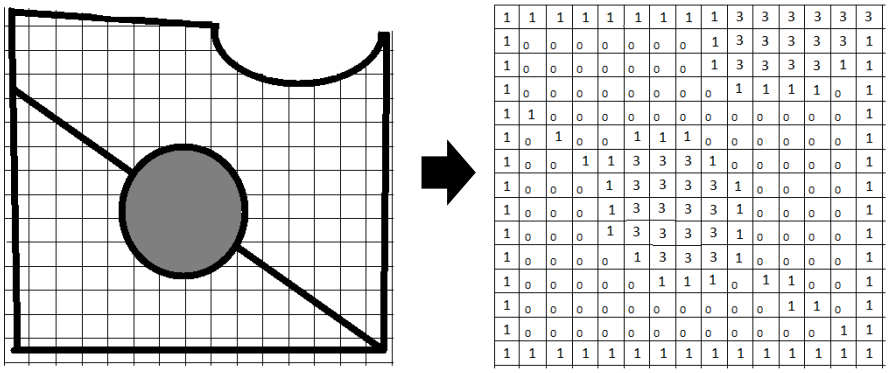


Figure 27 – Microstructure represented as a numberical phase matrix

The most important of the matrices, in terms of diffusional creep, is the concentration matrix. This matrix represents the concentration of vacancies at each nodal point of the material and allows the flux of vacancies to be tracked at every time step. The average vacancy concentration in the area is recorded at every nodal point. Initially, every element would be set as the equilibrium concentration of vacancies in the unstressed state, as calculated above. The concentration matrix then corresponds to the phase matrix where vacancy concentrations are reliant on the element. The grain and phase boundaries (1) become the boundary conditions of the FEM process. The concentration at these points is set and does not vary due to diffusion. At all matrix material elements, represented with a (0), the concentration changes only due to the diffusion of vacancies from neighbouring nodes. The concentration of vacancies within the precipitates (and voids) represented with (3) is set to zero as the vacancy concentration in these elements is not needed.

As is needed for any FEM process, before the iterative method can be employed, the initial conditions and boundary conditions need to be specified. The grain and phase boundaries become the boundary conditions for the calculations. The initial vacancy concentration is set to equilibrium across all elements as the stress is not yet applied. When a stress is applied the concentration at the nodes for grain boundaries (1) would be altered. The concentration at these nodes can only vary from an external input, such as the applied stress which changes due to damage evolution, and not from any internal diffusional functions. A blank phase matrix is set up. This gives a pure metal for the first stage of calculations and allows for alteration as further microstructural elements are added.

Once the inputs, boundary conditions and initial conditions have been specified the information can be entered into an iterative loop to calculate the creep strain. The calculations have been done in a stepwise fashion, using a “while” loop that will run for a specified number of time steps. This process allows multiple iterations of the complex system to be performed that would not be possible without a computer. Within every step of the iterations there are number of calculations performed. At the end of every step values are updated so that the next iteration can proceed.

From the inputs it is already specified whether the analysis is done on a pure material, a material containing precipitates, a material containing voids or a combination of both. If it is a pure metal then the step is skipped and moves onto stress concentrations due to stresses. If precipitates are added to the system a new selection of calculations is added in this stage of code, represented in Figure 28.

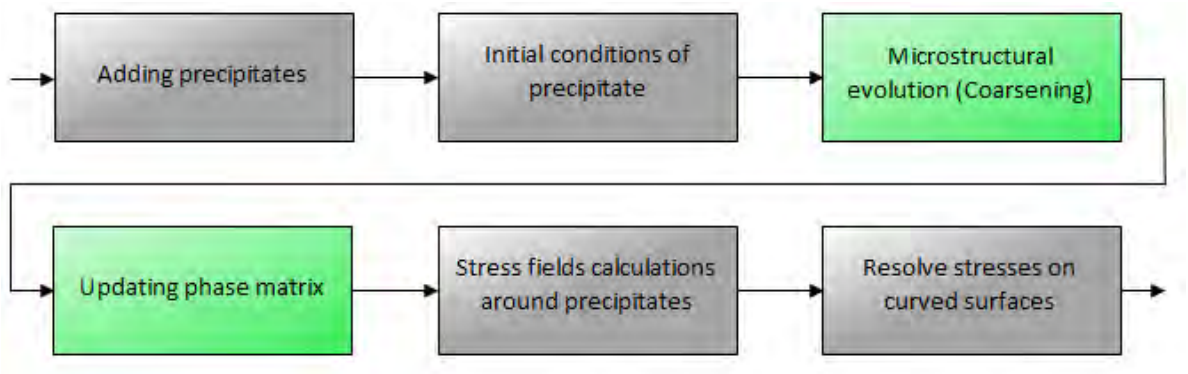


Figure 28 – Calculation process if precipitates are added to the system

The number of precipitates introduced in to the system is specified and then added in to the phase matrix. Initial conditions of the precipitates are included, giving position within the phase matrix and their initial size. If there is no microstructural evolution then the green steps are skipped. If there is microstructural evolution, such as in the later stages of the model, then its effect has to be calculated. For precipitates their evolution is coarsening. This uses equation 2.46, from section 2.5.4.6, to calculate the sizes of the growing precipitate:

$$r^{-3} - {r_0}^{-3} = K_c t \tag{2.46}$$

The size of the shrinking precipitate is calculated from removing the volume lost to the growing precipitates. A new phase matrix is then created to reflect any growing or shrinking precipitates and replaces the old phase matrix.

The stress fields are calculated using the size and position of the precipitates and combining them with equations 2.29 and 2.30 from section 2.4

$$\frac{\sigma_y}{\sigma} = 1 + \frac{(1 - k_y + k_x)R^2}{2{r_{xy}}^2}\left[1 + \frac{3R^2 + 10y^2}{{r_{xy}}^2} - \frac{8y^2(3a^2 + 2y^2)}{{r_{xy}}^4} + \frac{24R^2y^4}{{r_{xy}}^6}\right] - \frac{k_xR^2}{{r_{xy}}^2}\left(1 - \frac{2y^2}{{r_{xy}}^2}\right) \tag{2.29}$$

$$\frac{\sigma_x}{\sigma} = \frac{(1 - k_y + k_x)R^2}{2{r_{xy}}^2}\left[3 - \frac{3R^2 + 18y^2}{{r_{xy}}^2} - \frac{8y^2(3a^2 + 2y^2)}{{r_{xy}}^4} + \frac{24R^2y^4}{{r_{xy}}^6}\right] + \frac{k_xR^2}{{r_{xy}}^2}\left(1 - \frac{2y^2}{{r_{xy}}^2}\right) \tag{2.30}$$

These equations give the stress fields generated in both the X direction and Y direction when a stress is applied only along the Y direction. Shear stresses, from equation 2.31 in section 2.5.4.6 are excluded in this study as they do not have any effect on vacancy generation or diffusion. The stress fields form another matrix, the stress matrix, which overlays the phase and concentration matrices. The stress matrix would directly affect the vacancies in the concentration matrix.

Every time a precipitate is added their stress fields cause a synergistic effect that adds to the overall stress field. The stresses inside the precipitate are already given by equations 2.32 and 2.33 in section 2.4. As the precipitates are considered perfectly circular, the Y and X stresses have to be resolved to account for these curved surfaces. Only the stress normal to the surface is considered as this is the only force that can influence the concentration of vacancies. These factors all result in a normalised stress matrix that can be used to calculate vacancy concentrations later in the code.

If there are voids that are considered present in the model then another stage is added, shown in Figure 29, which has a similar layout to that of Figure 28.

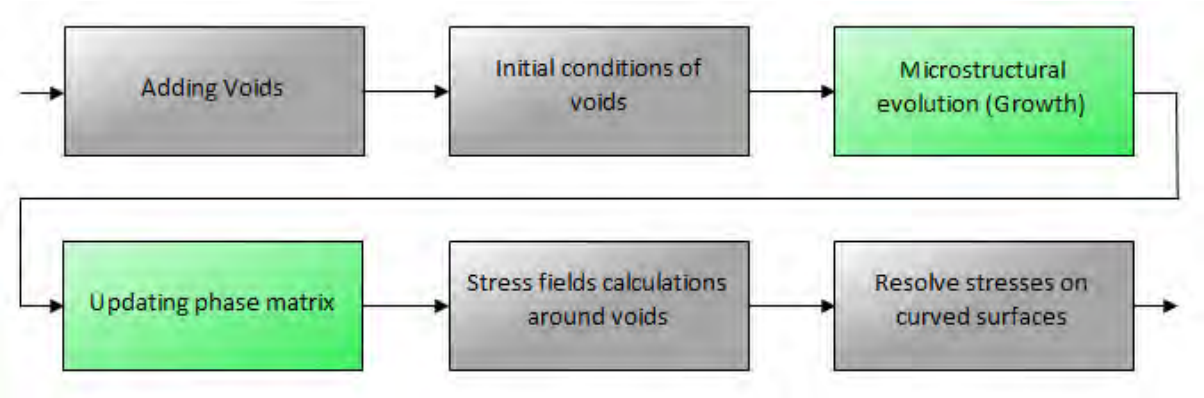


Figure 29– Calculation process if voids are added to the system

Nucleation theory of voids was not included in this study as it was complex and goes beyond the scope of the model. Rather, to see the effect of voids on a system, small pre-existing voids are added with their centre located at every corner of the grain within the phase matrix. The effect that a void would have on a grain is analysed separately by selecting large initial void size with no microstructural evolution or growth. Separately, if microstructural evolution was included, and the green steps were followed, then the voids would initially be very small. These small voids would then be subject to growth by diffusion according to equation 2.49 from section 2.5.4.8

$$\frac{da}{dt} \cong \frac{D_{gb}\delta\left(\sigma - \frac{2\gamma_m}{a}\right)\Omega}{2kT\lambda_s a} \tag{2.49}$$

The addition of voids at every corner results in the quartering of the system. This replicated the effect a single void would have on each whole grain. In the ideal situation, the meeting points of grain boundaries, such as the corner of cubic grains, is where voids are most likely to naturally form. This is due to the voids reducing the grain boundary potential. All void growth due to microstructural evolution would be reflected in an updated phase matrix.

The resulting stress fields from precipitates are also calculated using equations 2.29 and 2.30 but using different material properties. Changing material properties results in a much higher stress concentration with an inverse compared to that of hard precipitates. The stress fields are again resolved into normal forces along the curved surface of the voids.

During the last stage of the code, when both voids and precipitates are present within the system, both the processes from Figure 28 and Figure 29 would be used. These would interact to give the overall stress fields and stress concentrations throughout the material and would be presented as a stress matrix.

Before any vacancy concentrations due to stress can be calculated, the increase in stress due to thinning or damage must be included. At time = 0 there would be no thinning or void growth. However, as diffusion progresses and the grain elongates, its cross section would also thin. In terms of voids they would cause a loss in the load bearing area. These stresses manifest as a stress concentration and cause an increase in the applied stress and affect the normalized stress fields calculated previously. This damage is a representation of the loss in load bearing area compared to the original area. In terms of thinning and damage, both are calculated with equation 2.52 in section 2.5.4.8.

$$\sigma = \frac{\sigma_0}{1-\omega} \tag{2.52}$$

Here the damage parameters ω would be different for loss in area due to thinning compared to the loss in area due to void size. These two stress concentrations act parallel to one another and can be superimposed over one another.

Another form of stress concentration that is introduced in this study is the consideration of stress fields around the voids. Here the damage due to void growth and the stress concentration due to a 'soft' void in a harder matrix interact with each other. Here the stress increase due to a loss in area is accounted for in the stress field equations from section 2.4. Thus equation 2.52 only comes into play later for void damage, when the void is too large that the stress field equations from section 2.4 no longer include all stress concentrations within the grain.

Once all stresses are calculated the concentration matrix can be updated to reflect this. The changes in stress would affect the vacancy generation at the boundaries. Here the vacancy concentration changes according to equation 2.11 from section 2.2.4.

$$N_V^+ = N_V \exp(\frac{\sigma\Omega}{kT}) \tag{2.11}$$

By using a stress field that includes compressive stresses (negative stresses) then equation 2.12 is not needed. At all boundaries represented with a (1) in the phase matrix the stress is used to calculate the concentration changes at these nodal points. This updates the boundary conditions for the FEM process so that the diffusion can be calculated. The concentrations at all nodal points denoted with a (0) do not change as these concentrations are not affected by stress.

As the stress iteratively changes, the concentration is also updated to reflect this. The ability to include this incremental change in concentration at the boundaries is the reason Fick's second law was included in this model. The changes in concentration would not be reflected immediately but would have to first diffuse over time and Fick's first law alone cannot account for this. Any changes in stress would take some time to affect the concentration of vacancies through diffusion. The concentration boundary conditions would increase, which would then affect the rest of the diffusional process, something that the original Nabarro-Herring theory did not account for.

Once the initial and boundary conditions for a pure or a heterogeneous material have been updated the diffusion of vacancies can be calculated. This is solved by adapting Fick's second law, given in equation 2.9:

$$\frac{\delta c}{\delta t} = D\frac{\delta^2 c}{\delta x^2} \tag{2.9}$$

There are two ways to solve this partial differential equation, either explicitly or implicitly. This model makes use of the explicit discretized scheme to describe the change in concentration. This is chosen over the implicit scheme because it is much less computationally expensive. The drawback is that this type of code has a stability range for the time steps, which could lead to unstable code if the time step is not adhered to. Basically, this calculates the next incremental nodal concentration using the current concentration as well as the influence from diffusion at the surrounding nodes.

In an explicit discretized scheme equation 2.9 is represented as:

$$\frac{{c_{i,j}}^{n+1} - {c_{i,j}}^{n}}{\Delta t} = D\left(\frac{{c_{i,j+1}}^{n} - 2{c_{i,j}}^{n} + {c_{i,j-1}}^{n}}{\Delta x^2} + \frac{{c_{i+1,j}}^{n} - 2{c_{i,j}}^{n} + {c_{i-1,j}}^{n}}{\Delta y^2}\right)$$

This is especially important because of the large matrix size used in the diffusion calculations. The drawback of the explicit method is that this code can become unstable if conditions are outside of a certain range. An implicit method is always stable but requires much more computational power.

In the scenarios including a heterogeneous microstructure stress gradients are present within the body of the grain. This was the basis behind the derivation in section 4. Thus equation 4.15 describes Fick's second law in a stress gradient.

$$\frac{\partial \mathrm{c}}{\partial \mathrm{t}} = D_c \left(\frac{\partial^2 \mathrm{c}}{\partial \mathrm{s}^2} + \frac{N_{avo}}{E_{M,mol}} . \Omega_{at} . \frac{\partial^2 \sigma}{\partial \mathrm{s}^2} \right) \tag{4.15}$$

Applying an explicit scheme to this equation gives the change in concentration over time as:

$$\frac{{c_{i,j}}^{n+1} - {c_{i,j}}^{n}}{\Delta t} = D \left(\frac{{c_{i,j+1}}^{n} - 2{c_{i,j}}^{n} + {c_{i,j-1}}^{n}}{\Delta x^2} + \frac{N_{avo}}{E_{M,mol}} . \Omega_{at} \frac{\sigma {x_{i,j+1}}^{n} - 2\sigma {x_{i,j}}^{n} + \sigma {x_{i,j-1}}^{n}}{\Delta x^2} + \frac{{c_{i+1,j}}^{n} - 2{c_{i,j}}^{n} + {c_{i-1,j}}^{n}}{\Delta y^2} + \frac{N_{avo}}{E_{M,mol}} . \Omega_{at} \frac{\sigma {y_{i+1,j}}^{n} - 2\sigma {y_{i,j}}^{n} + \sigma {y_{i-1,j}}^{n}}{\Delta y^2} \right)$$

The two equations shown above for explicit schemes only calculate the change in concentration on matrix material nodes (0) in the phase matrix. In this way the boundary conditions specified at grain/phase boundaries (1) in the phase matrix influence the diffusion of vacancies but are not changed by the diffusion process.

Once both the vacancy has diffused for a single time step and the concentration matrix has been updated, then the vacancy flux can be calculated. The flux of vacancies away from the tensile boundaries and the flux of vacancies to the free boundaries are calculated. This is done using Fick's first law only on the grain boundary nodes. This would give the exact number of vacancies moving away/to the boundary layer, indicating the amount of material moving at these boundaries. Every vacancy leaving a face is replaced by an atom, effectively adding another atom to the surface and causing a miniscule amount of elongation. Conversely any vacancy that arrives at a side boundary removes an atom and causes miniscule thinning. By calculating the number of vacancies leaving the surface at every time step, we can get the strain rate over the entire process.

The flux of atoms is given using Fick's first law. This number is given as the molar concentration, which is multiplied by Avogadro's number to give the actual number of vacancies diffusing. Following the method set out in section 2.3.1 every vacancy that leaves a surface is replaced by an atom. Every arriving atom adds one atomic volume, Ω, to the surface layer. The number of diffusion vacancies and volume per atom gives the total volume added per time step. This volume spread over the unit area, and accounting for matrix size, would give the total elongation of the grain. Dividing by the original length gives the strain along the tensile axis. Conversely, the thinning strain along the X axis is calculated in the same way but with atoms being removed due to the arrival of vacancies.

The resulting strains in the X and Y directions give the new, deformed, dimensions of the cubic grain. These dimensions, and consequently the X and Y step sizes, are updated as they affect the diffusion rates, stress concentrations due to thinning, and strain rates in the next iteration.

The resulting strains, vacancy concentration fields and stress fields are all stored for the next iteration. The time step updates and if the overall specified time limit is under the desired number of increments the process will repeat. If the time limit is met and all necessary calculations have been performed, the process stops and the strain rate is displayed as a graphical output. Any additional information needed, such as the concentration profiles or stress profiles, can be called up and subjected to any necessary post processing.

6. Creep Simulations

The main physical parameters used in this study are given below in Table 4. Apart from the commonly known parameters, these are taken from literature and are typical parameters from type P91 steels.

Table 4 – Table of all important parameters used throughout the model procedure

Temperature (°C)	Stress MPa	Matrix size	Diffusion Coefficient (m^2s^{-1})	Grain Size (m)	Atomic Volume (m^3)	Young's Modulus Matrix (GPa)
600	30	100 x 100	2.85×10^{-19}	1×10^{-6}	1.18×10^{-29}	200
Young's Modulus Precipitate (GPa)	Precipitates Volume Fraction	Poisons ratio	Coarsening Coefficients	Effective Void Young's Modulus	Grain Boundary Diffusion Coefficient (m^2s^{-1})	Grain Boundary Thickness (m)
430	10 %	0.3	5.8×10^{-28}	0	9.20×10^{-12}	0.5×10^{-9}

For the first stage of code for a pure material a number of constants are needed. The temperature is set at 600 °C, a temperature high enough for lattice diffusional creep to occur. Another reason for choosing this temperature is that all the available constants were known at this temperature. An initial stress of 30 MPa was chosen as it is low enough to be in the diffusional creep range. The matrix size was chosen as 100 x 100. This provides enough accuracy without increasing the computational time significantly. The matrix size can be increased to increase accuracy or decreased to accelerate computational speed. The diffusion coefficient of 2.85×10^{-19} $m^2s^{-1}$is taken from [17]. An acceptable grain size of 1 micron is set for all calculations. The atomic volume was calculated using the molecular weight and density of iron and found to be 1.18×10^{-29} m^3.

When moving onto including precipitates, new constants would need to be introduced. The Young's modulus for the matrix is set as 200 GPa for iron, however this could be lowered to reflect the softening at the elevated temperature. For example, it could be lowered to 143 GPa, which is the effective Young's modulus at the operational temperature of 600 °C. The Young's Modulus of the carbide precipitate is set at 430 GPa. Volume fractions occupied by the precipitates was rounded up from 9% in [51] to 10%. Poisons ratio for all materials was set at the standard value of 0.3. The coarsening coefficient of $M_{23}C_6$ precipitates is calculated using the data in [52] and found to be 5.8×10^{-28}.

With the inclusion of voids in the model further additional constants are introduced. In order to keep coherency with equations 2.35 and 2.36 the void is considered to be a precipitate with a Young's modulus of 0 GPa. The grain boundary diffusion coefficient can be calculated from the lattice diffusion coefficient taken from [17]. The grain boundary diffusion coefficient has half the activation energy than that of the lattice diffusion coefficient, where the diffusion coefficient equation is an exponential. Therefore, dividing by half the exponential would give the grain boundary diffusion coefficient:

$$D_{gb} = \frac{D}{e^{0.5*\left(\frac{-Q}{RT}\right)}}$$

Where Q, the activation energy, is 251 kJ/mol K, and, finally the grain boundary thickness of 0.5 nanometres is taken from [79].

Table 5 below gives which parameters are used for each case of the model.

Table 5 – The spread of the use of parameters in the multiple model stages.

Parameter	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Temperature	X	X	X	X	X	X
Stress	X	X	X	X	X	X
Matrix size	X	X	X	X	X	X
Diffusion Coefficient	X	X	X	X	X	X
Grain Size	X	X	X	X	X	X
Atomic Volume	X	X	X	X	X	X
Young's Modulus Matrix		X	X	X	X	X
Young's Modulus Precipitate		X	X			X
Precipitate Volume Fraction		X	X			X
Poisson's Ratio		X	X	X	X	X
Coarsening Coefficient			X			X
Effects Young's Modulus Void				X	X	X
Grain Boundary Diffusion Coefficient				X	X	X
Grain Boundary Thickness				X	X	X

6.1. Case 1: Uniform Grain with no Stress Gradient

6.1.1. Description

The code for this section is found attached as **A_pure_metal.m**. This code follows Figure 26 and includes all the grey blocks.

The initial step of the code calculates the strain rate of a pure metal under similar circumstances to that of the original Nabarro-Herring model. This means the code is performed on a pure material that contains no other microstructural elements such as voids and precipitates and no stress gradients. This is done to ensure that when changing from Fick's first law to Fick's second law that the solution is still of the same order and still comparable to the Nabarro-Herring model. The initial stage of the model also sets up the necessary parameters that will allow the model to account for any change in concentration as the stress conditions change. In this case the change in stress is due to the thinning of the grain from diffusion. This stress change would be uniform across the entire specimen due to the homogeneity of the grain and thus no stress gradients are considered at this stage.

Further advantages of using Fick's second law are that the diffusional process can be analysed from the moment the stress is applied and that grain thinning can be included. The Nabarro-Herring model considers that a steady state concentration gradient has already formed between the grain boundaries. However, diffusion in metals is a very slow process and it would take some time before a steady state gradient is reached. Here vacancies are formed at the grain boundaries experiencing a tensile stress while the body of the grain has not yet experienced any change in the concentration of vacancies. Thus there would be an initial rapid flow of vacancies into the body of the grain due to the steep difference in concentration between the boundary and interior of the grain. As the body of the grain becomes saturated with vacancies, the gradient would decrease and the rate of vacancy flow would decrease. Thus the primary stage of creep, caused specifically by the flow of vacancies, would be represented because of this initial high generation rate of vacancies. The steady state gradient then forms when the rate of vacancies entering the body of the grain at the tensile grain boundaries balances out with the vacancies reaching, and being absorbed at, the free grain boundaries. Once the generation and annihilation of vacancies balances out the model would then lead into the secondary stage of creep.

Finally, Fick's second law is used to show the increased vacancy generation that would occur due to the thinning of the grain and subsequent stress increase. This continuous increase in vacancy generation would lead the curve into the tertiary creep stage, a feature not represented by the Nabarro-Herring model. This tertiary curve is expected to be more gradual than a regular ideal creep curve due to the lack of vacancies generated by microstructural and damage evolution.

The model is set up with no second phase particles in a cubic grain, with a uniaxial force applied along the Y axis. The boundary conditions are then set where the top and bottom boundaries, represented in red in Figure 30 (a), have the stress normal acting on them and will have an increased vacancy concentration. The side boundaries, represented in blue in figure 30 (b), have no normal stress acting on them and thus have no change in the equilibrium concentration of vacancies. Any meeting points between two grain boundaries of different orientation, circled in Figure 30 (c), are taken as an average in concentration between the two to keep coherency of the system.

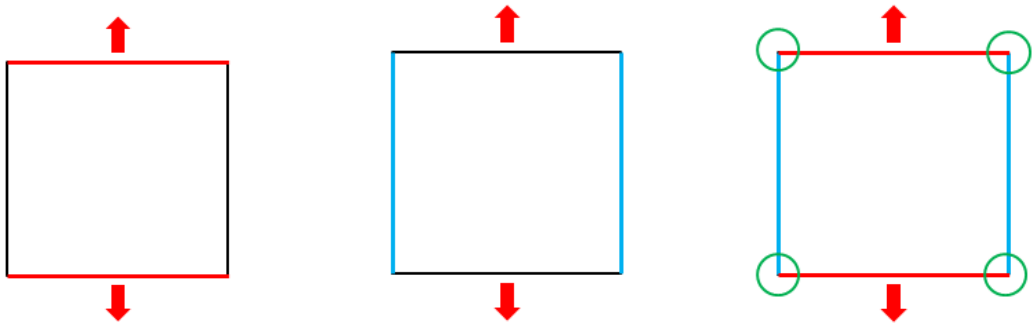
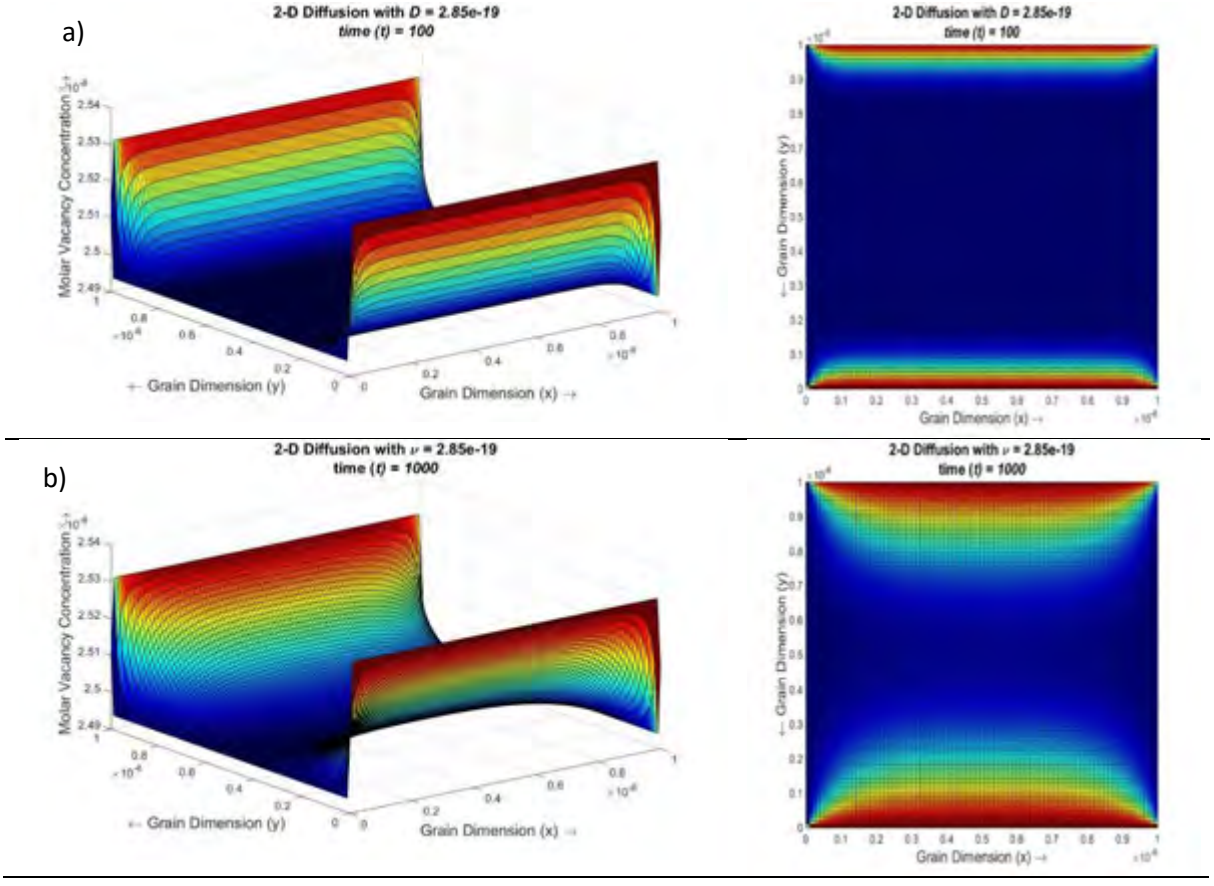


Figure 30 – Boundary conditions on a square grain

6.1.2. Results



c)

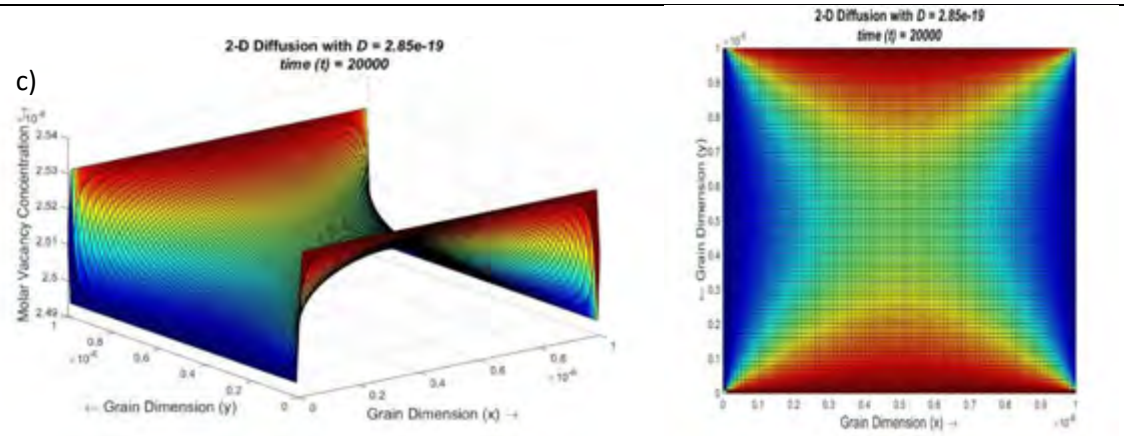


Figure 31 – Vacancy conentration fields during transient stage of creep

The resulting vacancy concentration fields are shown above in Figure 31, where the left hand column is a 3D view of the field while the right hand side is a view on the XY axis. Figure 31 (a) shows the vacancy concentration field very soon after the diffusion process has started. The red region represents the high concentration of vacancies present in the top and bottom boundaries. The dark blue represents the lowest concentration of vacancies, which in this case is the equilibrium concentration of vacancies in an unstressed state. The short width between the red and blue regions shows that there is a large concentration gradient and subsequent rapid flow of vacancies from the top and bottom faces into the body of the grain. The internal concentration of vacancies has not yet been increased and is still in equilibrium as not enough time has passed for diffusion to progress this far. The bowing at the corners is due to a small amount of vacancies already getting absorbed at the side boundaries.

Figure 31 (b) highlights the changes in the grain after a short amount of time has passed. The vacancies have progressed further into the body of the grain, represented by the change in colour away from blue. As the vacancy concentration inside the body of the grain is raised it causes a decrease in the boundary surface gradient. This results in a decreased vacancy generation rate and subsequent flux at the top and bottom faces. Further bowing is seen on the right hand side of Figure 31 (b). This is due to vacancies also moving in the X direction until they are annihilated at the free boundaries, maintaining a low vacancy concentration in these regions.

Vacancy generation and flow in the Y direction starts at a maximum and decreases as vacancies flow into the body of the grain. This flow into the grain causes the flow of vacancies out of the grain in the X direction to increase. The flow of vacancies into the body of the grain would decrease and the flow of vacancies out of the grain would increase until these two values converge. At this point the rate of vacancy generation and annihilation reaches a steady state. Figure 31 (c) shows this stage when the vacancy concentration field has reached its dynamic equilibrium. This equilibrium pattern is seen with the symmetrical spread of the colour gradient and is where the diffusional process moves from the primary stage of creep into the secondary, steady-state creep.

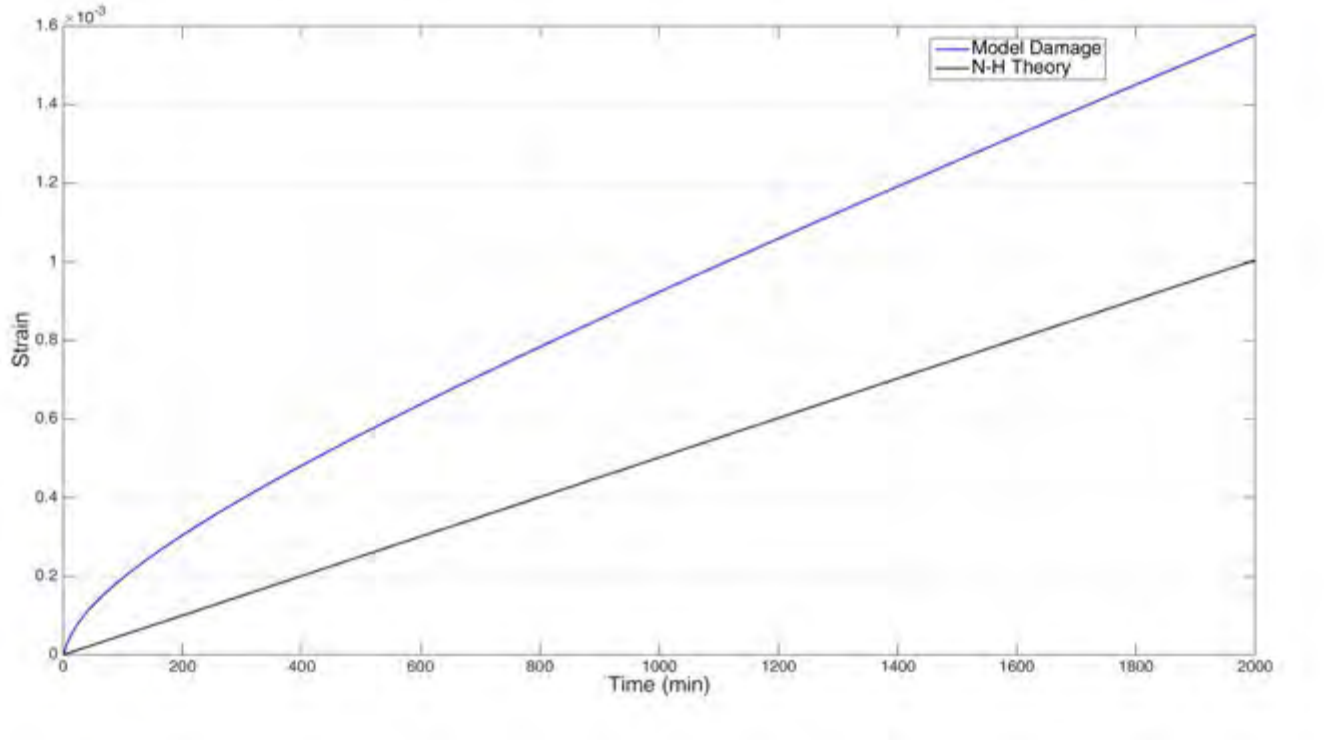


Figure 32 – Strain graph produced from model showing primary stage of creep

Figure 32 represents the strain rate from the aforementioned process. The blue line represents this diffusional creep model. The initial rapid generation of vacancies gives the initial high strain rate. As the vacancies flow into the body of the grain the generation of vacancies decreases, the creep rate then slows until it reaches the steady state gradient. The creep deformation from the Nabarro-Herring model is represented as the straight black line, which will be used as the reference in all cases. The steady state region of this model closely resembles that of the linear creep rate given by the Nabarro-Herring model with a very similar gradient. This model has a slightly higher creep rate than that of the Nabarro-Herring model. This slight difference is likely due to the slight difference caused by changing from the one dimensional concentration gradient to the concentration profile in two dimensions seen in Figure 31. This effect would be further influenced by the size of the matrix chosen. While a larger matrix would offer greater accuracy, it would increase the computational time by a squared factor.

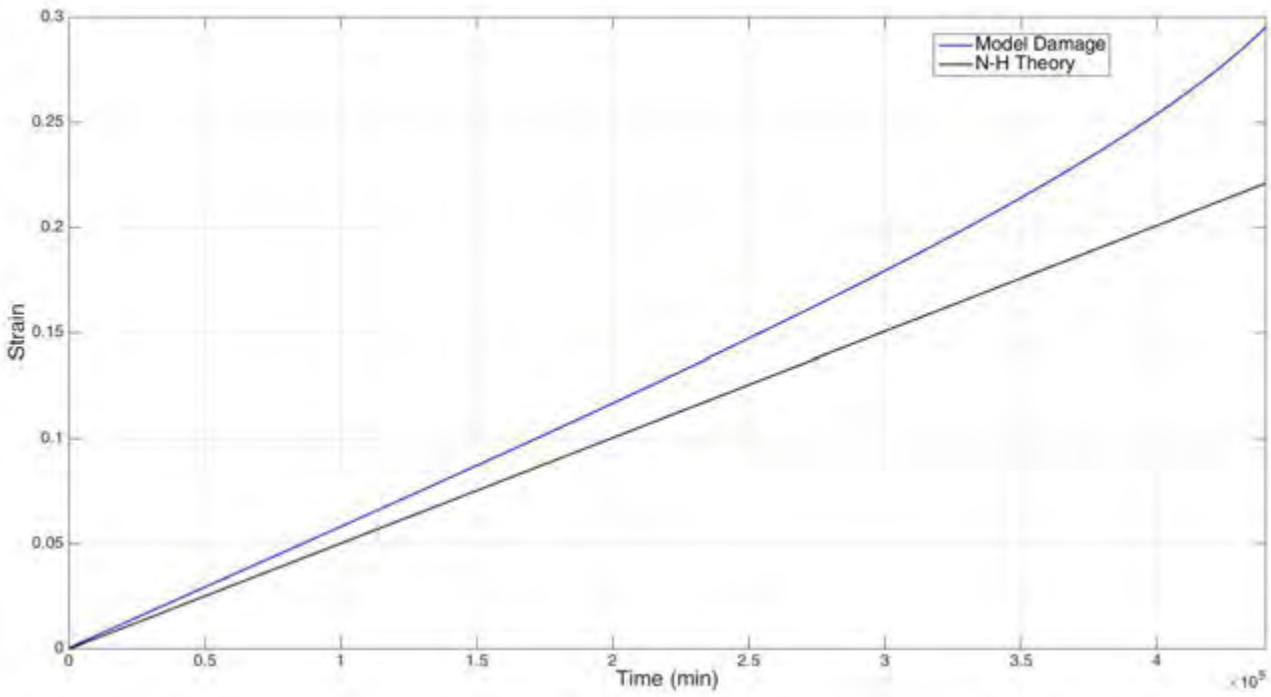


Figure 33 – Strain graph showing the increase in strain rate due to thinning of the grain

Using Fick's second law also allows for the addition of the effect the thinning of the grain would have on the stress and, therefore, change the strain rate over time. Although the reason for the tertiary stage of creep is void nucleation and damage evolution, if voids were neglected in first approximations then the thinning of the grain would cause a stress increase. This increase speeds up the diffusional process and would lead to the tertiary stage of creep and eventually failure. Thus this model is able to incorporate a tertiary stage of creep, which was not possible in the Nabarro-Herring model due to its linear nature.

As seen in Figure 33, the creep process is calculated over an extended amount of time where it is able to show the material moving into the tertiary stage of creep due to thinning. The primary stage of creep is found to be markedly smaller than anticipated. The generation and annihilation only truly balance after roughly 15 000 minutes when reaching the true dynamic equilibrium; however, as seen by the creep curve, this initial production of vacancies rapidly slows down with an exponential decay.

6.1.3. Discussion

Altering the Nabarro-Herring equation to include concentrations changes over time did allow for a primary diffusional creep stage to be expressed. However, this stage is much smaller than previously anticipated and would be inconsequential over expended creep curves. This would mean that, even at very low stresses, the primary creep stage is affected by many other factors.

The time step that was added to speed up the code was removed to see how this affected the rapidly changing transient stage of the vacancy concentration field. It was theorised that averaging the flux over 60 seconds would result in a loss in creep as the flux was rapidly decreasing. Figure 34 shows that while there is less than a 10% loss in predicted creep deformation, this is still not significant compared to the rest of the creep process.

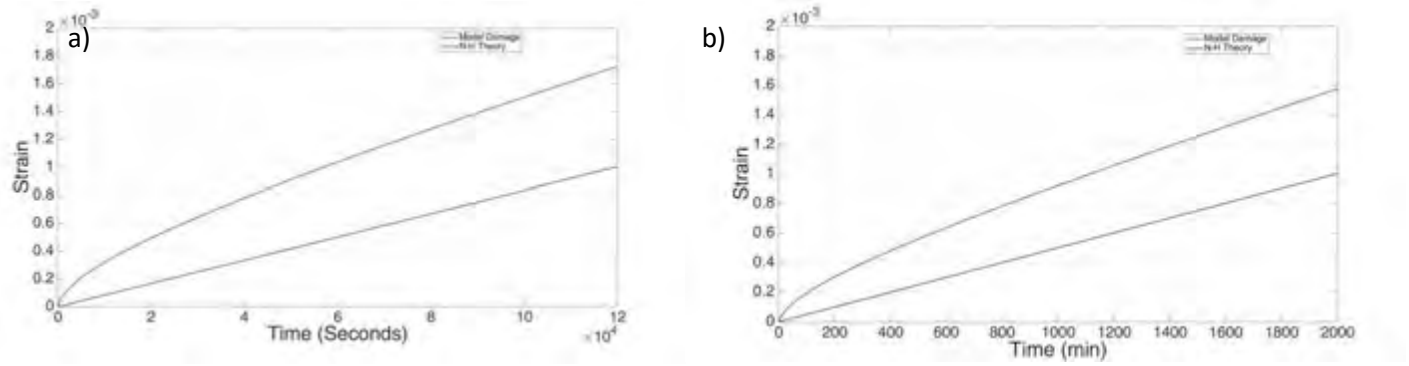


Figure 34 - Primary creep at (a) time step = 1 and (b) time step = 60

6.2. Case 2: Heterogeneous Grain Containing Precipitates

6.2.1. Description

The code for this section for precipitates is found attached as **E_Creep_Precipitates.m**. A separate code calculating the stress fields is attached as **B_Stress_Fields.m**. This code follows Figure 26 and includes all grey blocks, orange steps and the calculation process shown in Figure 28.

The next stage of the model was to calculate the influence that precipitates have on the strain rate when including precipitates into the material. Precipitates are added to simulate a similar microstructure to that of high chromium steels, although precipitates are not specific to high chromium steels. In high chromium steels the precipitates are found both at the grain boundary and inside the grain. However, it is theorised that the precipitates found at the grain boundaries would have the greatest effect on diffusional creep as this changes the effectiveness of the boundary as a source of vacancies.

The stress fields that form around hard particles would influence the stress in the grain boundaries, affecting the vacancy generation. These stress fields created by the precipitate would depend on the size of the precipitates as well as the difference in Young's modulus between the matrix and precipitate. Thus the largest, hardest precipitate, $M_{23}C_6$ is chosen as it is theorised to have the greatest influence on diffusional creep. These precipitates are also chosen as they are mainly found on the grain boundary as discussed in the literature review.

For the scope of this project only three scenarios of precipitates are modelled, where the three scenarios are shown below in Figure 35. The initial scenario is where one precipitate is added at each of the tensile faces. As a further precipitate is added, the size is reduced to keep the volume fraction constant. This is again kept constant when adding a third precipitate at the boundary for the last scenario. By only modelling a single precipitate allows for investigating the direct influence a precipitate would have on the creep rate; adding the further two precipitates would allow for investigating the effect a dispersion of precipitates would have. The addition of more precipitates also allows for adding, and analysing, the effect of coarsening in the later stage of the project.

The interface boundary between the matrix and precipitate is considered the same as the grain boundary in terms of being an ideal source of vacancies, even though in a realistic situation, the interface is a non-ideal source (or sink) of vacancies due to the energy of the structure. Thus for the three scenarios the initial grain boundary and the interface around the precipitates become the active sources for vacancies. This is illustrated as the red line in Figure 35, where the concentration at these points would only be influenced by the applied stress.

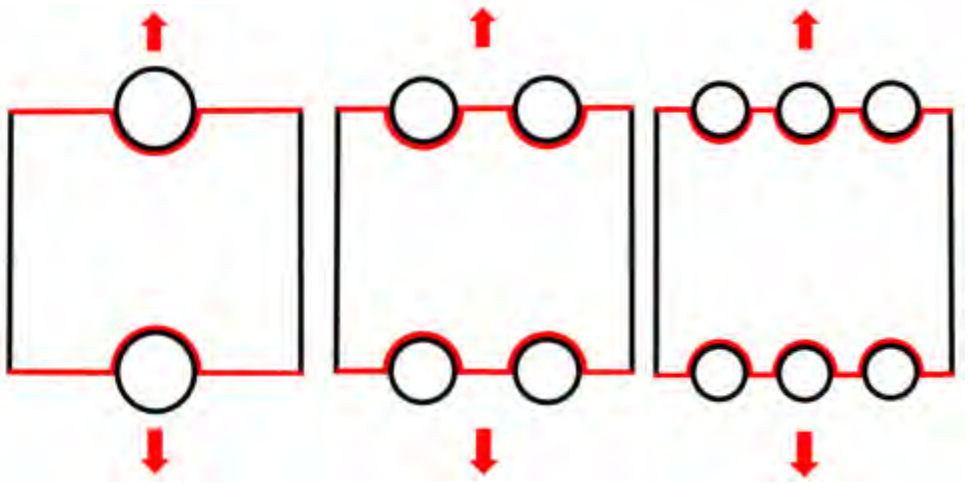
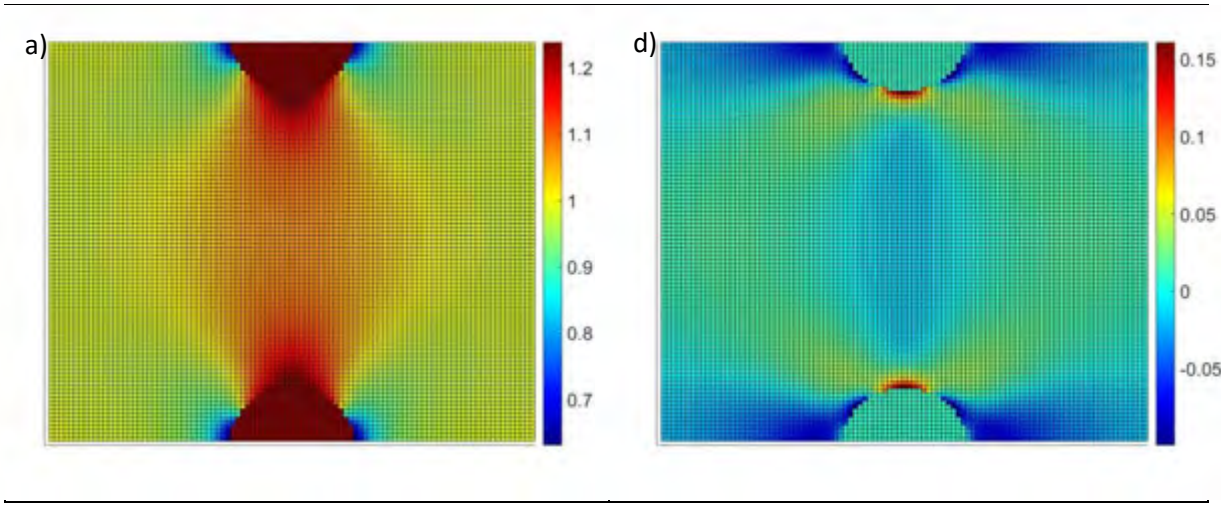


Figure 35 – Active sources of vacancies

6.2.2. Results

The stress field generated from each scenario containing one, two and three precipitate(s) is given below in Figure 36. The stress fields in the Y direction are given on the left and the stresses in the X direction are given on the right. The normalized scale of the stress field is shown to the right of each image. As more precipitates are added the stress distribution becomes more complicated. In the Y direction stresses are highest within the precipitate that decreases over the length of the grain between the two faces. The Y stresses have the greatest effect on vacancy creation, as expected, while the X stresses contribute very little towards vacancy creation. In the X direction, there is compression at the sides of the precipitate with a small amount of tension at the top and bottom of the precipitates. Once stresses have been resolved the X stresses have very little influence on the grain boundaries. In this model shear stresses have been neglected as once they are resolved normal to a boundary they have a nominal effect on vacancy generation. However, they can be added at a later stage by referring back to the paper on field stresses [45].



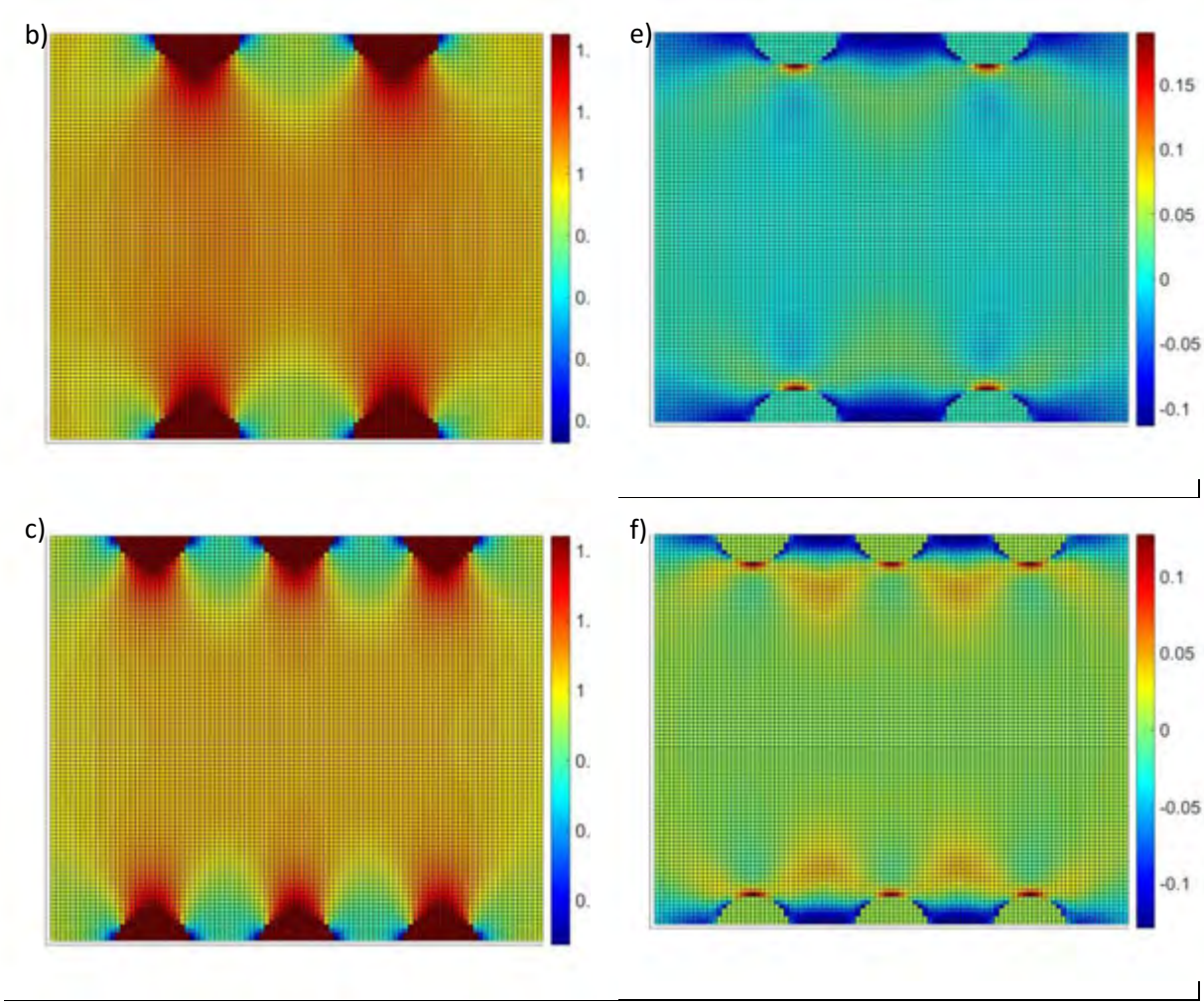


Figure 36 – Stress fields distribution around precipitates. Y stresses caused by (a) 1 precipitate (b) 2 precipitates (c) 3 precipitates and X stresses caused by (d) 1 precipitate (e) 2 precipitates (f) 3 precipitates

A closer look at the stresses directly affecting at the grain boundaries is given below in Figures 37 and 38.

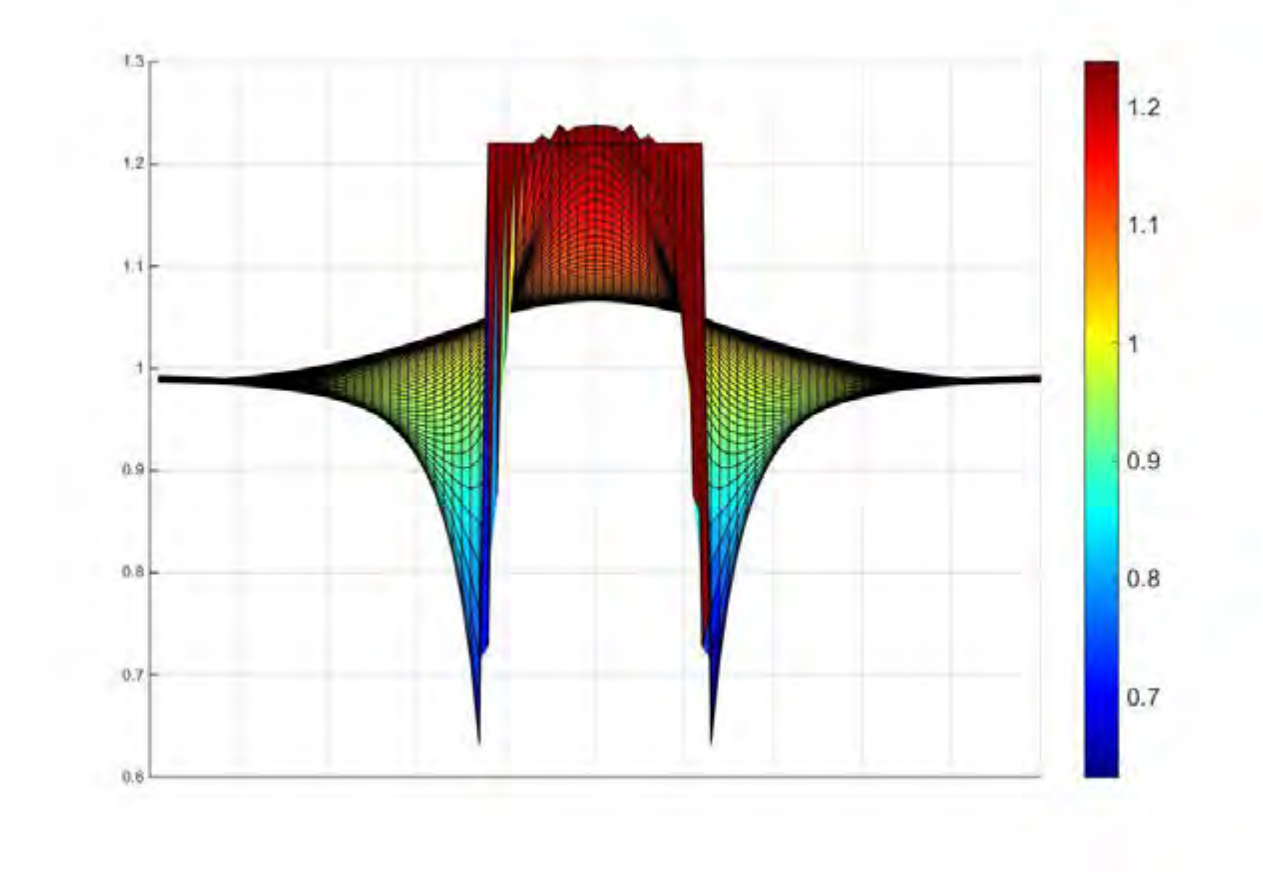


Figure 37 – Y stresses at grain boundary for 1 precipitate

Figure 37 shows a side on view of the Y stresses at the grain boundary when there is a single precipitate at the boundary. All stresses are normalized where the change in stress is divided by the applied stress. As stated before, the stresses inside the precipitate are high, with a peak at the centre of the precipitate. This stress decreases along the face of the precipitate and eventually causes a reduction in stress at the sides of the precipitate. The extent of the reduction would be affected by the hardness of the precipitate in comparison to the matrix. The higher stresses on the precipitate would be resolved into only normal forces, causing fewer vacancies to be generated at these curved surfaces. The lower stresses at the sides of the precipitate would cause a lower concentration of vacancies to form at these grain boundaries. The overall effect shows that the addition of precipitates would reduce the amount of vacancies generated. This would, of course, decrease the rate of creep deformation as there is a lower number of the diffusing species.

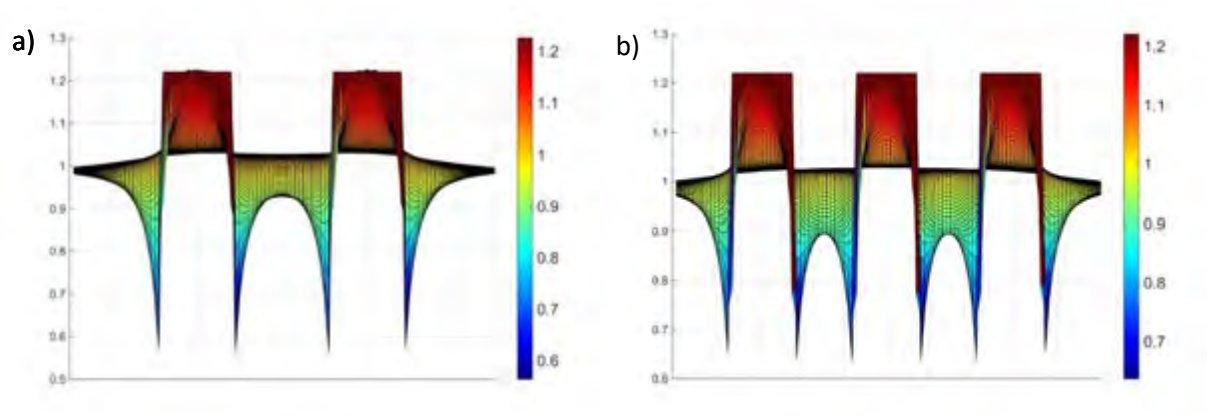


Figure 38 – Y stresses at the grain boundary for (a) 2 precipitates and (b) three precipitates

Figure 38 shows the further two scenarios where two and three precipitates are present at the boundary, respectively. Here the addition of precipitates shows that the stress fields between the precipitates start to interact. This causes a further lowering of the stress field next to the precipitates and a resulting decrease in the generation of vacancies in these regions. This compounds as more precipitates are added. A small decrease is even seen on the curved interface between the precipitate and matrix, where the peak stress has been slightly lowered.

The resulting creep curves given when precipitates are added to the boundaries are found below. These three scenarios show the respective increase in the number of precipitates at each boundary and produce the strain graphs shown in Figures 39, 40 and 41.

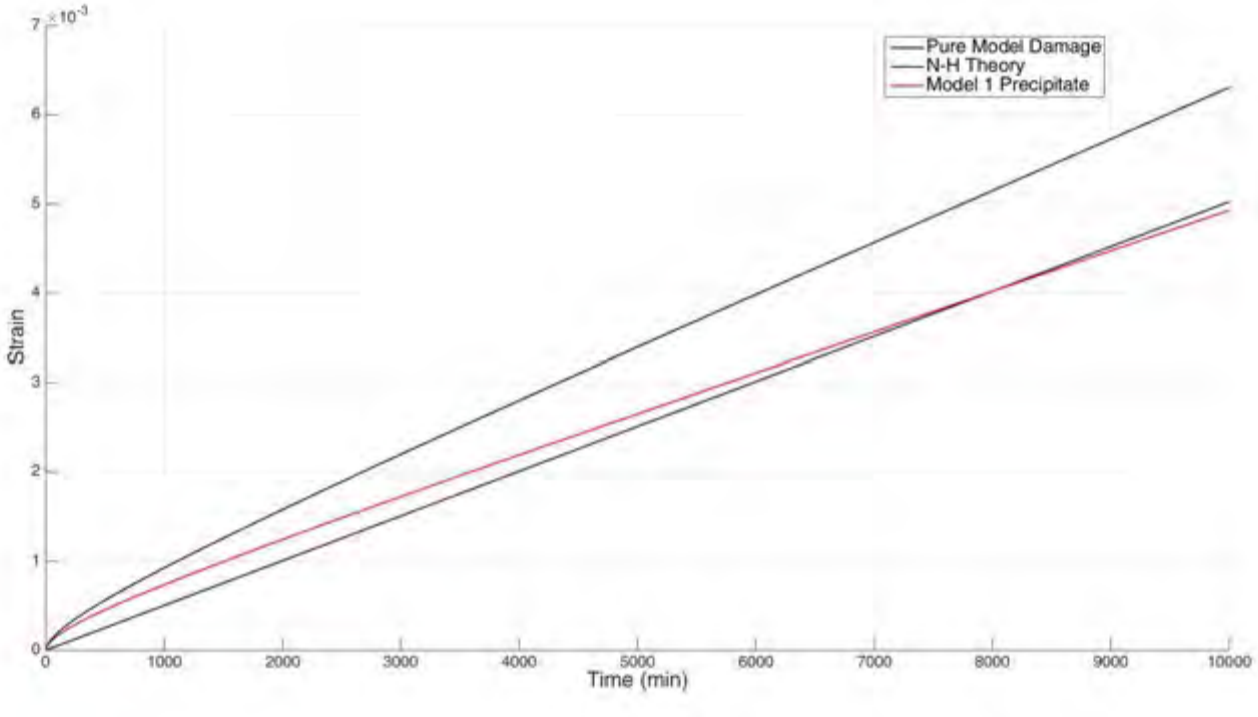


Figure 39 – Strain graph showing decrease in strain rate due to one precipitate

Figure 39 shows the creep rate when a single precipitate is added to the grain boundary. The bottom most linear black line is from the Nabarro-Herring theory while the topmost black line is the creep rate from the model on a pure metal, shown in the previous section. The red line represents the decrease in the creep rate due to the addition of a single, large precipitate. This graph shows that within the terms of the model, there is a significant decrease in the creep rate when precipitates are present in a metal. This finding fits in with experimental observations and offers credibility to theory that stress fields around precipitates causes the decrease in the diffusional creep rate.

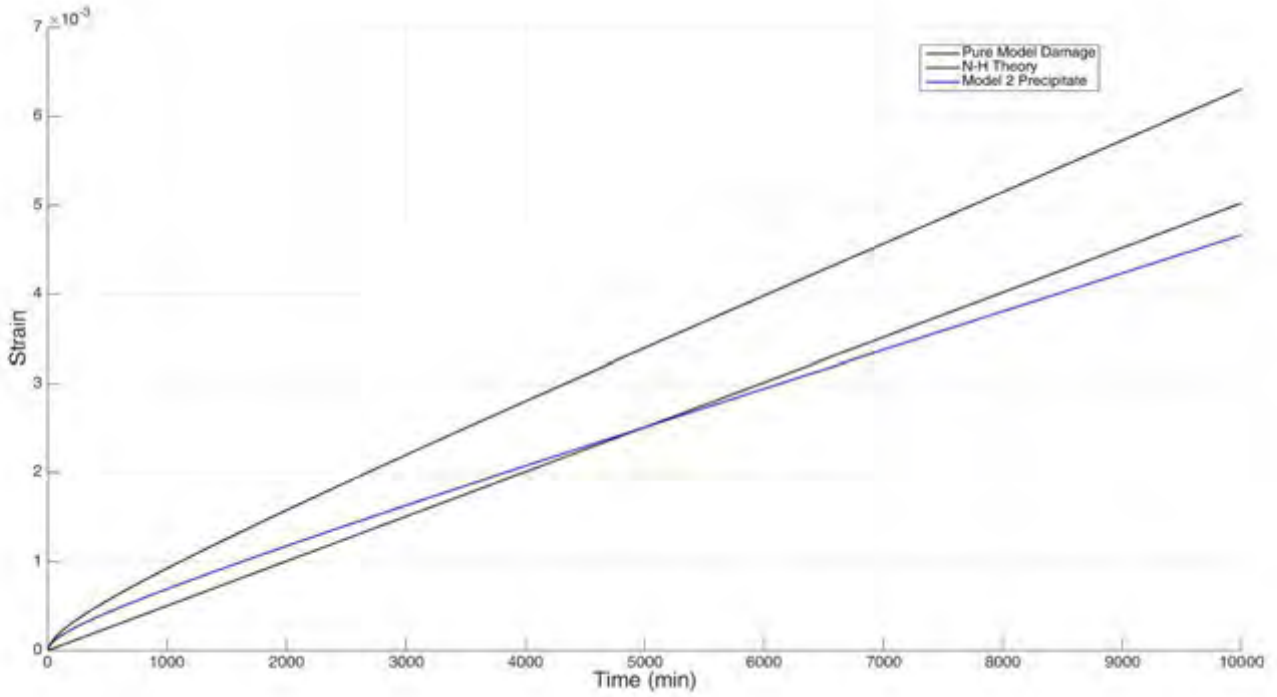


Figure 40 - Strain graph showing decrease in strain rate due to two precipitates

Figure 40 represents the scenario where there are two precipitates at the grain boundary. The black lines are exactly the same as that in Figure 39. The blue line represents the creep curve for two precipitates. As can be seen, the line is lower than that of the red line in Figure 39. Thus the creep rate has decreased even though the volume faction of the precipitates has remained the same.

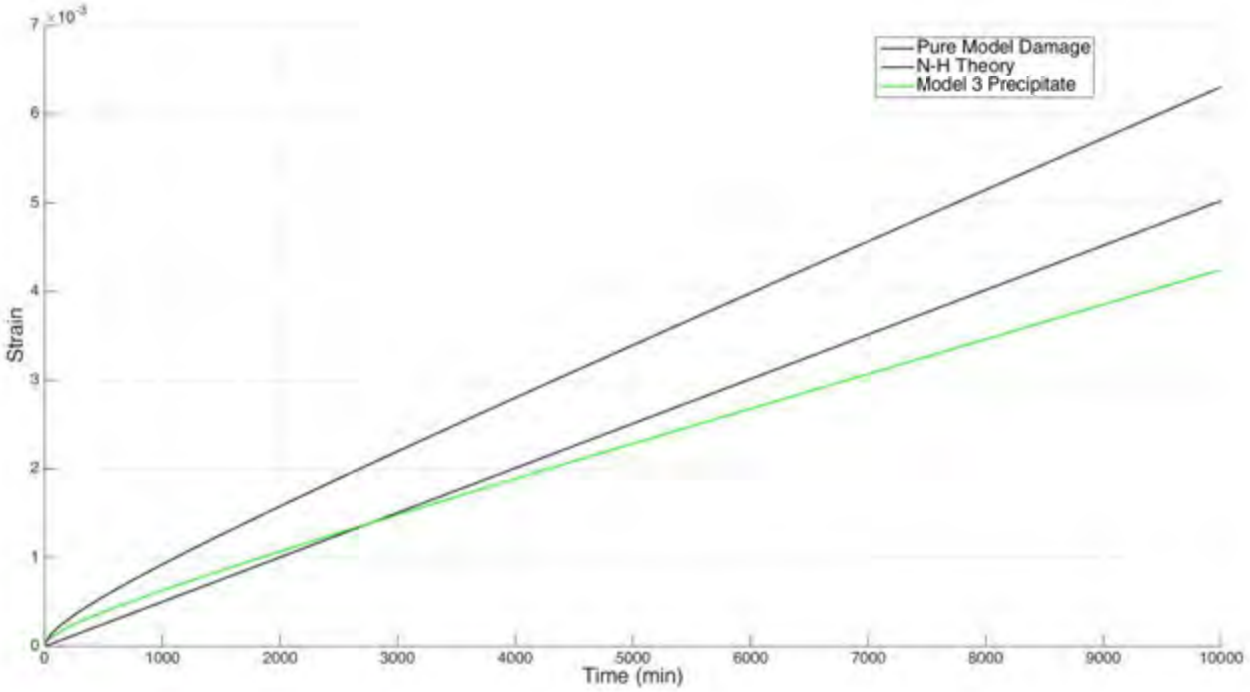


Figure 41 – Strain graph showing decrease in strain rate due to three precipitates

The green line in Figure 41 shows the steady state creep caused by having three precipitates at the boundary, with the creep rate further decreased by adding another precipitate even though the volume fraction is still kept constant.

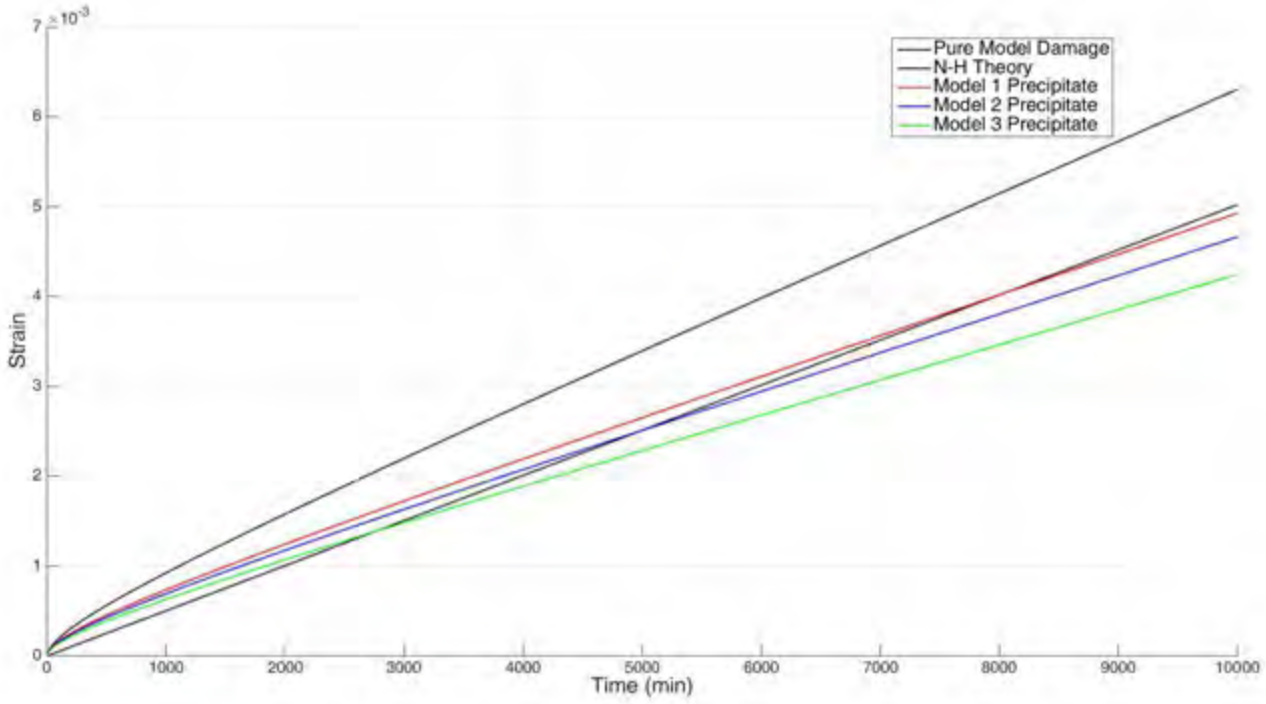


Figure 42 – Comparison of creep rate reduction due to precipitates

Figure 42 shows all three creep curves added on the same graph. The red, blue and green lines represent one, two and three precipitates at the grain boundary respectively. While the black lines are the Nabaro-Herring equation and model for a pure metal. This graph gives a clearer representation of how the creep rate decreases with the addition of precipitates.

This observation supports the proposal that the decrease in diffusional creep rate in the presence of precipitates is in fact due to the stress fields that form around these precipitates.

6.2.3. Discussion

The presence of precipitates in creep resistant steel does, in fact, affect diffusional creep by means of the stress fields that form around the precipitate once a force is applied. This stress field causes fewer vacancies to be generated by means of lowering the stress at the grain boundaries. As a result the diffusing species is decreased and the rate of diffusion is reduced.

Figure 42 shows what is already known through experimentation and observation. An increase in the number and distribution of precipitates, even when keeping the volume fraction of precipitates constant, results in a decrease in the creep rate. In terms of diffusional creep this is due to the increased disruption in the stress field caused from spreading out the precipitates.

It is noted that the decrease in vacancy production could be attributed to the curved boundary of the precipitate as opposed to the straight boundary normal to the stress. Figure 43 addresses this and shows that the stress fields alone have a large influence on the reduction in the creep rate. The solid red line shows the total decrease in strain caused by a single precipitate and the dashed red line shows the increase in creep rate when stress fields are not included.

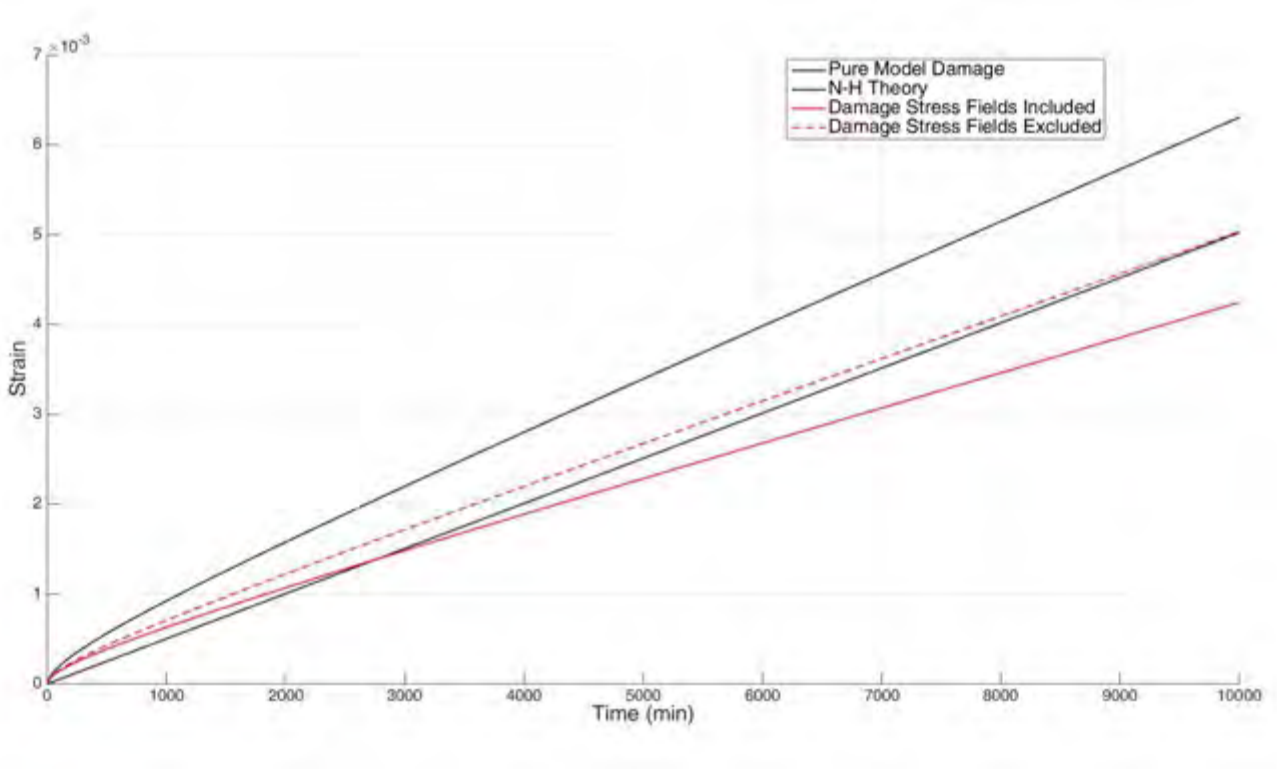


Figure 43 – Reduction in creep rate with and without including stress fields

The resulting dynamic equilibrium state of the vacancy concentration field when there are three precipitates at the boundary is given in Figure 44. This field is similar to those shown in Figure 31. The red/orange regions of this field represent the high concentrations of vacancies, while the blue regions represent the low concentrations of vacancies. What is observed, due to the stress state, is that the regions next to the precipitates have low concentrations of vacancies. This region at the side of the precipitates becomes a sink for vacancies as they would gather here and eventually nucleate into voids. This would offer one direct explanation as to why voids commonly form at grain boundaries next to second phase particles, and fits in with the theory described in the literature review.

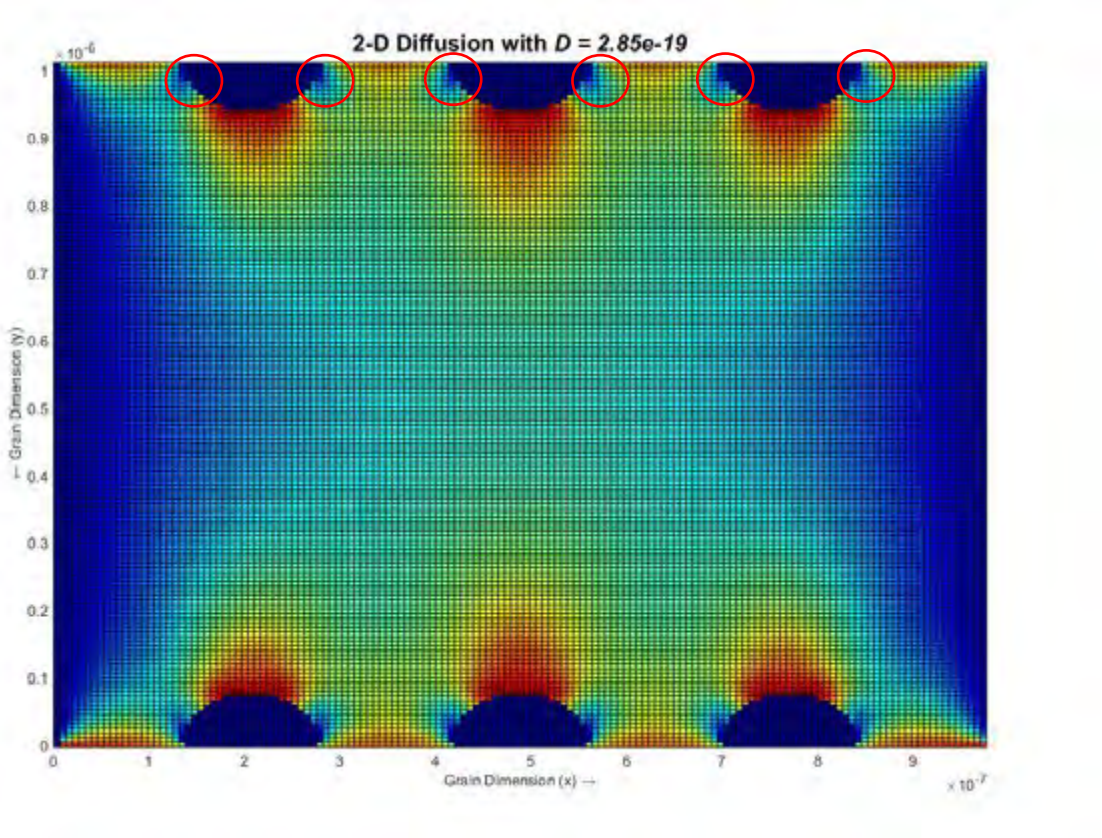


Figure 44 – Equilibrium vacancy concentration field with three precipitates at the grain boundary

A pattern arises when looking at Figure 42 and the previous figures. Under all three scenarios of adding precipitates, the resulting steady state creep rate would be identical to a curve predicted by the Nabarro-Herring equation if the applied stress were slightly lowered. This suggests that a precipitate hardening effect can be described as $\sigma_{precipitate}$. This can be included into the original Nabarro-Herring equation and would be able to give a prediction on the decrease in creep rate, which would depend on the number, mean size and Young's modulus of the precipitate without the need for a complex spatially resolved model.

The resulting equation would be given as:

$$\dot{\varepsilon} = A_{NH}\frac{D_L(\sigma - \sigma_{precipitate})\Omega}{d^2kT} \tag{6.1}$$

Further work would be needed to calculate this $\sigma_{precipitate}$ which is similar to that in section2.5.3.1 but for diffusion creep specifically.

6.3. Case 3: Heterogeneous Grain Containing Coarsening Precipitates

6.3.1. Description

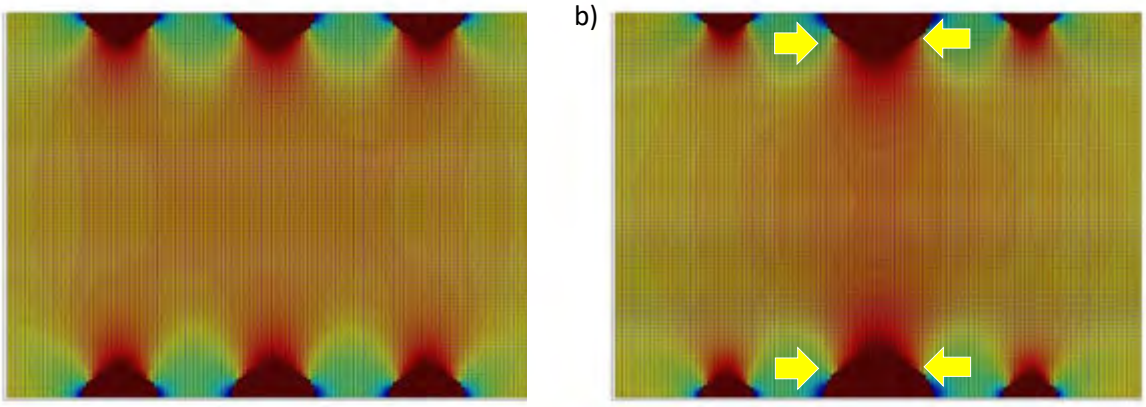
The code for this section for coarsening precipitates is found attached as **F_Creep_Precipitates_Coarsening.m**. A separate code calculating the stress fields is attached as **D_Stress_Fields_Coarsening.m**. This code follows all the grey and orange blocks in Figure 26 as well as the green blocks in the calculation process shown in Figure 28.

Coarsening is the microstructural evolution of precipitates. As precipitates have successfully been added to the model, the next step is to add this microstructural evolution. Precipitate coarsening is known to have an adverse effect on material creep strength, thus by factoring coarsening into the model, there should be an increase in the rate of creep deformation when compared to previous simulations that did not include coarsening.

As explained, coarsening is the process where smaller particles dissolve and are absorbed by larger particles. This causes a lower distribution of particles while increasing their size. This phenomenon is simulated within this model by taking the precipitates defined in the previous section and allocating which particles grow and which particles would shrink and disappear. The growth of the precipitates is calculated using equation 2.45, while the shrinking of the other precipitates is calculated from the loss of volume to the larger particles. Coarsening would continue until there is only one large precipitate remaining and all other precipitates have completely dissolved.

Figure 45 shows the process of coarsening within this model, choosing the scenario where there are three precipitates at the grain boundary. The initial configuration is shown in Figure 45 (a) where no coarsening has yet taken place; as the material is placed under operational conditions, the creep progresses and coarsening starts. Material will flow from the precipitates at the sides to the centre precipitate. This causes the side precipitates to shrink while the centre precipitate grows, as shown in Figure 45 (b). Naturally, coarsening would first occur to all the precipitates connected by a grain boundary as the grain boundary is a fast diffusion path for precipitate atoms. Once the precipitates have coarsened along the grain boundary to a single precipitate left at each face, the precipitate then coarsens through the body of the grain. The precipitate atoms would then flow through the grain causing a dominant precipitate, as shown in Figure 45 (c). Once there is only one large precipitate within the whole system, as in Figure 45 (d), coarsening stops and the precipitate no longer changes size.

a) b)



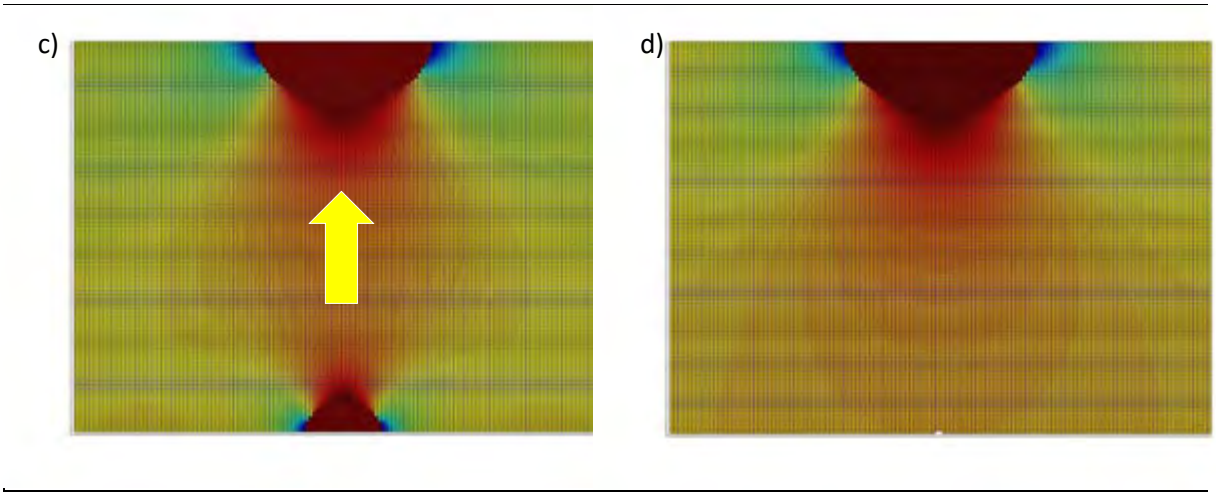


Figure 45 – Multiple precipitates coarsening into one precipitate

6.3.2. Results

The effect of coarsening has been added and the creep rates for the three scenarios of precipitates have been repeated. The results from comparing these strain rates are given below in Figures 46-48.

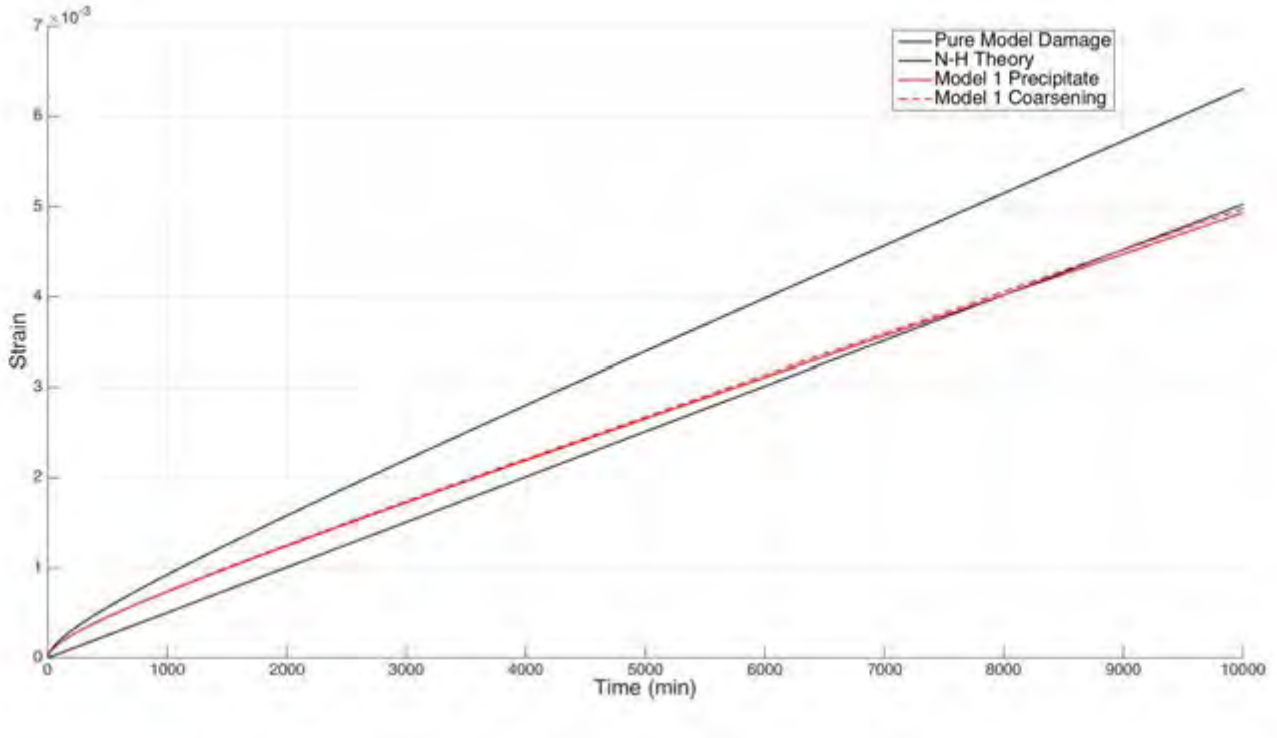


Figure 46 – Strain increase due to coarsening of one precipitate

Figure 46 gives the strain rate for one precipitate at each face. The solid red line is the creep curve for no coarsening and the dashed red line is for when the precipitates coarsens. In this scenario there is very little coarsening as there were only two precipitates in the system to begin with. There is very little difference in the gradients (with only a very slight increase in the gradient of the dashed line).

This correlates with the expected outcome of coarsening, showing that there has been a loss of creep strength and subsequent increase in the creep rate.

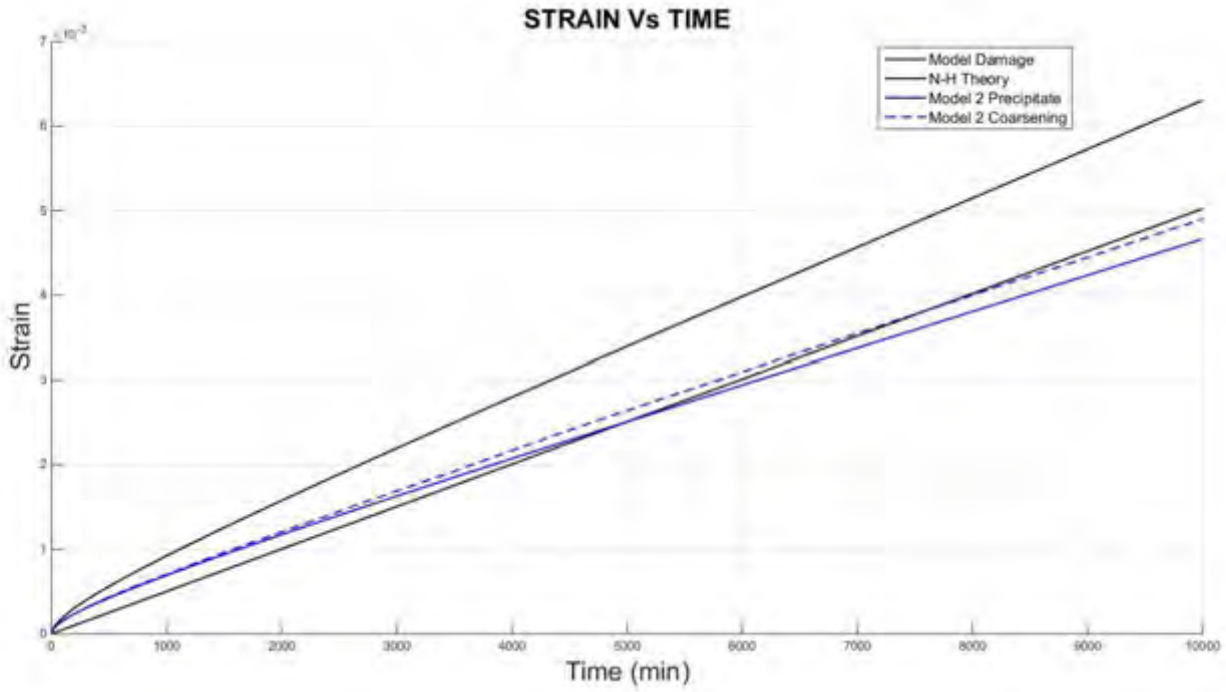


Figure 47 – Strain increase due to coarsening of two precipitates

Figure 47 represents the condition where there are two precipitates at the grain boundary, where the solid blue line is with no coarsening and the dashed blue line is where there is coarsening present. As there are more precipitates available, there would be more coarsening occurring and a greater loss of creep strength expected. Also with smaller initial precipitates, coarsening is initially more rapid than that of the previous example. They first coarsen into one precipitate at each boundary and then across the grain into a single precipitate. The bigger change in gradient shows a larger loss of creep strength when there are more precipitates undergoing coarsening.

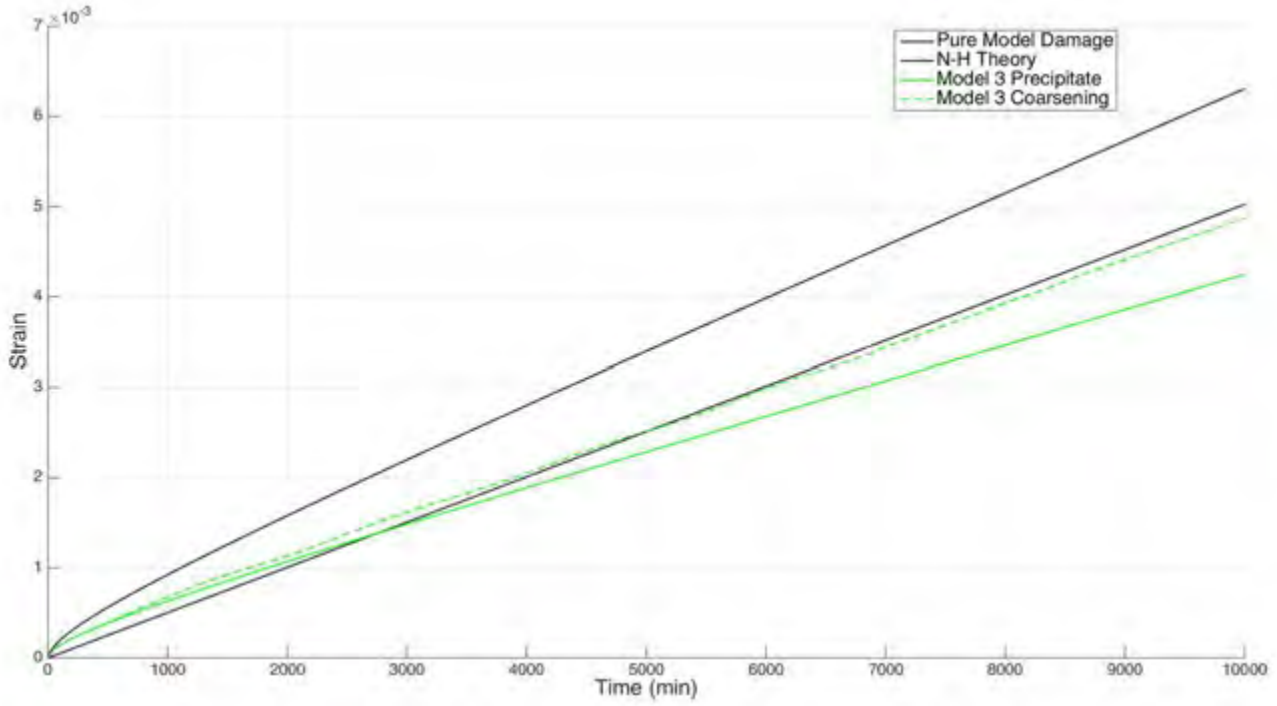


Figure 48 – Strain increase due to coarsening of three precipitates

Figure 48 represents coarsening for three precipitates. The solid green line represents no coarsening and the dashed green line is where there is coarsening present. The result is a further loss of creep strength and increase in creep rate, expected from having more initial precipitates. Again with smaller precipitates, the initial coarsening is more rapid.

6.3.3. Discussion

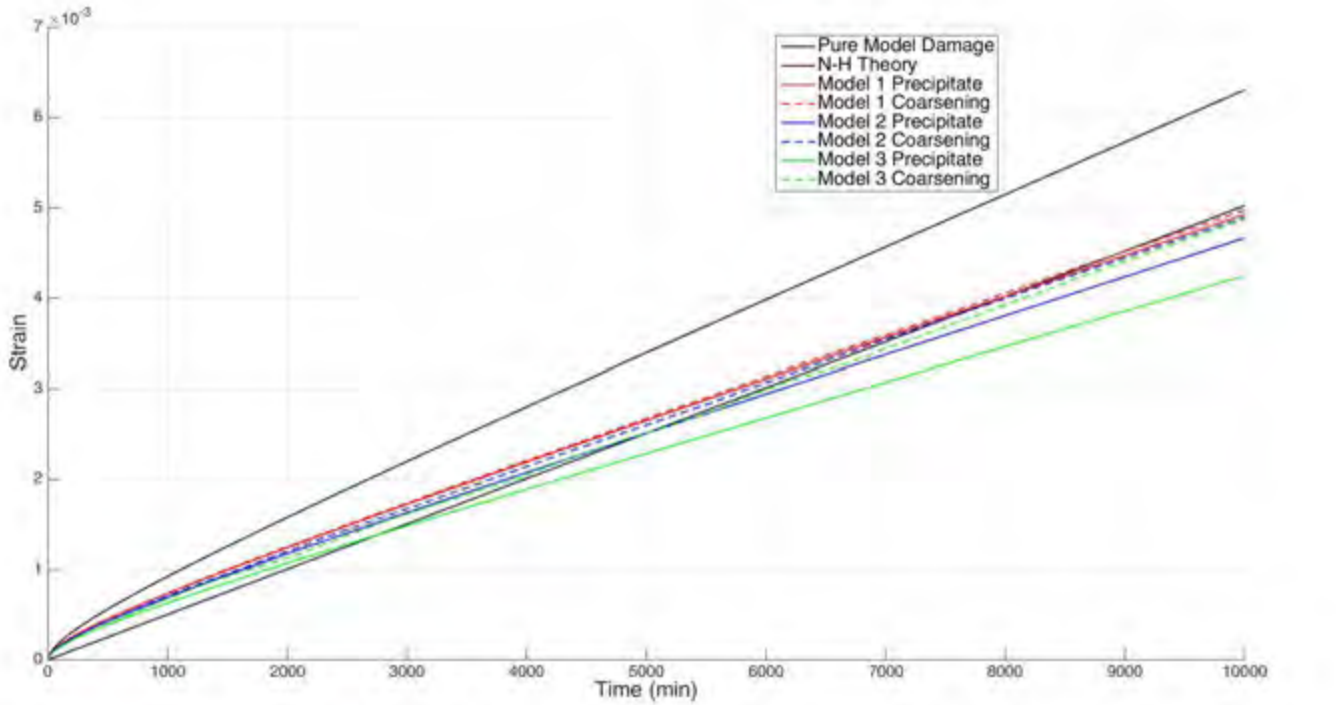


Figure 49 – Strain increases due to coarsening for all scenrios

By combining the three scenarios onto a single axis, shown in Figure 49, makes the process more cluttered but gives a clearer process of coarsening within this model. The offset between the dashed lines shows it takes longer for coarsening to complete when more precipitates are added, which correlated to observational studies. This also gives a greater scale of the loss in creep strength in each scenario, where higher numbers of initial precipitates, which cause the greater hardening against creep, have a greater loss in creep strength due to coarsening. This fits in with observations of coarsening within creep resistant steels and gives validation to this stage of the model.

6.4. Case 4: Heterogeneous Grain Containing Voids

6.4.1. Description

The code for this section for coarsening precipitates is found attached as **G_Creep_Void.m**. A separate code calculating the stress fields for voids is attached as **C_Stress_Fields_Voids.m**. This code follows all the grey and orange blocks in Figure 26 as well as the calculation process shown in Figure 29.

This section of the model looks at adding voids to the model without any void evolution or growth. The model will run using a few different sizes of voids, and even though growth is not yet taken into account, this section can give initial insight into how the presence of voids might affect diffusional creep.

The appearance of voids is considered to be detrimental to the structural stability of the material and is a major indication of the damage state of the material. As stated before, there are two types of precipitates that can be considered; voids that nucleate during the creep process and those that are already present in the new material due to the manufacturing process.

A comparison in the stress field generated around a void when compared to a precipitate of the same size is shown in Figure 50. Here it can be seen that the stress field forming around a void, on the left hand side, is many times greater than that forming around the precipitate on the right side. This difference in scale leads to the assumption that a void would have a much greater effect on increasing the number of vacancies generated, thus resulting in voids having a softening effect on diffusional creep. Due to the difference in stress fields, a void would have a much greater softening effect than the hardening effect on a precipitate of the same size.

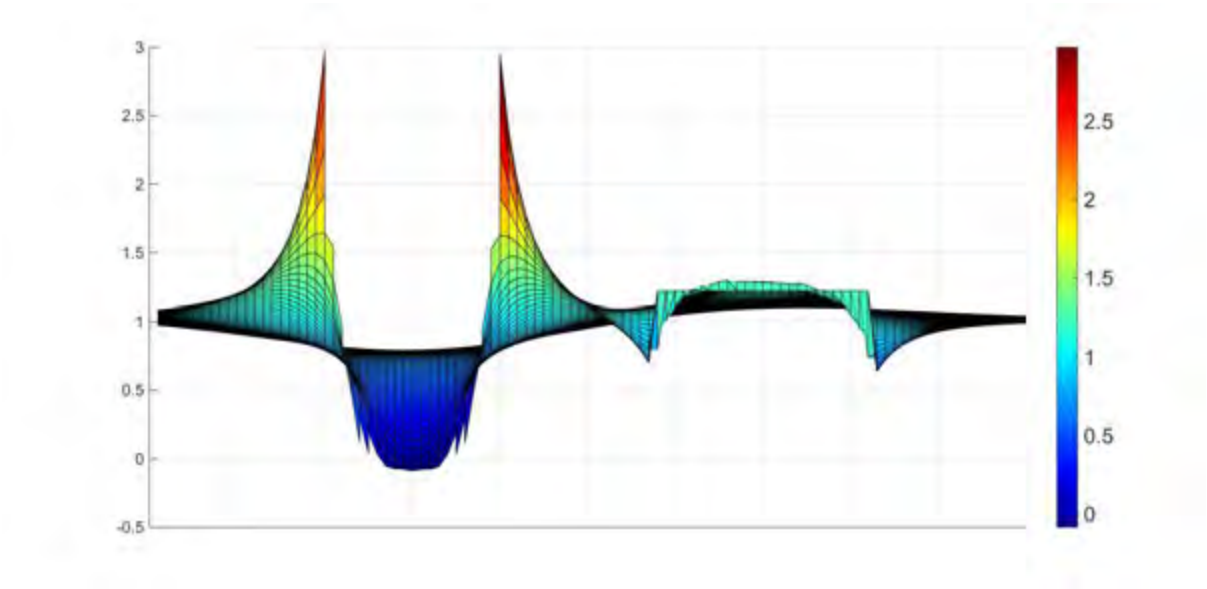


Figure 50 - Stress field of void (left) vs precipitate (right)

The nucleation of voids was not included in this model as it is a very complex process in and of itself that would be hard to define within this model. Rather, pre-existing voids are added within the system. These pre-existing voids are added at each of the four corners of the grain as shown in Figure 51. Analysing the effect four quarters of a void would have on the system would yield the same result as analysing a single void. The surface of the void becomes a source of vacancies (shown in red in Figure 51), even when the void grows to large sizes. Again, these regions are considered to be ideal sources

whilst, in reality, they may differ. The side boundaries, which act as sinks for the vacancies, decrease in size as void size increases.

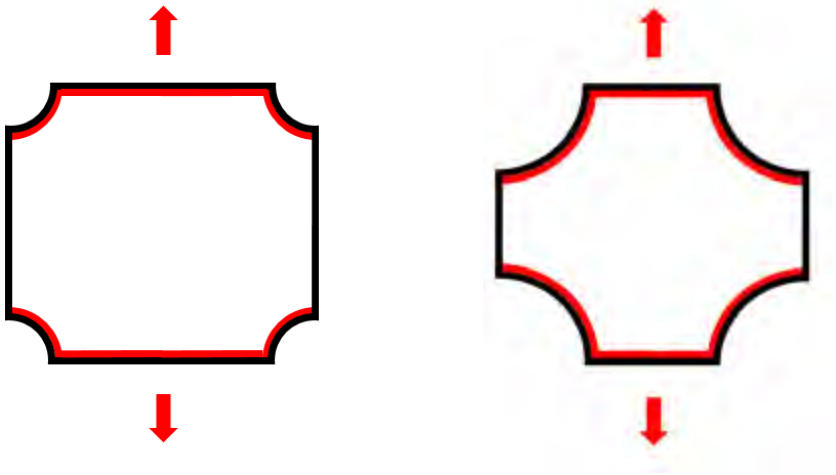


Figure 51 – Active vacancy sources in a grain containing voids

6.4.2. Results

The void sizes considered are in terms of the fraction of grain size, as the same ratios between void size and grain size produce the same effect. These sizes are 1/10th, 2/10ths, 3/10ths and 4/10ths of the grain size and are represented in Figure 52 (a) (b) (c) and (d) respectively. The left side of the figure represents the stresses along the Y axis and the right hand side represents the stresses in the X direction. Again, the stresses along the tensile axis, the Y axis, are higher than the stresses along the X axis. The stress distribution resulting from the high peak stresses at the sides of the void cause the median of the field along the Y axis to appear in blue while they are in fact higher than the stresses along the X axis on the right.

The stress concentration caused by the loss of load bearing area to the void is also included in this normalized stress field, resulting in the much higher spread in stress as the void size increases. The max stress at the peaks in Figure 52 (a) is just below 3 times the applied stress. This peak stress increases very slowly as the void increases in size until the stress fields from the voids interact, at which point the stress increases rapidly. This peaks at approximately 5.5 times the applied stress by Figure 52 (d), where the void occupied 60% of the load bearing area. The stresses along the X axis show a large amount of compression, yet due to the orientation of these stresses in relation to the curved surface, this lowering effect will be minimized when resolving to normalized stresses.

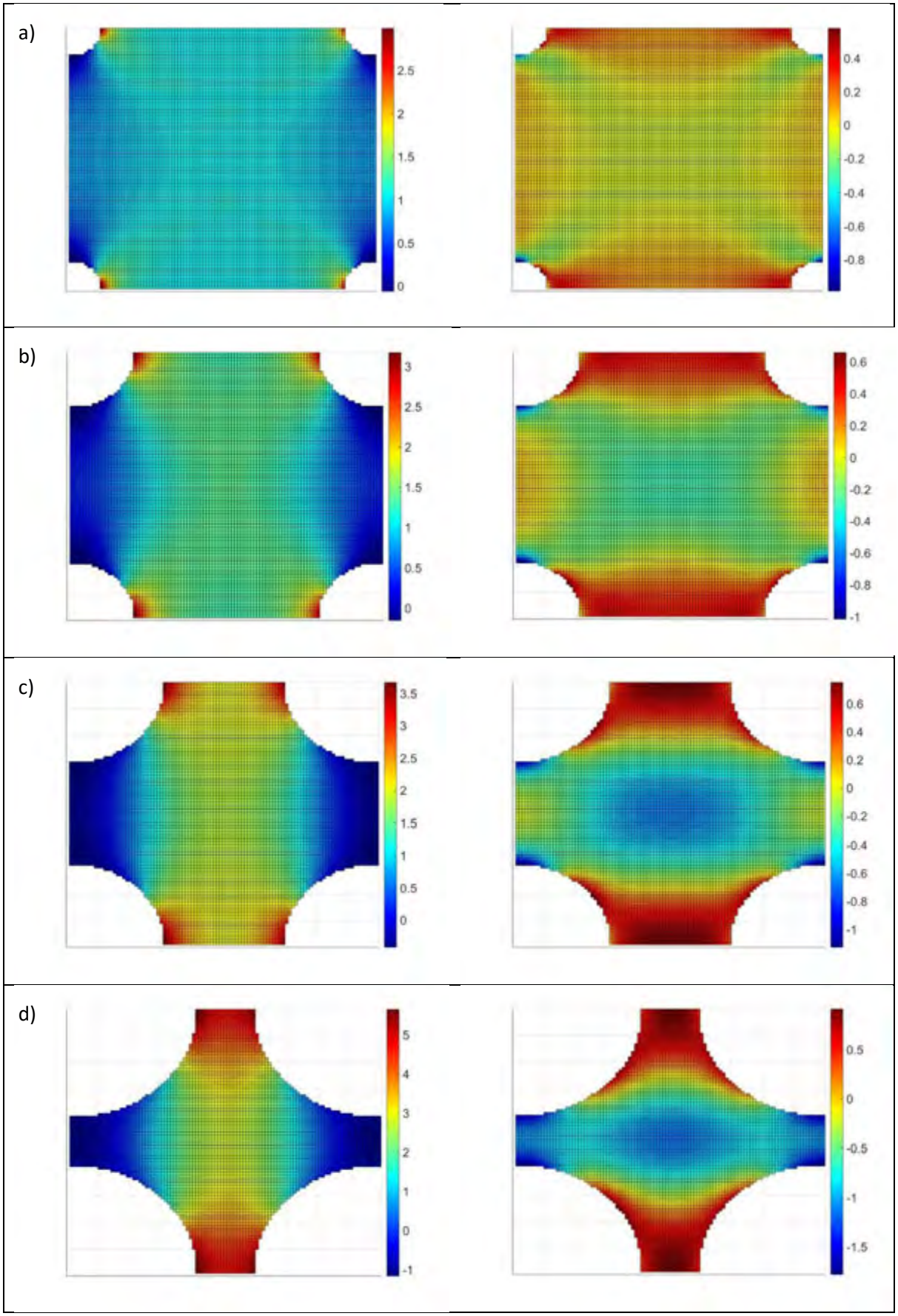


Figure 52 – Normalized Y stresses (left) and normalized X stresses (right) caused by voids

The creep curve resulting from non-growing voids in a material is shown in Figure 53 below where all voids have been added on to the same axis. The red line represents the creep rate when the void is at the smallest size possible for the simulation to remain stable, which is 2 mesh elements, or 1/50th of the grain size. The dark blue line is the creep curve for a void of 1/10th the grain size, (with stress fields shown in Figure 52 (a)). The green line is the creep curve for a void size 2/10th of the grain size, (stress fields shown in Figure 52 (b)). The cyan/light blue line is the creep curve for a void size 3/10th of the grain size, (stress fields shown in Figure 52 (c)) and the magenta/purple line is for 4/10ths the grain size, (stress fields shown in Figure 52 (d)). The bottom linear black line is the Nabarro-Herring model and the top black line is the model for a pure material with no voids; the scale used in previous sections has been reused here.

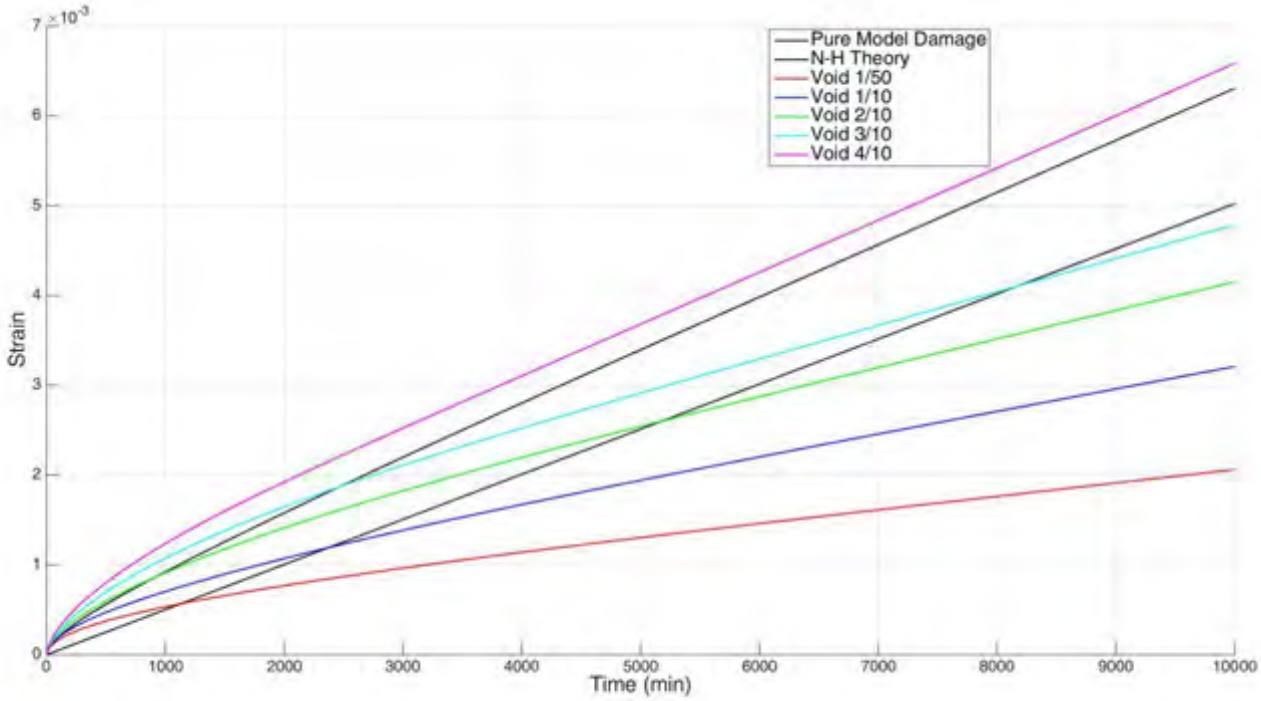


Figure 53 – Strain curves of material containing non-growing voids

6.4.3. Discussion

When looking at Figure 53 it shows an unexpected result that the initial presence of small voids slows down diffusional creep. This suggests the creep rate for a material containing voids under a certain size would be slower than a perfect material. This decrease in the creep rate could be caused by the surface between the void and material becoming a very ineffective source of vacancies. This ineffectiveness is caused by the loss of stress along the surface of the void, which can be seen as the dark blue regions next to the voids on the left hand side of Figure 52. At small void sizes the increased stress at the sides does not generate enough vacancies to overcome the loss of vacancies across the void surface.

There is no information available to verify the unlikely scenario where voids would, in fact, slow down diffusional creep initially. This loss in creep rate could also be attributed to the diffusion of vacancies occurring along the surface of the void as opposed to through the grain. Here surface diffusion, briefly mentioned in the literature review, requires much less effort than lattice diffusion. This suggests that the diffusion of vacancies in a material including voids is much more complicated than initially perceived. This loss of creep rate could be due to the model only explaining creep deformation due

purely to lattice diffusion. A less compelling explanation is that the decrease in creep rate could be due to voids absorbing vacancies from the system and causing the grains to grow, but this has no correlation to the equations used.

If the data from Figure 54 is correct then small voids are actually more successful at retarding lattice diffusional creep than precipitates of the same size. Here the blue line represents the creep curve for a precipitate with a radius 1/10th that of the grain, while the lower red line is the creep curve for a void with a lower creep rate. However, this strengthening mechanism (if it can be considered one) is not long term, nor would it be effective against any other creep mechanism besides diffusional creep.

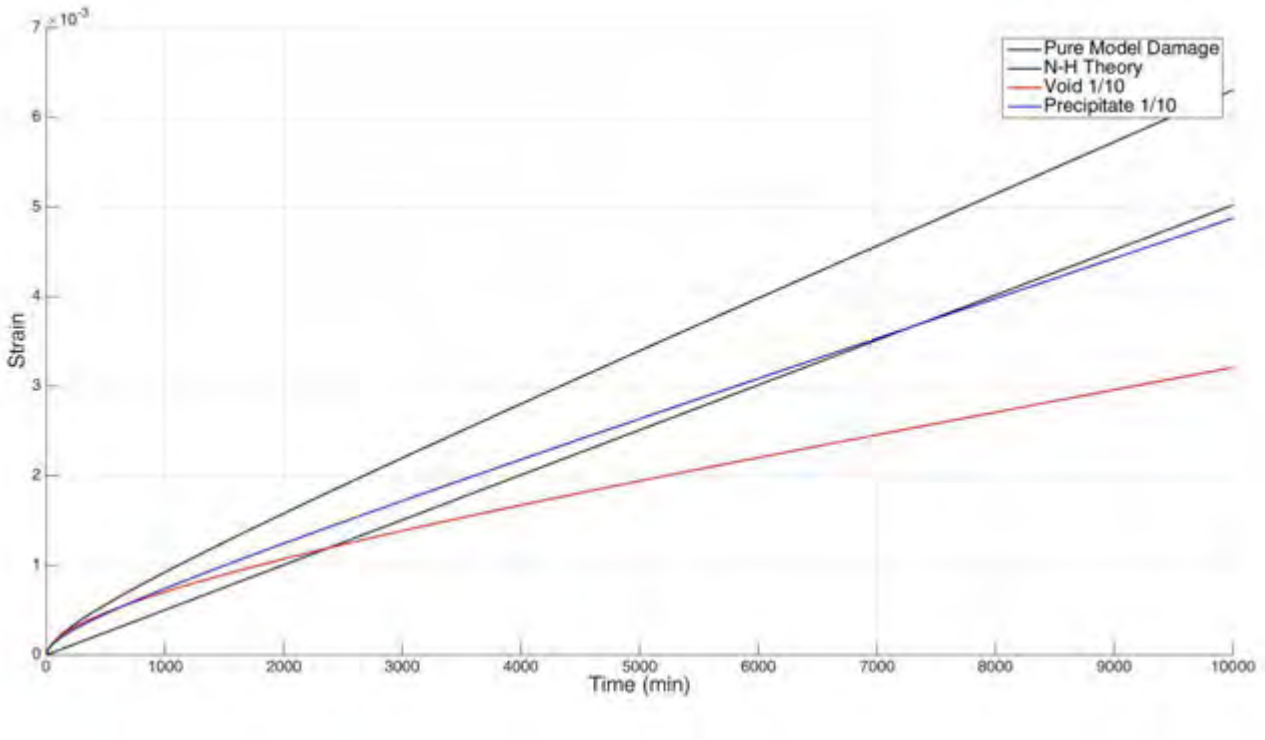


Figure 54 – Strain curves of precipitate and void of the same size

However, even though the initial presence of voids appears to have a hardening effect, as the void size is increased it shows a steeper steady state creep. This shows that void growth is detrimental to the creep strength of a material, as expected from literature. Once the void has reached a certain size within the grain the stress will increase enough to balance out the loss of void generation across the void surface and give a creep rate equal to that of a pure material. According to Figure 53 this is when the void size is above 4/10ths of the grain size, so roughly when the void has grown to occupy over 80% of the load bearing area. This is where the stress concentration would cause vacancy generation that is rapid enough to lead into the tertiary stage of creep. From the rapid increase at this void size, damage is seen to have come into play and caused an exponential increase in the stress at the grain boundary. At this point the stress increase due to damage is finally able to overwhelm the ineffectiveness of the void surface as a source. Another effect of increasing the stress due to damage is that other creep mechanisms, such as dislocation creep, can now come into play as the stress is now high enough. This means that this model would lose accuracy with increasing stress. However as these diffusional mechanisms are considered independent, this model can offer predictions to the part of the total creep caused by the vacancy flux and diffusional creep specifically.

6.5. Case 5: Heterogeneous Grain Containing Growing Voids

6.5.1. Description

The code for this section on growing voids is found attached as **H_Creep_Void_Growth.m**. This code follows all the grey and orange blocks in Figure 26 as well as the green microstructural evolution calculations process shown in Figure 29.

In a realistic situation voids would not remain the same size and would be subject to microstructural evolution, namely growth. This type of growth at low stresses occurs due to the absorption of vacancies and is therefore a diffusion controlled process. The number and size of voids is an indication of the damage level of the material. As the voids grow in size, so too would the damage level of the material. This damage causes a loss in load bearing area and a subsequent increase in the stress experienced by the material. The increased stress leads to a rapid increase in creep rate until material failure, making void growth and damage the main contribution for tertiary creep. The growth rate is also proportional to the stress, thus as a void grows and damage increases, the growth rate of the void would increase. Conversely, the growth rate is also inversely proportional to the size of the void, which slows the growth rate somewhat as the void size increases.

The previous section showed how the presence of voids would affect the diffusion creep process, where small voids would have a hardening effect on lattice diffusional creep but larger voids would have a weakening effect. Large voids are not generally found in the as-received material, but rather grow from smaller voids as they absorb vacancies. As smaller voids have a strengthening effect, the creep rate is expected to be lower than that of a pure material. However, the growth of a void is seen as a softening effect so, as the creep process continues and the void grows, the creep rate would increase. Eventually, after the void has grown large enough, the hardening effect from the void surface is overwhelmed and the creep rate would increase rapidly into its tertiary stage and until failure. Adding the growth equation into this stage of the model gives the incremental effects that void size would have on the creep rate as this void undergoes growth throughout the creep process.

The initial conditions for void growth are kept the same as the previous sections where the voids are present at each of the four corners, they are able to grow inwards into the grain, increasing stress and creep rate as they grow. The voids are given the smallest available starting size, which has been limited by the size of the mesh chosen. The original void has to be larger than a single element of the mesh, else the simulation becomes unstable. As the mesh is 100 x 100 the initial void size is set to 2/100ths of the grain size. These voids are able to grow until they meet at the grain boundaries.

6.5.2. Results

The results of adding void growth is shown in the figures overleaf. Void growth times have been accelerated in an attempt to speed up the simulation as computational time is extremely high at this point. The red line in Figure 55 is the creep curve generated from having a small void that grows until just before material failure. In this simulation the creep rate was initially lower than the Nabarro-Herring theory due to the initial hardening effect of voids, as predicted. Here the growth of the void slowly lessens the hardening effect previously caused by the void surface. The tertiary stage of creep and failure occurs very rapidly, due to the final damage equation superimposed on top of the stress field calculations. Observation of the primary creep stage is almost non-existent at this time scale.

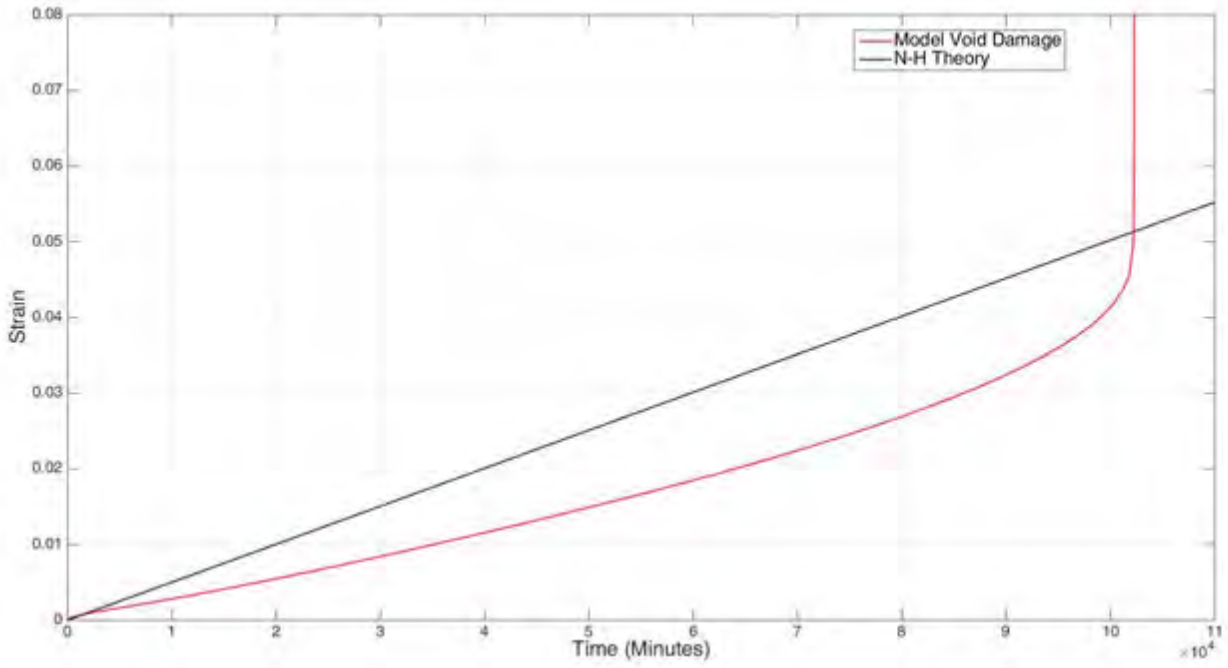


Figure 55 – Strain curve resulting from void growth

In Figure 56 the growth equation is not as accelerated as in Figure 55. Again the primary creep stage due to diffusional creep is basically non-existent. Here the red line has a more gradual increase into the tertiary stage. This is due to the growth rate being slow enough to allow stress increase due to grain thinning to come into effect. At this stage the computational time is too long to have a slower growth rate.

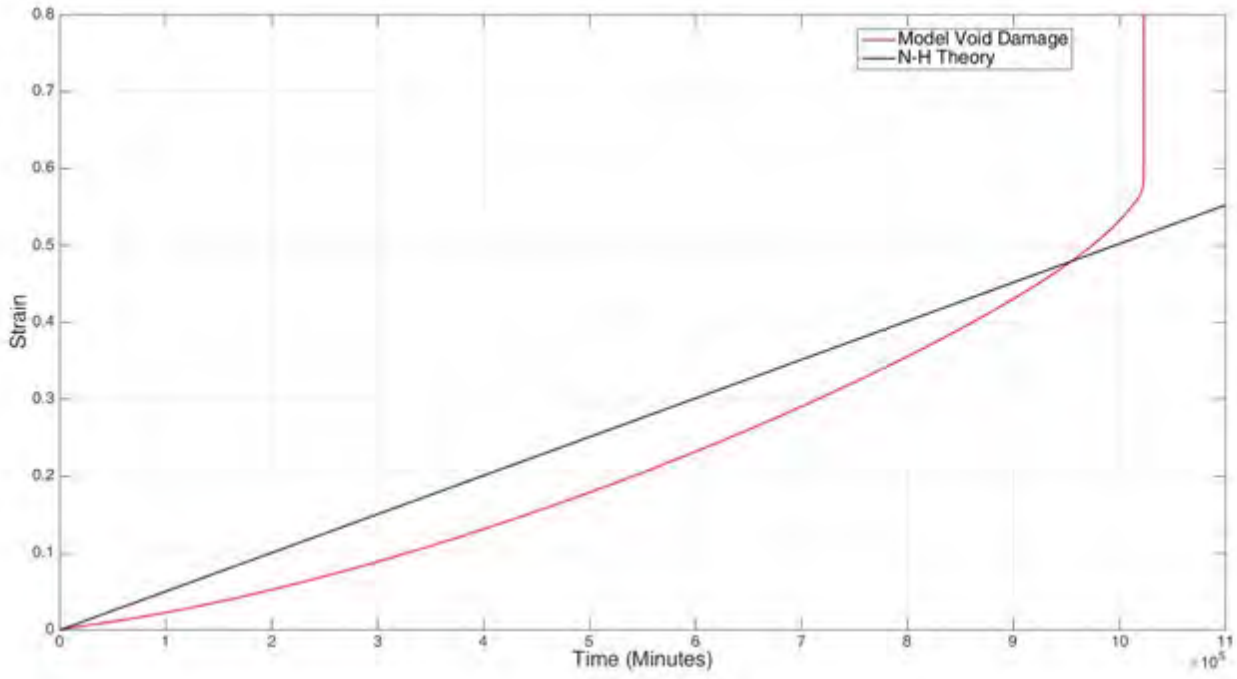


Figure 56 – Strain curve resulting from void growth and thinning

6.5.3. Discussion

The previous graphs demonstrate that the model is able to support a tertiary stage of creep by monitoring only the vacancy movement. Following the Nabarro-Herring model and changing it to include Fick's second law allows for a much more diverse creep curve to form. Definite movement into the tertiary creep is observed. The presence of voids initially slows down creep, but, as the voids enlarge, they cause the creep rate to increase as they increase the stress. However, the stress concentration needed to increase vacancy production enough for any form of tertiary creep to develop would cause other creep mechanisms or even plastic elongation to come into effect, resulting in a highly unpredictable region.

This simulation only shows the effect that a small number of voids would have, as the number of voids increases, the time to failure would be shortened. By analysing the void growth equation, equation 2.49, adding more voids to the system would both increase the damage-stress relationship as well as decrease the distance between voids. The result is that the more voids there are, the quicker they will grow. This means that the rate to failure is heavily influenced by the number of voids present in the grain. More voids would mean much shorter times until failure. As discussed and reinforced before, the nucleation zones of voids are likely to occur on grain boundaries at the precipitates. Theoretically, this may mean that higher numbers of precipitates may lead to a larger number of nucleation sites and, therefore, more voids would form; resulting in a much steeper gradient increase into the tertiary creep stage. Ultimately, this section provides evidence that the creep due to movement of vacancies by means of lattice diffusion would be slower in a material containing voids.

6.6. Case 6 : Heterogeneous Grain with Microstructural Evolution

6.6.1. Description

The code for the section on coarsening precipitates together with growing voids is found attached as **I_Creep_Precipitate_Void.m**. This code follows all the steps set out in Figures 26, 28 and 29.

Thus far the model has considered voids and precipitates in separate systems. In the final section of the code these two are added to the same system where the presence of voids and precipitates would affect the creep process at the same time. Here, the effects of precipitates and voids on diffusional creep would be compounded.

Here all precipitates and voids are added to the same system and are still kept in their original positions from the earlier section. The voids are kept at the corners and precipitates would still be on the transverse grain boundaries. The effective sources of vacancies include the grain boundary, the void surface and the precipitate matrix interface as shown in red in Figure 57. Here the precipitates would undergo coarsening while the voids would be subjected to growth. The precipitates would be considered fixed to their position so if the void grew large enough it would swallow the precipitate and remove its effect from the system. No new equations or constants are added in this section.

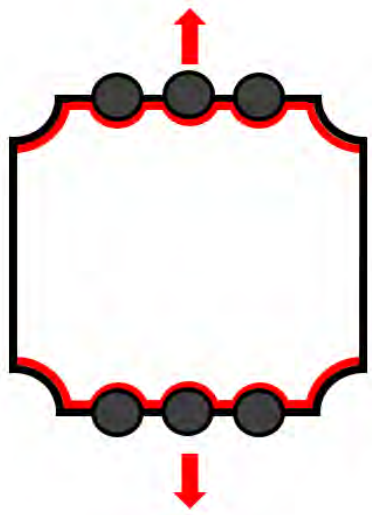


Figure 57 – Active vacancy sources in a grain containing precipitates and voids

6.6.2. Results

The following images are the stress fields resulting from adding voids and precipitates into the same simulation. Subsequently, the resulting creep curves produced in the system containing both of these elements is shown. At this stage computational time is the sum of all individual sections, roughly averaging 90 minutes per 10000 iterations.

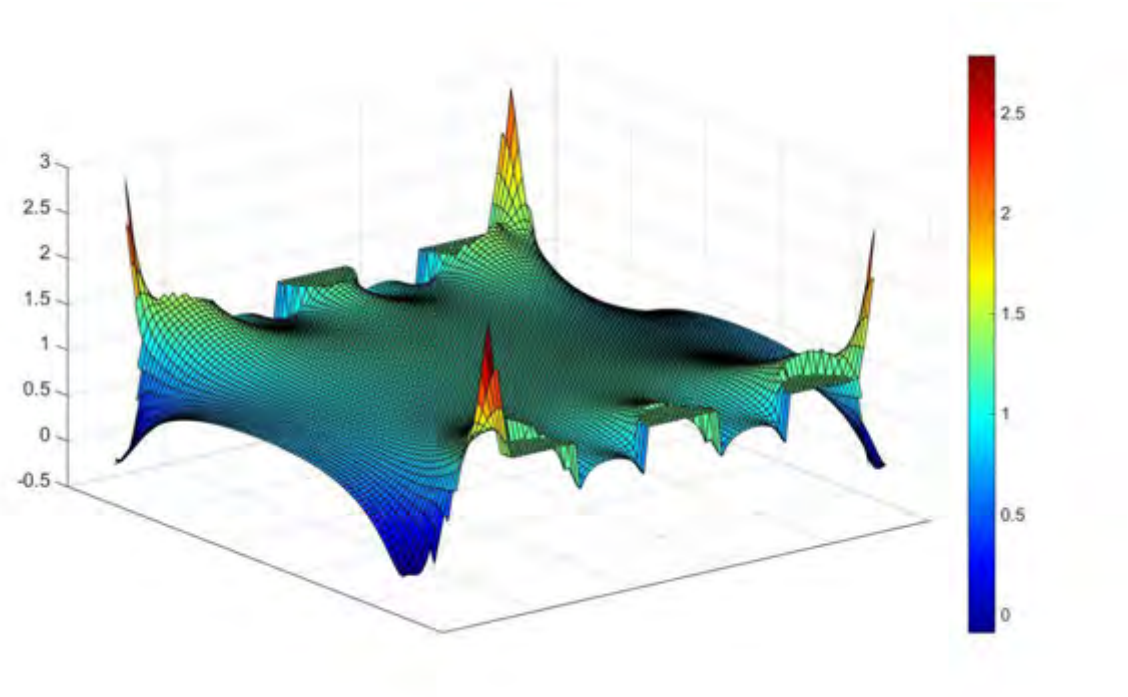


Figure 58 –Stress field of void interacting with precipitates

Figure 58 is the stress fields shown when the voids and precipitates start to interact. Up to this point the creep process had progressed normally. At this stage the surface of the void and the interface between the matrix and precipitate start to interact and merge, this effect is significant and will be explained later. The peak stresses from the voids overshadow the stresses around the precipitate, but the size of the void is not large enough to cause any increase in the applied stress. The interaction between the stress fields from the void and precipitate cause the normalized peak stresses to dip below the maximum of three.

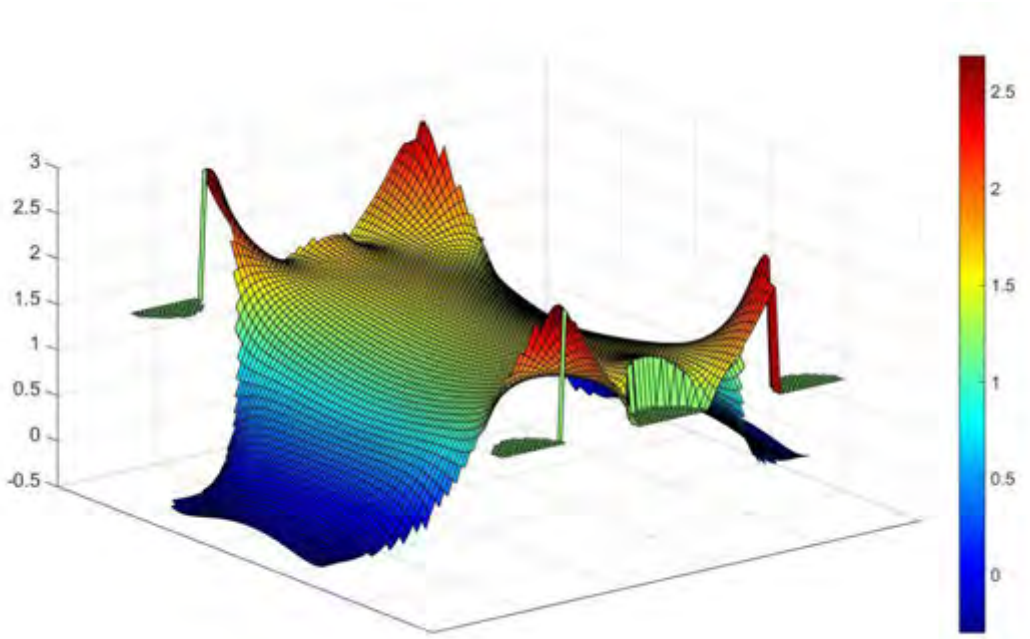
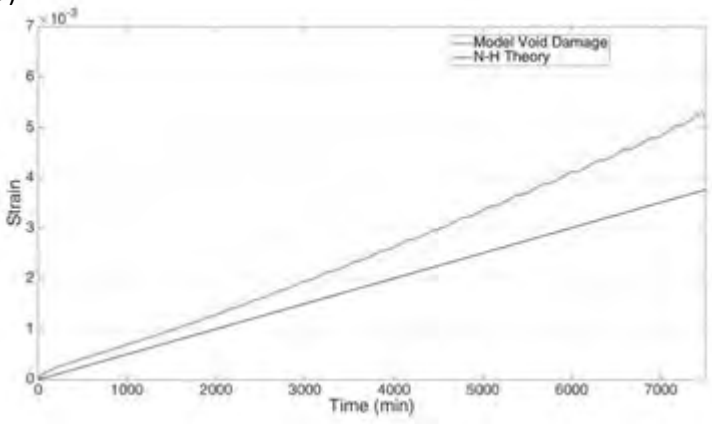


Figure 59 – Stress field of large void absorbing precipitate

Figure 59 shows where the void has grown large enough to completely absorb the precipitate. At this point the effect of the absorbed precipitate is removed from the system and the void surface becomes the only active vacancy source.

(a)



(b)

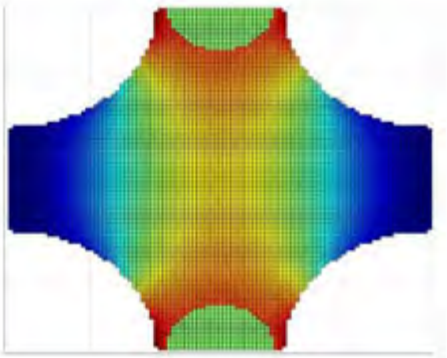


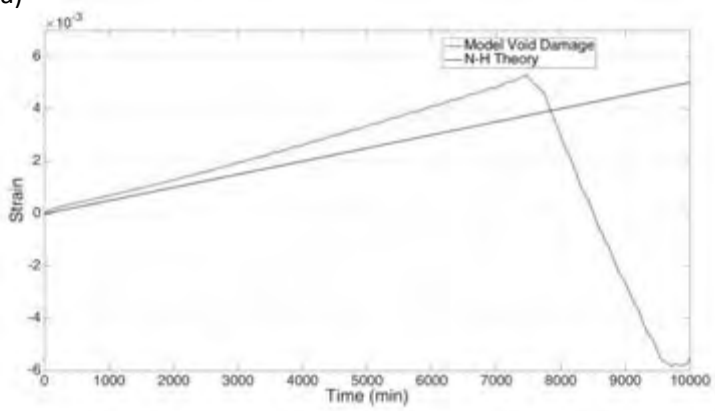
Figure 60 – Creep curve for precipitates and voids in the same system

Figure 60 (a) represents the creep curve of the scenario where there is one precipitate at each face and voids are able to grow. The creep curve progresses as expected where the creep rate increases as the void size increases. This part runs up until the stage right before the void and precipitate would interact, as described in Figure 58. The spikes in the creep curve are caused by the stress increase every time the void grows one element larger. They are predominant in this example because the mesh is too coarse and the growth rate of the void has been heavily accelerated. They would not appear over a longer simulation.

Figure 60 (b) shows the stress field around the precipitate and void after the creep has progressed up to the point shown in Figure 60 (a) . Here the void and precipitate do not touch and there is enough space (dependent on mesh size) between them for the flux of vacancies to be calculated.

Figure 61 (a) below shows the creep curve a few time steps later in the iterations. The creep curve becomes unstable and shows a complete loss of information. Figure 61 (b) then shows the stress fields at the time of the creep curve in 61 (a). This shows that the loss in stability occurs over the time where the void and precipitate are in contact with each other. This shows the difficulties faced in this section. The interaction of voids and precipitates cancels out information on the vacancy generation, thus causing the code to become unstable.

(a)



(b)

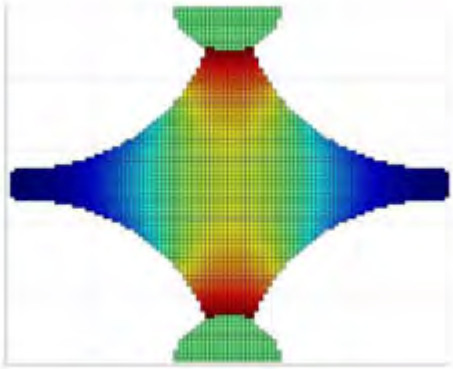


Figure 61 – Creep curve with loss of stability

An attempt was made to address this loss of stability in the code. Figure 62 below shows where code has been altered to account for the loss in creep information and keep stability when the voids and precipitates come into contact. The creep curve given in the red line proceeds as expected and is seen to stay under the Nabarro-Herring creep prediction for a long time, which is expected when the precipitate and small voids would both cause slowing of the creep rate. It then has a much more rapid advancement into the tertiary stage of creep than the previous simulation cases.

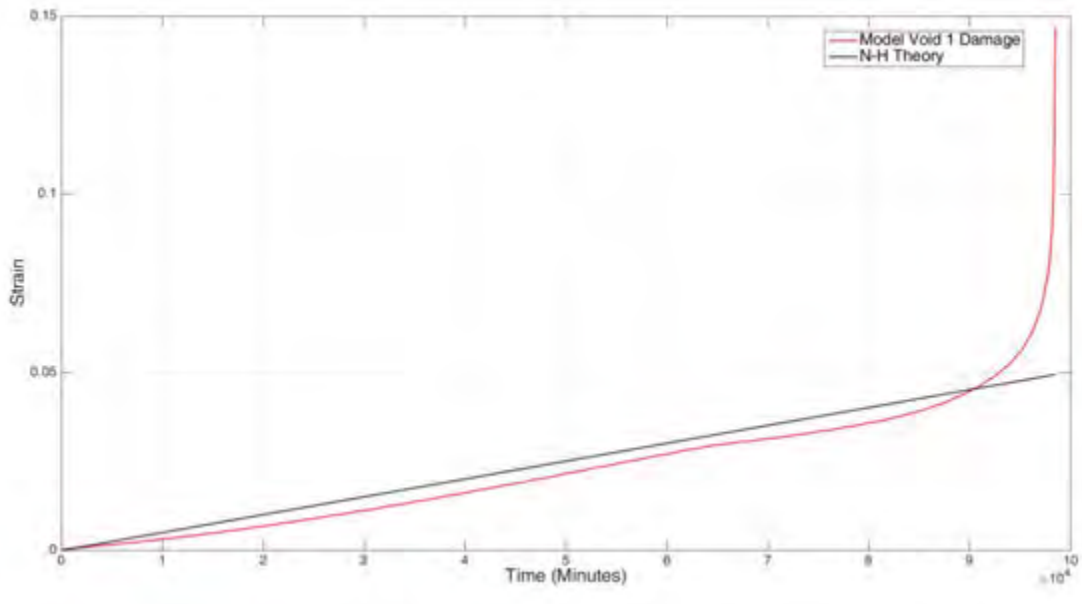


Figure 62 – Creep curve with a fix for stability

Figure 63 shows in blue when two precipitates are added to the boundary; the position and size of the precipitates are now different to the previous code. The blue line shows the loss of information again

once the void and precipitate interact, and where the code returns to normal once the precipitate has been absorbed. This shows that the fix that was used for one precipitate previously does not work for other situations. It results in the loss in the ability of this type of model to show the effects of different microstructural elements combined into the same system. Figure 63 provides evidence that the situation for adding different microstructural elements would need to be altered for every scenario, which is not a feasible outcome. Therefore, a different method may be needed to include both microstructural elements into the same system.

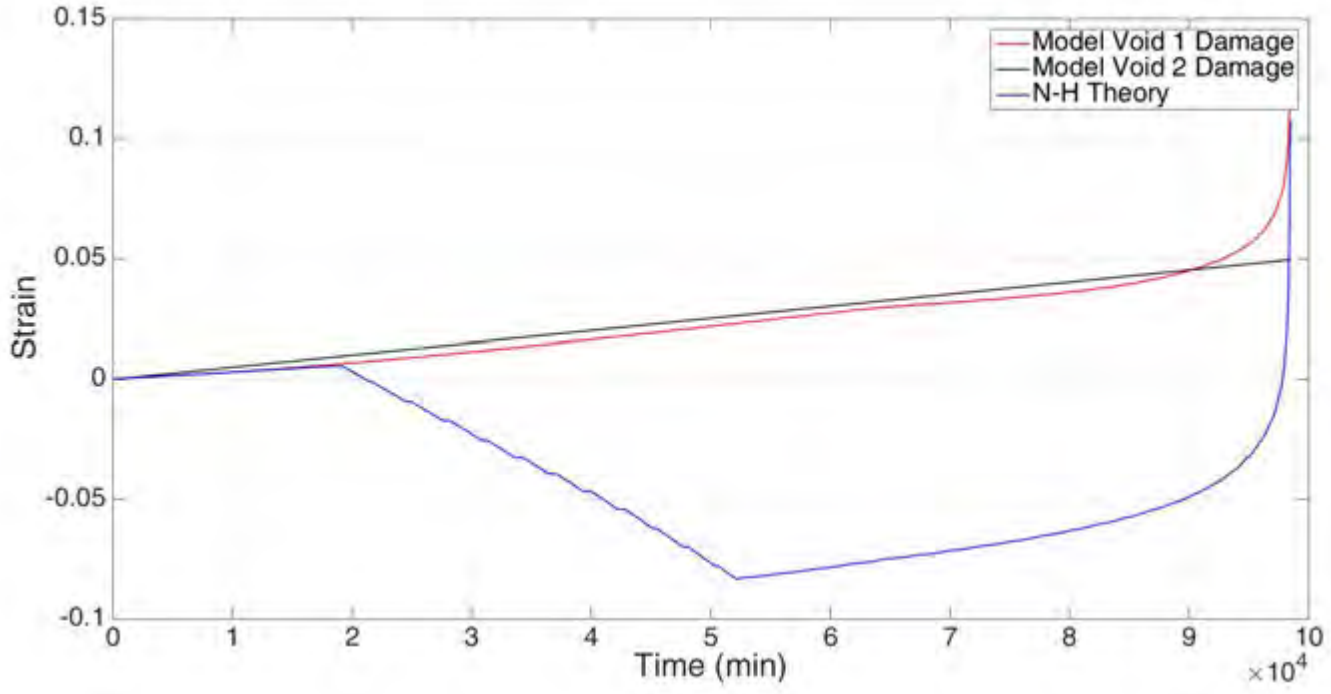


Figure 63 – Creep curve with for additional precipitates

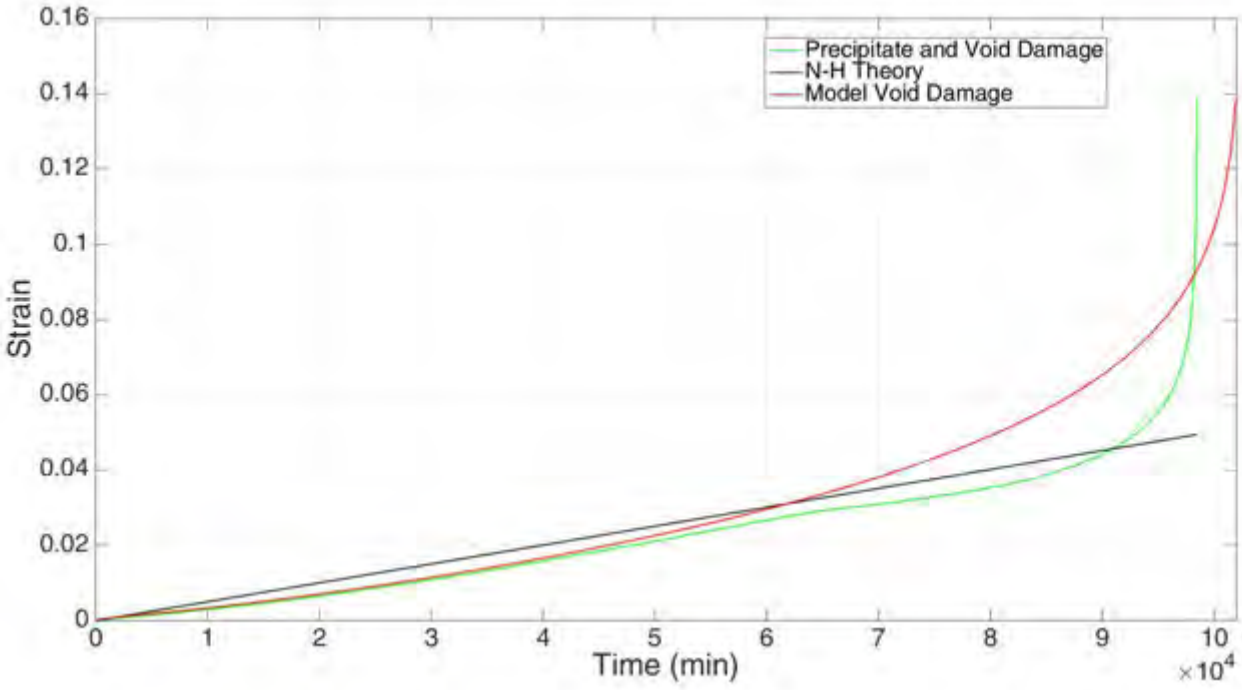


Figure 64 – Creep curve comparison between precipitates and voids in the material or voids only

Figure 64 shows the final simulation where the creep rate for the working scenario of one precipitate and void, shown in green, is plotted against the scenario where there are only growing voids in red, taken from the previous section. This shows that the addition of voids and precipitates into the same system causes a lower creep rate for a longer period. It also results in a much steeper transition into tertiary creep than for the scenario not containing precipitates. The shorter rupture time and steeper transition into the tertiary stage would be due to the stress concentration from the voids converging on the higher stress state at a precipitate, resulting in stress compounding into a much higher state.

6.6.3. Discussion

This section gave the expected result. When combining the two processes, their effects on the creep deformation had an additive effect. The addition of precipitates leads to a longer steady state creep regime before tertiary creep is reached. However the tertiary creep curve is markedly steeper. Nevertheless, it proved difficult to combine all processes once they came into contact with each other. This is due, in part, to the limitation of the matrix size; when the parts were within one element of each other, information would be lost. Another part is due to the way the effect of voids and precipitates has been calculated. When the elements overlap, the data calculated from one element would alter the data calculated from the other, meaning information from one of the processes is lost. This loss would result in a steep drop off in the creep rate. There have been measures to remedy this but they were not able to fix the underlying problem. This means that the previous sections describing the separate effect from voids and precipitates cannot be included in the same system using the current method. Rather, as the effect of each element overwrites each other, their effects need to be separated from the overall creep rate and added separately. The suggested method is to calculate the strengthening effect from precipitates, proposed as equation 6.1 in section 6.2. Using the decrease in creep rate from that equation and adding it to the current model on calculating the effect of voids would allow for both microstructural elements to be added to the same system without the loss of stability in the code.

7. Summary

The locally varying microstructure of high chromium steels has been taken into consideration within this model. The already accepted Nabarro-Herring theory has been altered to account for the local changes in vacancy concentration over time. This was accomplished by introducing Fick's second law of diffusion into the vacancy concentration equations. This change enabled the model to describe the primary, secondary and tertiary stages purely for diffusional creep. These three stages of creep have been generated purely from tracking vacancy motion away from the tensile faces of the grain. The primary creep stage caused by diffusional creep is inconsequential; the primary creep stage in a real life experiment would be caused by other factors and not influenced by diffusional creep. The tertiary creep stage was found to appear very late in the materials lifetime, >0.9 t_r, at which time the stress would already be high enough to cause plastic deformations.

The model has shown how the inclusion of microstructural elements would affect the diffusional creep process purely by examining how they alter the stress states within the material. These main elements considered are the precipitates and voids, as well as their microstructural evolution connected to the creep lifetime.

It was not previously agreed upon how precipitates slow down diffusional creep in particular. The leading theory was that the interface between a precipitate becomes a poor source of vacancies due to the energy state at this boundary. This project rather proposed and gave supporting evidence, by means of simulation, that this hardening against diffusional creep is due to the stress fields that form around the precipitates as a stress concentration. This decrease in the creep rate from precipitates is directly caused by lowering the stress in the grain boundaries near the precipitates, resulting in fewer numbers of vacancies produced, and thus lowering the diffusion rate.

Continuing from this observation, the model has shown that the coarsening of precipitates is detrimental to the creep strength of the material by means of lowering the distribution of stress fields caused by the precipitates. By coarsening into fewer particles, the reduction in stress at the grain boundaries is lowered.

An unexpected discovery in this model is that the presence of small voids actually has an initial hardening effect against diffusional creep. This is caused by the loss of the void surface as an effective source of vacancies due to the loss of stress along this surface. However, the growth of these vacancies does have a softening effect as previous experiments have documented. The stress increase due to the concentration around voids causes a moderate increase in the generation in vacancies and subsequent increase in diffusional creep rate. It is only later, once damage not accounted for by the stress field equations comes into play, that there is the creation of the tertiary stage of creep and subsequent material failure.

The combination of voids and precipitates in the same system presented problems in the loss of stability in the code. This instability was caused when the different elements come into contact and the data from one element over writes the other, as they are both calculated in the same way. It was therefore suggested to separate the effect of precipitates, as it can be seen as a constant stress lowering effect. This effect can then be added to the result from the standard model used for voids only. In this way the effects of both microstructural elements would be represented in the same system.

In general the modelling of this process was very computationally expensive which limited the result. Adding the time step to increase in minutes rather seconds sped up the process as much as possible while still remaining low enough to keep the code stable. Speeding up the simulation had no effect on

the secondary stage of creep, as it is linear, and it had very little effect on the slow approach to the tertiary stage. However, it would have a much larger effect on the primary stage of creep, where the vacancy flux changes rapidly. This means for better results, a more powerful computer is needed that can handle the removal of this time step whilst maintaining an acceptable computational time. An increase in mesh size would also give much better results; however, this was also heavily limited by the current computational power. Real service material would also be much more complex with many more influencing factors. However, this does give insight and explanations into diffusional creep in precipitate hardened material, the understanding of which is very incomplete.

8. Conclusions

The results of the FEM simulations in this study demonstrate the diffusion rates of vacancies in a material with a locally varying microstructure. From these results a number of conclusions can be made:

- Adapting to Fick's second law allows for the Nabarro-Herring based model to display primary and tertiary stages of creep in addition to the initial secondary creep stage.
- The primary stage of creep created by lattice diffusional creep is found to be miniscule over long term creep exposure.
- The spatially inhomogeneous stress fields created by precipitates at the transverse grain boundaries in high chromium steels cause a hardening effect against creep by means of lowering the vacancy generation.
- The effect of precipitates on the diffusional process could be analysed into a creep strengthening mechanism or back stress, $\sigma_{precipitate}$.
- Coarsening causes the reduction of these stress fields and as a result causes a loss in creep strength.
- Voids also create an inhomogeneous stress field inverse to that of precipitates. These stress fields were theorized to increase the vacancy generation and thus accelerate creep but this was found to not be the case. The stress fields at the sides were not high enough to compensate for the loss in vacancy generation at the void surface for small voids. Thus, small voids lowered the initial creep rate.
- This lowering effect was found to be minimized as the voids grew larger, especially as damage came into play and complemented the boundary stresses enough to increase the vacancy generation.
- Combining the effect of both precipitates and voids results in the additive effect of their individual effects on the creep rate.
- The loss in code stability when precipitates and voids are added to the same system can be overcome by calculating the precipitate effect separately and adding this to the creep rate for voids.
- The FEM process chosen is very computationally expensive, resulting in time limited results.

9. Recommendations

This project took the first steps in describing diffusional creep in complex microstructures. As such it only addressed simple, ideal microstructures and configurations. Thus to get a greater understanding and prediction of creep, the simulations have to move towards more complex and realistic microstructural representations. The following recommendations are made for future improvement of the model:

- Investigation of how grain shape and orientation affect the diffusional creep process. Here only cubic grains were utilized to simplify vacancy generation due to stress. As only stresses normal to a surface affect vacancy generation, the curvature of the grain boundary would have a lowering effect on the creation of vacancies. Similarly, the orientation of the grain lattice would have a similar effect on the generation of vacancies.

- Evaluate the addition of multiple types of precipitates other than just $M_{23}C_6$. This includes the small, stable MX precipitates as well as the Z phase and Laves phases that precipitate and grow during creep exposure. In this study precipitates were placed only at the tensile boundaries, where they were theorized to have the greatest effect. Changing their location within the grain may yield different results.

- The inclusion of precipitates was seen to have a constant effect on decreasing the creep rate of the material. This effect can be seen as a back stress concept which limits vacancy generation. Thus this effect can be investigated and formulated with results dependent on precipitate size, number, and hardness with respect to the matrix.

- The addition of the hardening effect of precipitates (mentioned above) to the model on creep in a material containing voids would allow for both the effects of the precipitates and the voids to be used in the same system, without a loss of stability in the code.

10. References

[1] M. Sibanda, “Design of a Ferritic Martensitic Creep-Resistant Steels,” M.Sc Thesis, University of Cambridge, 2013.

[2] F. R. N. Nabarro, “Creep at very low rates,” *Metall. Mater. Trans. A*, vol. 33, pp. 213–218, 2002.

[3] F. Abe, “Introduction to creep resistant steels,” in *Creep Resistant Steels*, F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 3–14, 2008.

[4] F. R. N. Nabarro and F. de Villiers, *Physics Of Creep And Creep-Resistant Alloys*. London: Taylor & Francis, 1995.

[5] T. Ohtani, “Creep-induced microstructural evolution and acoustic characterization in tempered martensitic stainless steel,” *Metall. Mater. Trans. A Phys. Metall. Mater. Sci.*, vol. 38 A, no. 7, pp. 1587–1597, 2007.

[6] T. G. Langdon, “Identifiying creep mechanisms at low stresses,” *Mater. Sci. Eng. A*, vol. 283, no. 1–2, pp. 266–273, 2000.

[7] M. F. Ashby, “A first report on deformation-mechanism maps,” *Acta Metall.*, vol. 20, pp. 887–897, 1972.

[8] K. Rastogi, “Mesoscopic Simulation of Grain Boundary Diffusion Creep in Inhomogeneous,” M.Sc Thesis, Louisiana State University, 2005.

[9] K. Maruyama, “Fundamental aspects of creep deformation and deformation mechanism map,” in *Creep Resistant Steels*, F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 265–278, 2008.

[10] K. G. F. Janssens, D. Raabe, E. Kozeschnik, M. A. Miodownik, M. Upmanyu, and B. Nestler, *Computational Materials Science An Introduction to Computational Microstructure Evolution of Polycrystalline Materials*. London: Academic Press, 2009.

[11] H. Okinwa. and Y. Iijima., “Diffusion behavior of creep resistant steels,” in *Creep Resistant Steels*, F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 241–264, 2008.

[12] J. Crank, *The Mathematics of Diffusion*, 2nd ed. London: Oxford University Press, 1975.

[13] H. Mehrer, *Landolt-Bornstein: Numerical Data and Functional Relationships in Science and Technology*. Berlin: Springer-Verlag, 1990.

[14] P. Haasen and R. Cahn, *Phase Transformations in Materials*. Weinheim: Wiley-VCH, 2001.

[15] F. R. N. Nabarro, “Diffusion and precipitation in alloys,” in *Strength of Solids*, vol. 1096, London: The Physical Society, pp. 237–250, 1947.

[16] K. Ralls, T. Courtney, and J. Wulff, *Introduction to Materials Science and Engineering*. New York: Wiley, 1976.

[17] C. Pein, “Microstructural Modelling of Diffusional Creep in Polycrystals,” PhD Thesis, Graz University of Technology, 2010.

[18] A. G. Guy, *Essentials of Materials Science*. New York: McGraw-Hill, 1976.

[19] M. E. Glicksman, *Diffusion in solids: Field Theory, Solid-State Principles, and Applications*. New York: Wiley-Interscience, 1999.

[20] V. Berdichevsky, P. Hazzledine, and B. Shoykhet, “Micromechanics of Diffusional Creep,” *Int. J. Eng. Sci.*, vol. 35, no. 10, pp. 1003–1032, 1997.

[21] H. Riedel, *Fracture at High Temperatures*. Berlin: Springer-Verlag, 1987.

[22] J. Čadek, *Creep in Metallic Materials*. New York: Elsevier Ltd, 1988.

[23] H. J. Frost and M. F. Ashby, *Deformation-Mechanism Maps: The Plasticity and Creep of Metals and Ceramics*. Oxford: Pergamon Press, 1982.

[24] V. Yamakov, D. Wolf, S. R. Phillpot, and H. Gleiter, “Grain-boundary diffusion creep in nanocrystalline palladium by molecular-dynamics simulation,” *Acta Mater.*, vol. 50, pp. 61–73, 2002.

[25] R. Korla and A. H. Chokshi, “A Constitutive Equation for Grain Boundary Sliding: An Experimental Approach,” *Metall. Mater. Trans. A*, vol. 45, no. 2, pp. 698–708, 2013.

[26] Y. F. Yin and R. G. Faulkner, “Physical and elastic behavior of creep-resistant steels,” in *Creep Resistant Steels*, F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 217–240, 2008.

[27] J. Shi and M. A. Zikry, “Grain size, grain boundary sliding, and grain boundary interaction effects on nanocrystalline behavior,” *Mater. Sci. Eng. A*, vol. 520, no. 1–2, pp. 121–133, 2009.

[28] J. Wadsworth, O. A. Ruano, and O. D. Sherby, “Denuded zones, diffusional creep, and grain boundary sliding,” *Metall. Mater. Trans. A*, vol. 33, pp. 219–229, 2002.

[29] J. B. Bilde-Sørensen and D. A. Smith, “Scripta Metall. Mater,” vol. 30, pp. 383–386, 1994.

[30] J. Fiala, J. Novotný, and J. Čadek., “Coble and Harper-Dorn creep in iron at homologous temperatures of 0.40- 0.54,” *Mater. Sci. Eng.*, vol. 60, no. 3, pp. 195–206, 1983.

[31] M. . Ashby, “On interface-reaction control of Nabarro-Herring creep and sintering,” *Scr. Metall.*, vol. 3, no. 11, pp. 837–842, 1969.

[32] D. A. Smith, V. Vitek, and R. C. Pond, “Computer simulation of symmetrical high angle boundaries in aluminium,” *Acta Metall.*, vol. 25, no. 5, pp. 475–483, 1977.

[33] H. Gleiter, “Grain boundaries as point defect sources or sinks-Diffusional creep,” *Acta Metall.*, vol. 27, no. 2, pp. 187–192, 1979.

[34] E. Arzt, M. F. Ashby, and R. A. Verrall, “Interface controlled diffusional creep,” *Acta Metall.*, vol. 31, no. 12, pp. 1977–1989, 1983.

[35] K. A. Peterson, I. Dutta, and M. W. Chen, “Diffusionally accommodated interfacial sliding in metalsilicon systems,” *Acta Mater.*, vol. 51, pp. 2831–2846, 2003.

[36] K. Kuchařová, S. J. Zhu, and J. Čadek, “Creep in copper dispersion strengthened with alumina particles,” *Mater. Sci. Eng. A*, vol. 348, pp. 170–179, 2003.

[37] B. Cai, Q. . Kong, L. Lu, and K. Lu, “Low temperature creep of nanocrystalline pure copper,” *Mater. Sci. Eng. A*, vol. 286, pp. 188–192, 2000.

[38] C. Herring, “Diffusional Viscosity of a Polycrystalline Solid,” *J. Appl. Phys.*, vol. 21, pp. 437–445, 1950.

[39] F. R. N. Nabarro, *Deformation of Crystals by the Motion of Single Ions*. London: The Physical Society, 1948.

[40] R. L. Coble, “A Model for Boundary Diffusion Controlled Creep in Polycrystalline Materials,” *J.*

Appl. Phys., vol. 34, no. 6, pp. 1679–1682, 1963.

[41] M. Basirat, T. Shrestha, G. P. Potirniche, I. Charit, and K. Rink, "A study of the creep behavior of modified 9Cr-1Mo steel using continuum-damage modeling," *Int. J. Plast.*, vol. 37, pp. 95–107, 2012.

[42] D. M. Owen and T. G. Langdon, "Low stress creep behaviour: an examination of Nabarro-Herring and Harper-Dorn creep," *Mater. Sci. Eng.*, vol. 216, pp. 20–29, 1996.

[43] B. Burton and G. . Reynolds, "In defense of diffusional creep," *Mater. Sci. Eng.*, vol. 191, pp. 135–141, 1995.

[44] F. A. Mohamed and T. J. Ginter, "On the nature and origin of Harper-Dorn creep," *Acta Metall.*, vol. 30, no. 10, pp. 1869 – 1881, 1982.

[45] D. Y. Yevgeny and G. Lasko, "Field of Stresses in an Isotropic Plane with Circular Inclusion under Tensile Stress," *Engineering*, vol. 04, no. 9, pp. 583–589, 2012.

[46] D. E. Eshelby, "Definition of the Stress Field, Which Was Creating by Elliptical Inclusion," in *Proceedings of the Royal Society A*, 1957, vol. 241, no. 1226, p. 376, 1957.

[47] M. E. Kassner and T. A. Hayes, "Creep cavitation in metals," *Int. J. Plast.*, vol. 19, no. 10, pp. 1715–1748, 2003.

[48] A. Aghajani, "Evolution of Microstructure during Long term Creep of a Tempered Martensite Ferritic Steel," PhD Thesis, Ruhr-University Bochum, 2009.

[49] Z. Hu, "Heat Resistant Steels, Microstructural Evolution and Life Assessment in Power Plants," in *Thermal Power Plants*, M. Rasul, Ed. Rijeka: InTech, pp. 195–226, 2012.

[50] M. Saber, "Experimental and finite element studies of creep and creep crack growth in P91 and P92 weldments," PhD Thesis, University of Notting-ham, 2011.

[51] W. Yan, W. Wang, Y.-Y. Shan, and K. Yang, "Microstructural stability of 9–12%Cr ferrite/martensite heat-resistant steels," *Front. Mater. Sci.*, vol. 7, no. 1, pp. 1–27, 2013.

[52] B. Sonderegger, "Charakterisierung der Substruktur in modernen Kraftwerkswerkstoffen mittels der EBSD Methode," PhD Thesis, Graz University of Technology, 2005.

[53] R. W. Cahn and P. Haasen, *Physical Metallurgy*, 4th ed. New York: Elsevier Ltd, 1983.

[54] V. Foldyna, A. Jakobova, A. Gemperle, and R. Riman, "Creep and Fracture of Engineering Materials and Structures," *Inst. Mater.*, pp. 507–516, 1990.

[55] J. Orr and D. Burton, *Materials for Advanced Power Engineering, Part I*. Netherlands: Kluver Academic, 1994.

[56] Z. Kubon and V. Foldyna, "Steel Research," vol. 66, p. 389, 1995.

[57] E. Cerri, E. Evangelista, S. Spigarelli, and P. Bianchi, "Evolution of microstructure in a modified 9Cr–1Mo steel during short term creep," *Mater. Sci. Eng. A*, vol. 245, no. 2, pp. 285–292, May 1998.

[58] D. Jara, "9-12 % Cr heat resistant steels : alloy design , TEM characterisation of microstructure evolution and creep response at 650 ° C," PhD Thesis, Rubr University Bochum, 2011.

[59] F. Abe, "Strengthening mechanisms in steel for creep and creep rupture," in *Creep Resistant Steels*, F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 279–304, 2008.

[60] K. Kimura, H. Kushima, F. Abe, K. Suzuki, S. Kumai, and A. Satoh, “Microstructural change and degradation behaviour of 9Cr-1Mo-V-Nb steel in the long term,” in *Proceedings of the 5th International Charles Parsons Turbine Conference*, pp. 590–602, 2000.

[61] K. Sawada, M. Taneike, K. Kimura, and F. Abe, “In situ observation of recovery of lath structure in 9% chromium creep resistant steel,” *Mater. Sci. Technol.*, vol. 19, no. 6, pp. 739–742, 2003.

[62] H. Armaki, R. Chen, K. Maruyama, and M. Igarashi, “Creep Behavior and Degradation of Subgrain Structures Pinned by Nanoscale Precipitates in Strength-Enhanced 5 to 12 Pct Cr Ferritic Steels,” *Metall. Mater. Trans. A*, vol. 42, no. 10, pp. 3084–3094, 2011.

[63] C. G. Panait, A. Zielińska-Lipiec, and T. Koziel, “Evolution of dislocation density, size of subgrains and MX-type precipitatesin a P91 steel during creep and during thermal ageing at 600°C for more than 100,000 h,” *Mater. Sci. Eng. A*, vol. 527, no. 16–17, pp. 4062–4069, 2010.

[64] G. Eggeler, “The effect of long-term creep on particle coarsening in tempered martensite ferritic steels,” *Acta Metall.*, vol. 37, no. 12, pp. 3225–3234, 1989.

[65] J. Hald, “Microstructure and long-term creep properties of 9-12% Cr steels,” *Int. J. Press. Vessel. Pip.*, vol. 85, no. 1–2, pp. 30–37, 2008.

[66] W. Blum, “Mechanisms of creep deformation in steel,” in *Creep Resistant Steels*, 1st ed., F. Abe, T. Kern, and R. Viswanathan, Eds. Cambridge England: Woodhead Publishing Limited, pp. 365–400, 2008.

[67] K. R. Jian Hua Yao, H. G. Elder, and M. Grant, “Theory and Stimulation of Ostwald Rippening,” *Phys. Rev. B*, vol. 47, no. 21, pp. 14110–14125, 1993.

[68] I. M. Lifshitz and V. V. Slyozov, “The Kinetics of Precipitation From Supersaturated Solid Solutions,” *J. Phys.*, vol. 91, pp. 35–50, 1961.

[69] K. C. Russell, “Precipitate Coarsening and Grain Growth in Steels,” *Iron Steel Soc.*, no. 1, pp. 437–456, 2003.

[70] A. Aghajani, C. Somsen, and G. Eggeler, “On the effect of long-term creep on the microstructure of a 12% chromium tempered martensite ferritic steel,” *Acta Mater.*, vol. 57, no. 17, pp. 5093– 5106, 2009.

[71] H. K. Danielsen and J. Hald, “On the nucleation and dissolution process of Z-phase Cr(V, Nb)N in martensitic 12%Cr steels,” *Mater. Sci. Eng. A*, vol. 505, no. 1–2, pp. 169–177, 2009.

[72] K. Sawada, M. Takeda, K. Maruyama, R. Ishii, M. Yamada, Y. Nagae, and R. Komine, “Effect of W on recovery of lath structure during creep of high chromium martensitic steels,” *Mater. Sci. Eng. A*, vol. 267, no. 1, pp. 19–25, 1999.

[73] S. D. Yadav, B. Sonderegger, B. Sartory, C. Sommitsch, and C. Poletti, “Characterisation and quantification of cavities in 9Cr martensitic steel for power plants,” *Mater. Sci. Technol.*, vol. 31, no. 5, pp. 554–564, 2015.

[74] P. M. Anderson and P. G. Shewmon, “Stress redistribution and cavity nucleation near a diffusively growing grain boundary cavity,” *Mech. Mater*, vol. 32, no. 3, pp. 175–191, 2000.

[75] J. R. Mayeur, D. L. McDowell, and R. W. Neu, “Crystal plasticity simulations of fretting of Ti-6Al-4V in partial slip regime considering effects of texture,” *Comput. Mater. Sci.*, vol. 41, pp. 356–365, 2008.

[76] M. Zhang, J. Zhang, and D. L. McDowell, “Microstructure-based crystal plasticity modeling of cyclic deformation of Ti–6Al–4V.,” *Int. J. Plast.*, vol. 23, pp. 1328–1348, 2007.

[77] D. McLean, “Resistance to hot deformation,” *Trans. Metall. Soc. AIME*, vol. 242, pp. 1193–1203, 1968.

[78] A. Strang, *Microstructural Development and Stability in High Chromium Ferritic Power Plant Steels*. London: Maney Materials Science, 1996.

[79] D. Prokoshkina, V. A. Esin, G. Wilde, and S. V. Divinski, “Grain boundary width, energy and self-diffusion in nickel: Effect of material purity,” *Acta Mater.*, vol. 61, no. 14, pp. 5188–5197, 2013.

Appendix A: MATLAB Code

A.1 A_pure_metal.m

```
%% A pure metal cube under stress, no precipitates/voids, no stress
considerations on diffusion

clc
clear all
% close all

%% Constants
m = 100;
% Size of matrix
dt = 60;
% Time step size
tmax = 10000-1;
% Max time step
t = 1;
% While loop start point
G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
% Grain Width at start
G_Sy = G_S;
% Grain Length at start
dx = G_Sx/(m-1);
% size of each element x dir
dy = G_Sy/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin

%% Vacancy Concentrations Constants

A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
sigma = 30*10^6;
% Applied Stress
Omega = 1.18*10^-29;
% Volume of atom

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix == 0
% Grain boundary == 1
% Phase boundary == 1
```

```
% Voids will be == 2
% Precipitate == 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary

%% Matrix formation
% Concentration Matrix
% Gices the local concentration of vacancies in each element
NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
NVC_plus = NVC_Eq*((exp((sigma*Omega)/(k*T_max))+1)/2);
% Initial molar concentration of vacancies increased due to tensile stress
C = NVC_Eq.*ones(m,m);
% Concentration matrix before stress
C_new = C;
% Replication for later

% Boundary Conditions due to stress
C(1,1:m) = NVC_plus;
% Top boundary
C(m,1:m) = NVC_plus;
% Bottom boundary

% Corner boundary conditions
% Average out corners
C(1,1)=(NVC_Eq+NVC_plus)/2;
C(m,1)=(NVC_Eq+NVC_plus)/2;
C(1,m)=(NVC_Eq+NVC_plus)/2;
C(m,m)=(NVC_Eq+NVC_plus)/2;

%% Blank Matrices
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
% Blank

%% While loop
while t <= tmax

sigma_thin = sigma/(G_Sx/G_S);
% Increase in stress due to thinning

NVC_plus = NVC_Eq*((exp((sigma_thin*Omega)/(k*T_max))+1)/2);
% Molar concentration changes due to stress increase
```

```
C(1,1:m) = NVC_plus;
% Top boundary
C(m,1:m) = NVC_plus;
% Bottom boundary
C(1,1)=(NVC_Eq+NVC_plus)/2;
C(m,1)=(NVC_Eq+NVC_plus)/2;
C(1,m)=(NVC_Eq+NVC_plus)/2;
C(m,m)=(NVC_Eq+NVC_plus)/2;

% Calculating vacancy diffusion
for i = 1:m
% For loop in y direction
for j = 1:m
% For loop in x direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matrix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in y direction
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1));
% flow in x direction
    C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end
P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_matrix2(2:m-1,1) = 0;
% Make sides zero
P_matrix2(2:m-1,m) = 0;
% Make sides zero
for ii = 1:m
% Defining change in concentration between boundary and closest element,
allows for addition of inclusions later
    for jj = 1:m
        if ii <= m-1
        if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
            Face_1m(ii,jj) = (C(ii,jj));
            Face_2m(ii+1,jj) = (C(ii+1,jj));
        end
        end
        if ii >= 2
        if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
             Face_1m(ii,jj) = (C(ii,jj));
             Face_2m(ii-1,jj) = (C(ii-1,jj));
        end
        end
    end
end

Flux_x = D*((sum(C(:,1))+sum(C(:,m)))-(sum(C(:,2))+sum(C(:,m-1))))/(dx)...
% Flux to side boundaries for thinning, Ficks 1st law J = D dC/dx
            *(dt)...
% Time step
            *(Av_no);
% Changes concentration to number of atoms
```

```
Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Flux to top and bottom boundaries for elongation, Ficks 1st law J = D
dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration and Avogadro's number for number of vacancies

strain_y =  sum(V_Flux) ...
% Strain at each time step. Total vacancy flux divided by number of points
            *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
            *(m^2);
% Account for size of matrix

strain_x =  Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
            *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
            *(m^2);
% Account for size of matrix


Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction

% change to grain dimensions dx and dy with thinning and elongation
G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_Sx1;

% Add next time step
t=t+1;
end

%% Plot vacancy concentration surface

% x =0:dx:G_Sx;
%Range of x(0,2) and specifying the grid points
% y =0:dy:G_Sy;
%Range of y(0,2) and specifying the grid points
% surf(x,y,C)
% % % shading interp
% title({['2-D Diffusion with \itD = ',num2str(D)];['time (\itt) =
',num2str(t)]},'FontSize',20)
% xlabel('Grain Dimension (x) \rightarrow','FontSize',20)
% ylabel('{\leftarrow} Grain Dimension (y)','FontSize',20)
% zlabel('Molar Vacancy Concentration \rightarrow','FontSize',20)
% set(gca,'FontSize',15)
% colormap(jet)

%% Plot strain vs theoretical strain
```

```
hold on
t = 1:t;
% Time vector
NH = dt*((D*sigma*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation
plot(t,Strain_y,'b',t,NH,'k','LineWidth',1.4);
% Plot theory vs grain
set(gca,'FontSize',15)
% Axes font size
xlabel('Time (Minutes)','FontSize',20)
ylabel('Strain','FontSize',20)
legendd = legend('Model Damage','N-H Theory');
set(legendd,'FontSize',18)
% Legend font size
grid on
```

A.2 B_Stress_Fields.m

```
%% Stress fields of precipitates with no coarsening

clc
clear all
% close all

%% Constants
m = 100;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0.3;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix
E_2 = 430;
% Young's Modulus Precipitate

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix = 0
% Grain boundary = 1
% Phase boundary = 1
% Precipitate = 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary

%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor

    %% Imbedded FORloop to Calculate Stress Matrix
% Current consideration for 1, 2 or 3 precipitates on each boundary, later
% plans to write up for any amount of precipitates
N = 1;
% Number of precipitates on top and bottom boundary


% If there is only one precipitate at the boundary
if N == 1
```

```
R_P_Percent = 10;
% percentage of surface area occupied by precipitates
r_p = round(sqrt((m*m*(R_P_Percent/100))/pi/2)/0.5)*0.5;
% Radius of Precipitate
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = m/2;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = m/2;
        y_pos2 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        %%
        if r >= r_p && r2 >= r_p
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2);
%             Sigma_xy(y,x) = ((1-k_y-k_x)*r_p^2*(x-x_pos)*(y-
y_pos))/r^4*(3-(2*(3*r_p^2+4*(y-y_pos)^2))/r^2+(12*r_p^2*(y-y_pos)^2)/r^4)-
(2*k_x*r_p^2*(x-x_pos)*(y-y_pos))/r^4 ...
%                               +((1-k_y-k_x)*r_p^2*(x-x_pos2)*(y-
y_pos2))/r2^4*(3-(2*(3*r_p^2+4*(y-y_pos2)^2))/r2^2+(12*r_p^2*(y-
y_pos2)^2)/r2^4)-(2*k_x*r_p^2*(x-x_pos2)*(y-y_pos2))/r2^4;
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
%             Sigma_xy(y,x) = 0;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end
% Defining phase boundary with grain boundary value
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
```

```
            P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
            P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
            P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
            P_matrix(ii,jj-1) = 1;
        end
    end
end

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary


% Resloving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1
        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
```

```
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end

% If there is two precipitates at the boundary

elseif N == 2
r_p = round(sqrt((m*m*0.1)/pi/4)/0.5)*0.5;
% Radius of Precipitate
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = round((m-r_p*4)/3+r_p);
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = round((m-r_p*4)/3+r_p);
        y_pos2 = 0;
        x_pos3 = m-round((m-r_p*4)/3+r_p);
        y_pos3 = m;
        x_pos4 = m-round((m-r_p*4)/3+r_p);
        y_pos4 = 0;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p.^2.*(y-y_pos4).^4))/r4.^6;
        %%
        if r >= r_p && r2 >= r_p && r3 >= r_p && r4 >= r_p
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r3^2)*(1+(3*r_p^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r4^2)*(1+(3*r_p^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
```

```
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r3^2)*(3-
(3*r_p^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r4^2)*(3-
(3*r_p^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
%            Sigma_xy(y,x) = ((1-k_y-k_x)*r_p^2*(x-x_pos)*(y-
y_pos))/r^4*(3-(2*(3*r_p^2+4*(y-y_pos)^2))/r^2+(12*r_p^2*(y-y_pos)^2)/r^4)-
(2*k_x*r_p^2*(x-x_pos)*(y-y_pos))/r^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos2)*(y-
y_pos2))/r2^4*(3-(2*(3*r_p^2+4*(y-y_pos2)^2))/r2^2+(12*r_p^2*(y-
y_pos2)^2)/r2^4)-(2*k_x*r_p^2*(x-x_pos2)*(y-y_pos2))/r2^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos3)*(y-
y_pos3))/r3^4*(3-(2*(3*r_p^2+4*(y-y_pos3)^2))/r3^2+(12*r_p^2*(y-
y_pos3)^2)/r3^4)-(2*k_x*r_p^2*(x-x_pos3)*(y-y_pos3))/r3^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos4)*(y-
y_pos4))/r4^4*(3-(2*(3*r_p^2+4*(y-y_pos4)^2))/r4^2+(12*r_p^2*(y-
y_pos4)^2)/r4^4)-(2*k_x*r_p^2*(x-x_pos4)*(y-y_pos4))/r4^4;
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
%             Sigma_xy(y,x) = 0;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end

for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
```

```
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary                                                    % Bottom
boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
```

```
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end

% If there is three precipitates at the boundary

elseif N == 3
r_p = round(sqrt((m*m*0.1)/pi/6)/0.5)*0.5;
% Radius of Precipitate
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = round((m-r_p*6)/4+r_p);
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = round((m-r_p*6)/4+r_p);
        y_pos2 = 0;
        x_pos3 = round(((m-r_p*6)/4+r_p)*2+r_p);
        y_pos3 = m;
        x_pos4 = round(((m-r_p*6)/4+r_p)*2+r_p);
        y_pos4 = 0;
        x_pos5 = m-round((m-r_p*6)/4+r_p);
        y_pos5 = m;
        x_pos6 = m-round((m-r_p*6)/4+r_p);
        y_pos6 = 0;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        r5 = sqrt((x-x_pos5).^2 +(y-y_pos5).^2);
        r6 = sqrt((x-x_pos6).^2 +(y-y_pos6).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p.^2+2.*(y-y_pos4).^2))/r4.^4;
        F5 = (8*(y-y_pos5).^2.*(3.*r_p.^2+2.*(y-y_pos5).^2))/r5.^4;
        F6 = (8*(y-y_pos6).^2.*(3.*r_p.^2+2.*(y-y_pos6).^2))/r6.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p.^2.*(y-y_pos4).^4))/r4.^6;
        G5 = (24.*(r_p.^2.*(y-y_pos5).^4))/r5.^6;
        G6 = (24.*(r_p.^2.*(y-y_pos6).^4))/r6.^6;
        %%
        if r >= r_p && r2 >= r_p && r3 >= r_p && r4 >= r_p ...
% Conditions for Stress Field Outside Precipitate
                && r5 >= r_p && r6 >= r_p
            Sigma_y(y,x) =  1 ...
```

```
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r3^2)*(1+(3*r_p^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r4^2)*(1+(3*r_p^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2)...;
                            +((1-
k_y+k_x)*r_p^2)/(2*r5^2)*(1+(3*r_p^2+10*(y-y_pos5)^2)/r5^2-F5+G5)-
(k_x*r_p^2)/r5^2*(1-(2*(y-y_pos5)^2)/r5^2)...;
                            +((1-
k_y+k_x)*r_p^2)/(2*r6^2)*(1+(3*r_p^2+10*(y-y_pos6)^2)/r6^2-F6+G6)-
(k_x*r_p^2)/r6^2*(1-(2*(y-y_pos6)^2)/r6^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r3^2)*(3-
(3*r_p^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r4^2)*(3-
(3*r_p^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r5^2)*(3-
(3*r_p^2+18*(y-y_pos5)^2)/r5^2+F5-G5)+(k_x*r_p^2)/r5^2*(1-(2*(y-
y_pos5)^2)/r5^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r6^2)*(3-
(3*r_p^2+18*(y-y_pos6)^2)/r6^2+F6-G6)+(k_x*r_p^2)/r6^2*(1-(2*(y-
y_pos6)^2)/r6^2);
%             Sigma_xy(y,x) = ((1-k_y-k_x)*r_p^2*(x-x_pos)*(y-
y_pos))/r^4*(3-(2*(3*r_p^2+4*(y-y_pos)^2))/r^2+(12*r_p^2*(y-y_pos)^2)/r^4)-
(2*k_x*r_p^2*(x-x_pos)*(y-y_pos))/r^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos2)*(y-
y_pos2))/r2^4*(3-(2*(3*r_p^2+4*(y-y_pos2)^2))/r2^2+(12*r_p^2*(y-
y_pos2)^2)/r2^4)-(2*k_x*r_p^2*(x-x_pos2)*(y-y_pos2))/r2^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos3)*(y-
y_pos3))/r3^4*(3-(2*(3*r_p^2+4*(y-y_pos3)^2))/r3^2+(12*r_p^2*(y-
y_pos3)^2)/r3^4)-(2*k_x*r_p^2*(x-x_pos3)*(y-y_pos3))/r3^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos4)*(y-
y_pos4))/r4^4*(3-(2*(3*r_p^2+4*(y-y_pos4)^2))/r4^2+(12*r_p^2*(y-
y_pos4)^2)/r4^4)-(2*k_x*r_p^2*(x-x_pos4)*(y-y_pos4))/r4^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos5)*(y-
y_pos5))/r5^4*(3-(2*(3*r_p^2+4*(y-y_pos5)^2))/r5^2+(12*r_p^2*(y-
y_pos5)^2)/r5^4)-(2*k_x*r_p^2*(x-x_pos5)*(y-y_pos5))/r5^4 ...
%                             +((1-k_y-k_x)*r_p^2*(x-x_pos6)*(y-
y_pos6))/r6^4*(3-(2*(3*r_p^2+4*(y-y_pos6)^2))/r6^2+(12*r_p^2*(y-
y_pos6)^2)/r6^4)-(2*k_x*r_p^2*(x-x_pos6)*(y-y_pos6))/r6^4;
        else            % Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
%             Sigma_xy(y,x) = 0;
            P_matrix(y,x) = 3;
        end
    end
end
```

```
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary
% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/3
% Lower third
```

```
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = -(iii-y_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/3
% Lower third
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = (iii-y_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end

else
    display('too many precipitates for current works')
end

% subplot(1,2,1)
surf(Sigma_y)
colormap(jet)
view([0 0])
% subplot(1,2,2)
```

```
% surf((Sigma_x))
xlabel('X-axis','fontsize',20)
ylabel('Y-axis','fontsize',20)
zlabel('Normalized Stress','fontsize',20)
colorbar
set(gca,'FontSize',15,'XTick','','YTick','')
set(colorbar,'fontsize',20)
```

A.3 C_Stress_Fields_Voids.m

```
%% Stress fields of void with no growth

clc
clear all
% close all

%% Constants
m = 100;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix
E_2 = 0;
% Young's Modulus Precipitate

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix = 0
% Grain boundary = 1
% Phase boundary = 1
% Void = 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary

%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor

    %% Imbedded FORloop to Calculate Stress Matrix
% Current consideration for 1, 2 or 3 precipitates on each boundary, later
% plans to write up for any amount of precipitates
N = 2;
% Number of precipitates on top and bottom boundary


r_p = 10;
% Radius of void
```

```
for y = 1:m        % Y vector
    for x = 1:m    % X vector
        %%
        x_pos = 1;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = 1;
        y_pos2 = 1;
        x_pos3 = m;
        y_pos3 = m;
        x_pos4 = m;
        y_pos4 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p.^2.*(y-y_pos4).^4))/r4.^6;
        %%
        if r >= r_p && r2 >= r_p && r3 >= r_p && r4 >= r_p
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r3^2)*(1+(3*r_p^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r4^2)*(1+(3*r_p^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);

            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r3^2)*(3-
(3*r_p^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r4^2)*(3-
(3*r_p^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
        else
% No Stress Field Inside Void
            Sigma_y(y,x) = NaN;
            Sigma_x(y,x) = NaN;
```

```
%             Sigma_xy(y,x) = 0;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3
        end
    end
end

% Damage correction for stress fields
corr = (1+(100-sum(Sigma_y(1,ceil(r_p)+1:m-
ceil(r_p))))/sum(Sigma_y(1,ceil(r_p)+1:m-ceil(r_p))));
Sigma_y = Sigma_y*corr;
Sigma_x = Sigma_x*corr;

% Stress field plotting
% subplot(1,2,1)
surf(Sigma_y)
colormap(jet)
view([2])
colorbar
set(colorbar,'fontsize',20)
set(gca,'xtick','','ytick','')
% subplot(1,2,2)
% surf((Sigma_x))
% colormap(jet)
% view([2])
% xlabel('X-axis','fontsize',20)
% ylabel('Y-axis','fontsize',20)
% zlabel('Normalized Stress','fontsize',20)
% colorbar
% set(colorbar,'fontsize',15)
```

A.4 D_Stress_Fields_Coarsening.m

```
%% Stress fields around a precipitate subject to coarsening

clc
clear all
% close all

%% Constants - Stress fields
m = 150;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0.3;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix (iron)
E_2 = 430;
% Young's Modulus Precipitate (silicon carbide)

%% Constants- Coarsening
N1 = 3;
tmax = 100;
r_0 = 4*10^-9;
% original precipitate radius
K_p = 5*10^-18;
% Constant
Q_p = 286.8;
% Activation energy
R_g =8.31451;
% Gas Constant
T = 600+293;
% Thermodynamic Temperature

%% Constants

G_S = 1*10^-6;
% Grain Size
dx = G_S/(m-1);
% size of each element x dir
dy = G_S/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix == 0
% Grain boundary == 1
% Phase boundary == 1
% Voids will be == 2
% Precipitate == 3
```

```
P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop
%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor


dt = 60;
% Starting points

Sigma_x = zeros(m,m);
% Blank
Sigma_y = zeros(m,m);
% Blank

%% Imbedded FORloop to Calculate Stress Matrix
% Current consideration for 1, 2 or 3 precipitates on each boundary, later
% plans to write up for any amount of precipitates
% a = 0;
N = N1;
% Number of precipitates on top and bottom boundary
t = 1;

while t <= tmax
    % While loop for coarsening

% If there is only one precipitate at the boundary
if N == 1
a = 1;
r_p = (sqrt((m*m*0.1)/pi/2));
% Radius of Precipitate
Vol = 2*pi*(r_p)^2;
% Total volume of precipitates
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6*m;
% Growing precipitate
r_p2 = ((Vol/pi)- r_p1^2)^(1/2);
% Shrinking precipitate

if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end
```

```
for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = m/2;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = m/2;
        y_pos2 = 0;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        %%
        if r >= r_p1 && r2 >= r_p2
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end

% If there is two precipitates at the boundary

elseif N == 2
a = 2;
r_p = sqrt((m*m*0.1)/pi/4);
% Radius of Precipitate
Vol = 4*pi*(r_p)^2;
% Total volume of precipitates
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p2 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
```

```
r_p3 = ((Vol/2/pi)- r_p1^2)^(1/2);
r_p4 = ((Vol/2/pi)- r_p2^2)^(1/2);

if r_p1 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi/2);
    r_p3 = 0;
end
if r_p2 >= sqrt(((m*m*0.1))/pi/2)
    r_p2 = sqrt(((m*m*0.1))/pi/2);
    r_p4 = 0;
end

if r_p3 == 0 && r_p4 == 0
    r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p2 = ((Vol/pi)- r_p1^2)^(1/2);
if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end
end


for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = (m-r_p*4)/3+r_p;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = (m-r_p*4)/3+r_p;
        y_pos2 = 1;
        x_pos3 = m-((m-r_p*4)/3+r_p);
        y_pos3 = m;
        x_pos4 = m-((m-r_p*4)/3+r_p);
        y_pos4 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
```

```
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end

% If there is three precipitates at the boundary

elseif N == 3
r_p = sqrt((m*m*0.1)/pi/6);
% Radius of Precipitate
Vol = 6*pi*(r_p)^2;
% Total volume of precipitates
r_p4 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
r_p2 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p1 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);
r_p6 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p5 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);

if r_p4 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p4 = sqrt(((m*m*0.1))/pi/2);
    r_p2 = 0;
    r_p6 = 0;
end
if r_p3 >= sqrt(((m*m*0.1))/pi/2)
    r_p3 = sqrt(((m*m*0.1))/pi/2);
    r_p1 = 0;
    r_p5 = 0;
```

```
end
if r_p1 == 0 && r_p2 == 0 && r_p5 == 0 && r_p6 == 0
    r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p4 = ((Vol/pi)- r_p3^2)^(1/2);
if r_p3 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p3 = sqrt(((m*m*0.1))/pi);
    r_p4 = 0;
end
end
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = (m-r_p*6)/4+r_p;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = (m-r_p*6)/4+r_p;
        y_pos2 = 1;
        x_pos3 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos3 = m;
        x_pos4 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos4 = 1;
        x_pos5 = m-((m-r_p*6)/4+r_p);
        y_pos5 = m;
        x_pos6 = m-((m-r_p*6)/4+r_p);
        y_pos6 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        r5 = sqrt((x-x_pos5).^2 +(y-y_pos5).^2);
        r6 = sqrt((x-x_pos6).^2 +(y-y_pos6).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        F5 = (8*(y-y_pos5).^2.*(3.*r_p5.^2+2.*(y-y_pos5).^2))/r5.^4;
        F6 = (8*(y-y_pos6).^2.*(3.*r_p6.^2+2.*(y-y_pos6).^2))/r6.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;
        G5 = (24.*(r_p5.^2.*(y-y_pos5).^4))/r5.^6;
        G6 = (24.*(r_p6.^2.*(y-y_pos6).^4))/r6.^6;
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4 ...
% Conditions for Stress Field Outside Precipitate
                && r5 >= r_p5 && r6 >= r_p6
            Sigma_y(y,x) =  1 ...
                           +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                           +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
```

```
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2)...;
                            +((1-
k_y+k_x)*r_p5^2)/(2*r5^2)*(1+(3*r_p5^2+10*(y-y_pos5)^2)/r5^2-F5+G5)-
(k_x*r_p5^2)/r5^2*(1-(2*(y-y_pos5)^2)/r5^2)...;
                            +((1-
k_y+k_x)*r_p6^2)/(2*r6^2)*(1+(3*r_p6^2+10*(y-y_pos6)^2)/r6^2-F6+G6)-
(k_x*r_p6^2)/r6^2*(1-(2*(y-y_pos6)^2)/r6^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2)...
                            +((1-k_y+k_x)*r_p5^2)/(2*r5^2)*(3-
(3*r_p5^2+18*(y-y_pos5)^2)/r5^2+F5-G5)+(k_x*r_p5^2)/r5^2*(1-(2*(y-
y_pos5)^2)/r5^2)...
                            +((1-k_y+k_x)*r_p6^2)/(2*r6^2)*(3-
(3*r_p6^2+18*(y-y_pos6)^2)/r6^2+F6-G6)+(k_x*r_p6^2)/r6^2*(1-(2*(y-
y_pos6)^2)/r6^2);
        else            % Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            P_matrix(y,x) = 3;
        end
    end
end

else
    display('too many precipitates for current works')
end


t=t+1;
end

% Stress field plot
% subplot(1,2,1)
surf((Sigma_y))
colormap(jet)
view([0 90])
% subplot(1,2,2)
% surf((Sigma_x))
% xlabel('X-axis','fontsize',20)
% ylabel('Y-axis','fontsize',20)
% zlabel('Normalized Stress','fontsize',20)
% colorbar
set(gca,'FontSize',15,'XTick','','YTick','')
% set(colorbar,'fontsize',20)
```

A.5 E_Creep_Precipitates.m

```
%% Precipitate with no caorsening


clc
clear all
% close all

%% Constants - Stress fields
m = 100;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0.3;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix (iron)
E_2 = 430;
% Young's Modulus Precipitate (carbide)

N1 = 3;
% Number of precipitates

%% Constants

tmax = 10000-1;
% Max time step
dt = 60;
% Time step size
G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
% Grain Width at start
G_Sy = G_S;
% Grain Length at start
dx = G_Sx/(m-1);
% size of each element x dir
dy = G_Sy/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin
E_mol = 5.4*10^3;
%% Vacancy Concentrations Constants

A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
sigma1 = 30*10^6;
% Applied Stress `
```

```
Omega = 1.18*10^-29;
% Volume of atom

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix == 0
% Grain boundary == 1
% Phase boundary == 1
% Voids will be == 2
% Precipittate == 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop
%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor

% Starting points
NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
C = NVC_Eq.*ones(m,m);
% Blank
C_new = C;
% Blank
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
Flux = zeros(1,m);
% Blank
V_Flux = zeros(1,m);
% Blank
Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
% Blank
Sigma_x = zeros(m,m);
% Blank
Sigma_y = zeros(m,m);
% Blank
%% Imbedded FORloop to Calculate Stress Matrix
```

```
% Current consideration for 1, 2 or 3 precipitates on each boundary,
N = N1;
% Number of precipitates on top and bottom boundary
t = 1;

while t <= tmax
    % While loop for coarsening

P_matrix = P_matrix_2;
% reset matrix back to original
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
% If there is no precipitate at the boundary
if N == 0

Stress_GBy = zeros(m,m);
% Uniform blank field
Stress_GBx = zeros(m,m);
Stress_GBy(1,1:m) = 1;
% Stress at top face
Stress_GBy(m,1:m) = 1;
% Stress at bottom face
a = 0;


% If there is only one precipitate at the boundary
elseif N == 1
a = 1;
r_p = (sqrt((m*m*0.1)/pi/2));
% Radius of Precipitate

% Total volume of precipitates

for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = m/2;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = m/2;
        y_pos2 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        %%
        if r >= r_p && r2 >= r_p
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
```

```
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end
% Defining phase boundary with grain boundary value
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
if a == 1

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
```

```
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1
        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end
% If there is two precipitates at the boundary

elseif N == 2
a = 2;
r_p = sqrt((m*m*0.1)/pi/4);
% Radius of Precipitate

for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = round((m-r_p*4)/3+r_p);
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = round((m-r_p*4)/3+r_p);
        y_pos2 = 1;
        x_pos3 = round(m-((m-r_p*4)/3+r_p));
        y_pos3 = m;
        x_pos4 = round(m-((m-r_p*4)/3+r_p));
        y_pos4 = 1;
        %%
```

```
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p.^2.*(y-y_pos4).^4))/r4.^6;
        %%
        if r >= r_p && r2 >= r_p && r3 >= r_p && r4 >= r_p
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r3^2)*(1+(3*r_p^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p^2)/(2*r4^2)*(1+(3*r_p^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r3^2)*(3-
(3*r_p^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p^2)/(2*r4^2)*(3-
(3*r_p^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end

for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
```

```
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
if a == 2

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/2
% Right half
```

```
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end

% If there is three precipitates at the boundary

elseif N == 3
a = 3;
r_p = sqrt((m*m*0.1)/pi/6);
% Radius of Precipitate


for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = (m-r_p*6)/4+r_p;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = (m-r_p*6)/4+r_p;
        y_pos2 = 1;
        x_pos3 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos3 = m;
        x_pos4 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos4 = 1;
        x_pos5 = m-((m-r_p*6)/4+r_p);
        y_pos5 = m;
        x_pos6 = m-((m-r_p*6)/4+r_p);
        y_pos6 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
```

```
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        r5 = sqrt((x-x_pos5).^2 +(y-y_pos5).^2);
        r6 = sqrt((x-x_pos6).^2 +(y-y_pos6).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p.^2+2.*(y-y_pos4).^2))/r4.^4;
        F5 = (8*(y-y_pos5).^2.*(3.*r_p.^2+2.*(y-y_pos5).^2))/r5.^4;
        F6 = (8*(y-y_pos6).^2.*(3.*r_p.^2+2.*(y-y_pos6).^2))/r6.^4;
        G = (24.*(r_p.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p.^2.*(y-y_pos4).^4))/r4.^6;
        G5 = (24.*(r_p.^2.*(y-y_pos5).^4))/r5.^6;
        G6 = (24.*(r_p.^2.*(y-y_pos6).^4))/r6.^6;
        %%
        if r >= r_p && r2 >= r_p && r3 >= r_p && r4 >= r_p ...
% Conditions for Stress Field Outside Precipitate
               && r5 >= r_p && r6 >= r_p
            Sigma_y(y,x) =  1 ...
                           +((1-k_y+k_x)*r_p^2)/(2*r^2)*(1+(3*r_p^2+10*(y-
y_pos)^2)/r^2-F+G)-(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                           +((1-
k_y+k_x)*r_p^2)/(2*r2^2)*(1+(3*r_p^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                           +((1-
k_y+k_x)*r_p^2)/(2*r3^2)*(1+(3*r_p^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                           +((1-
k_y+k_x)*r_p^2)/(2*r4^2)*(1+(3*r_p^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2)...;
                           +((1-
k_y+k_x)*r_p^2)/(2*r5^2)*(1+(3*r_p^2+10*(y-y_pos5)^2)/r5^2-F5+G5)-
(k_x*r_p^2)/r5^2*(1-(2*(y-y_pos5)^2)/r5^2)...;
                           +((1-
k_y+k_x)*r_p^2)/(2*r6^2)*(1+(3*r_p^2+10*(y-y_pos6)^2)/r6^2-F6+G6)-
(k_x*r_p^2)/r6^2*(1-(2*(y-y_pos6)^2)/r6^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p^2)/(2*r^2)*(3-(3*r_p^2+18*(y-
y_pos)^2)/r^2+F-G)+(k_x*r_p^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                           +((1-k_y+k_x)*r_p^2)/(2*r2^2)*(3-
(3*r_p^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                           +((1-k_y+k_x)*r_p^2)/(2*r3^2)*(3-
(3*r_p^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                           +((1-k_y+k_x)*r_p^2)/(2*r4^2)*(3-
(3*r_p^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2)...
                           +((1-k_y+k_x)*r_p^2)/(2*r5^2)*(3-
(3*r_p^2+18*(y-y_pos5)^2)/r5^2+F5-G5)+(k_x*r_p^2)/r5^2*(1-(2*(y-
y_pos5)^2)/r5^2)...
                           +((1-k_y+k_x)*r_p^2)/(2*r6^2)*(3-
(3*r_p^2+18*(y-y_pos6)^2)/r6^2+F6-G6)+(k_x*r_p^2)/r6^2*(1-(2*(y-
y_pos6)^2)/r6^2);
        else            % Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
```

```
            P_matrix(y,x) = 3;
        end
    end
end
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
else
    display('too many precipitates for current works')
end
if a == 3

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary
```

```
% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/3
% Lower third
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = -(iii-y_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/3
% Lower third
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = (iii-y_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end
```

```
if a == 1 ||a == 2||a == 3
matrix_scarcey(matrix_scarcey == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
% matrix_scarcex(matrix_scarcex == 0) = 1;
Stress_GBy = matrix_scarcey.*Stress_GB_unmodifiedy;
% Normal stress around phase boundary X Original stress state present at
boundaries
Stress_GBx = matrix_scarcex.*Stress_GB_unmodifiedx;

end


% sigma2 = sigma1/(1-((K_p*10^-2*t^2.5)/(G_S^2)));
sigma_thin = sigma1/(G_Sx/G_S);
% Increase in stress due to thinning

NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
NVC_plus =
NVC_Eq*((exp(((sigma_thin.*(Stress_GBy+Stress_GBx))*Omega)/(k*T_max))+1)/2)
; % Molar concentration of vacancies increased due to tensile stress
(adding x and y fields)

%% Matrix formation
% Boundary Conditions
for xx = 1:m
    for yy = 1:m
if P_matrix(xx,yy) == 1
C(xx,yy) =  NVC_plus(xx,yy);
end
if P_matrix(xx,yy) == 3
    C(xx,yy) = NVC_Eq;
end
if P_matrix(xx,yy) == 0
C(xx,yy) = C_new(xx,yy);
end
    end
end

% Corner boundary conditions
% Average out corners
C(1,1)=(NVC_Eq+NVC_plus(1,1))/2;
C(m,1)=(NVC_Eq+NVC_plus(m,1))/2;
C(1,m)=(NVC_Eq+NVC_plus(1,m))/2;
C(m,m)=(NVC_Eq+NVC_plus(m,m))/2;


for i = 1:m
% For loop in x direction
for j = 1:m
% For loop in y direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matrix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in x direction
```

```
                    D*dt/(dy^2)*(Sigma_y(i+1,j)-2*Sigma_y(i,j)+Sigma_y(i-
1,j))*(Av_no/E_mol)*Omega+...
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1))+...;
% flow in y direction
                    D*dt/(dx^2)*(Sigma_x(i,j+1)-2*Sigma_x(i,j)+Sigma_x(i,j-
1))*(Av_no/E_mol)*Omega;
    C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end
P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_matrix2(2:m-1,1) = 0;
% Make sides zero
P_matrix2(2:m-1,m) = 0;
% Make sides zero
for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
% Layer above precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii+1,jj) = (C(ii+1,jj));
            end
        end
        if jj >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj-1)== 0
% Layer left of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj-1) = (C(ii,jj-1));
            end
        end
        if ii >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
% Layer below precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii-1,jj) = (C(ii-1,jj));
            end
        end
        if jj <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj+1)== 0
% Layer right of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj+1) = (C(ii,jj+1));
            end
        end
    end
end

Flux_x = D*((sum(C(:,1))+sum(C(:,m)))-(sum(C(:,2))+sum(C(:,m-1))))/(dx)...
% Flux to side boundaries for thinning, Ficks 1st law J = D dC/dx
            *(dt)...
% Time step
            *(Av_no);
```

```
Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Ficks 1st law J = D dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration times Avogadro's number for number of vacancies



strain_y =    sum(V_Flux) ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix


strain_x =    Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix

Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction

% change to dx and dy with thinning and elongation
G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_Sx1;

t=t+1;
end

% x =0:dx:G_Sx;
%Range of x(0,2) and specifying the grid points
% y =0:dy:G_Sy;
%Range of y(0,2) and specifying the grid points
% surf(x,y,C)
% xlim([0 G_Sx])
% ylim([0 G_Sy])
% % shading interp
% title({['2-D Diffusion with \itD = ',num2str(D)]},'FontSize',15)
% xlabel('Grain Dimension (x) \rightarrow','FontSize',10)
% ylabel('{\leftarrow} Grain Dimension (y)','FontSize',10)
% zlabel('Transport property profile (u) \rightarrow')
% view([0 90])

%% Plot strain vs theoretical strain

hold on
```

```
t = 1:t;
% Time vector
NH = dt*((D*sigma1*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation
plot(t,Strain_y,'r','LineWidth',1.4);%,t,NH,'k','LineWidth',1.4);  %
,'LineWidth',1.4);%                              % Plot theory vs grain
% title({['STRAIN Vs TIME']},'FontSize',20)
xlabel('Time (min)','FontSize',20)
ylabel('Strain','FontSize',20)
set(gca,'FontSize',15)
legendd = legend('Model Damage','N-H Theory');
set(legendd,'FontSize',18);
grid on
```

A.6 F_Creep_Precipitates_Coarsening.m

```
%% Precipitate with coarsening

clc
clear all
% close all

%% Constants - Stress fields
m = 100;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0.3;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix (iron)
E_2 = 430;
% Young's Modulus Precipitate (carbide)

%% Constants- Coarsening
N1 = 1;
% Number of precipitates
K_p = 5*10^-28;
% Constant

%% Constants

tmax = 10000-1;
% Max time step
dt = 60;
% Time step size
G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
% Grain Width at start
G_Sy = G_S;
% Grain Length at start
dx = G_Sx/(m-1);
% size of each element x dir
dy = G_Sy/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin
E_mol = 5.4*10^3;
%% Vacancy Concentrations Constants

A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
```

```
sigma1 = 30*10^6;
% Applied Stress `
Omega = 1.18*10^-29;
% Volume of atom

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix == 0
% Grain boundary == 1
% Phase boundary == 1
% Voids will be == 2
% Precipitate == 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop
%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor

% Starting points
NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
C = NVC_Eq.*ones(m,m);
% Blank
C_new = C;
% Blank
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
Flux = zeros(1,m);
% Blank
V_Flux = zeros(1,m);
% Blank
Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
% Blank
Sigma_x = zeros(m,m);
% Blank
```

```
Sigma_y = zeros(m,m);
% Blank
%% Imbedded FORloop to Calculate Stress Matrix
% Current consideration for 1, 2 or 3 precipitates on each boundary,
N = N1;
% Number of precipitates on top and bottom boundary
t = 1;

while t <= tmax
    % While loop for coarsening

P_matrix = P_matrix_2;
% reset matrix back to original
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
% If there is no precipitate at the boundary
if N == 0

Stress_GBy = zeros(m,m);
% Uniform blank field
Stress_GBx = zeros(m,m);
Stress_GBy(1,1:m) = 1;
% Stress at top face
Stress_GBy(m,1:m) = 1;
% Stress at bottom face
a = 0;


% If there is only one precipitate at the boundary
elseif N == 1
a = 1;
r_p = 10;%(sqrt((m*m*0.1)/pi/2));
% Radius of Precipitate
Vol = 2*pi*(r_p)^2;
% Total volume of precipitates
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p2 = abs(((Vol/pi)- r_p1^2)^(1/2));
% Shrinking precipitate

if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end

for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = m/2;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = m/2;
        y_pos2 = 1;
        %%
```

```
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        %%
        if r >= r_p1 && r2 >= r_p2
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                           +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                           +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                           +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end
% Defining phase boundary with grain boundary value
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
if a == 1
```

```
matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1
        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end
% If there is two precipitates at the boundary

elseif N == 2
a = 2;
r_p = sqrt((m*m*0.1)/pi/4);
% Radius of Precipitate
Vol = 4*pi*(r_p)^2;
% Total volume of precipitates
```

```
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p2 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
r_p3 = ((Vol/2/pi)- r_p1^2)^(1/2);
r_p4 = ((Vol/2/pi)- r_p2^2)^(1/2);

if r_p1 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi/2);
    r_p3 = 0;
end
if r_p2 >= sqrt(((m*m*0.1))/pi/2)
    r_p2 = sqrt(((m*m*0.1))/pi/2);
    r_p4 = 0;
end

if r_p3 == 0 && r_p4 == 0
    r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p2 = ((Vol/pi)- r_p1^2)^(1/2);
if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end
end


for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = round((m-r_p*4)/3+r_p)+0.0001;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = round((m-r_p*4)/3+r_p)+0.0001;
        y_pos2 = 1;
        x_pos3 = round(m-((m-r_p*4)/3+r_p))+0.0001;
        y_pos3 = m;
        x_pos4 = round(m-((m-r_p*4)/3+r_p))+0.0001;
        y_pos4 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;
```

```
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
    end
end

for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
```

```
if a == 2
matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
```

```
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end

% If there is three precipitates at the boundary

elseif N == 3
a = 3;
r_p = sqrt((m*m*0.1)/pi/6);
% Radius of Precipitate
Vol = 6*pi*(r_p)^2;
% Total volume of precipitates
r_p4 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
r_p2 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p1 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);
r_p6 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p5 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);

if r_p4 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p4 = sqrt(((m*m*0.1))/pi/2);
    r_p2 = 0;
    r_p6 = 0;
end
if r_p3 >= sqrt(((m*m*0.1))/pi/2)
    r_p3 = sqrt(((m*m*0.1))/pi/2);
    r_p1 = 0;
    r_p5 = 0;
end
if r_p1 == 0 && r_p2 == 0 && r_p5 == 0 && r_p6 == 0
    r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p4 = ((Vol/pi)- r_p3^2)^(1/2);
if r_p3 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p3 = sqrt(((m*m*0.1))/pi);
    r_p4 = 0;
end
end
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_pos = (m-r_p*6)/4+r_p;
% x position of precipitate
        y_pos = m;
% y position of precipitate
```

```
        x_pos2 = (m-r_p*6)/4+r_p;
        y_pos2 = 1;
        x_pos3 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos3 = m;
        x_pos4 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos4 = 1;
        x_pos5 = m-((m-r_p*6)/4+r_p);
        y_pos5 = m;
        x_pos6 = m-((m-r_p*6)/4+r_p);
        y_pos6 = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        r5 = sqrt((x-x_pos5).^2 +(y-y_pos5).^2);
        r6 = sqrt((x-x_pos6).^2 +(y-y_pos6).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        F5 = (8*(y-y_pos5).^2.*(3.*r_p5.^2+2.*(y-y_pos5).^2))/r5.^4;
        F6 = (8*(y-y_pos6).^2.*(3.*r_p6.^2+2.*(y-y_pos6).^2))/r6.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;
        G5 = (24.*(r_p5.^2.*(y-y_pos5).^4))/r5.^6;
        G6 = (24.*(r_p6.^2.*(y-y_pos6).^4))/r6.^6;
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4 ...
% Conditions for Stress Field Outside Precipitate
                && r5 >= r_p5 && r6 >= r_p6
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2)...;
                            +((1-
k_y+k_x)*r_p5^2)/(2*r5^2)*(1+(3*r_p5^2+10*(y-y_pos5)^2)/r5^2-F5+G5)-
(k_x*r_p5^2)/r5^2*(1-(2*(y-y_pos5)^2)/r5^2)...;
                            +((1-
k_y+k_x)*r_p6^2)/(2*r6^2)*(1+(3*r_p6^2+10*(y-y_pos6)^2)/r6^2-F6+G6)-
(k_x*r_p6^2)/r6^2*(1-(2*(y-y_pos6)^2)/r6^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
```

```
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2)...
                            +((1-k_y+k_x)*r_p5^2)/(2*r5^2)*(3-
(3*r_p5^2+18*(y-y_pos5)^2)/r5^2+F5-G5)+(k_x*r_p5^2)/r5^2*(1-(2*(y-
y_pos5)^2)/r5^2)...
                            +((1-k_y+k_x)*r_p6^2)/(2*r6^2)*(3-
(3*r_p6^2+18*(y-y_pos6)^2)/r6^2+F6-G6)+(k_x*r_p6^2)/r6^2*(1-(2*(y-
y_pos6)^2)/r6^2);
        else            % Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            P_matrix(y,x) = 3;
        end
    end
end
for ii = 2:m-1
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 2:m-1
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
    end
end
else
    display('too many precipitates for current works')
end
if a == 3

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
```

```
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(m,1:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_y;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_x;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/3
% Lower third
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = -(iii-y_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/3
% Lower third
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj >= m/3 && jjj <= m*2/3
% Middle third
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
```

```
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end
            if jjj >= m*2/3
% Upper third
            matrix_scarcey (iii,jjj) = (iii-y_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end

if a == 1 ||a == 2||a == 3
matrix_scarcey(matrix_scarcey == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
% matrix_scarcex(matrix_scarcex == 0) = 1;
Stress_GBy = matrix_scarcey.*Stress_GB_unmodifiedy;
% Normal stress around phase boundary X Original stress state present at
boundaries
Stress_GBx = matrix_scarcex.*Stress_GB_unmodifiedx;

end



% sigma2 = sigma1/(1-((K_p*10^-2*t^2.5)/(G_S^2)));
sigma_thin = sigma1/(G_Sx/G_S);
% Increase in stress due to thinning

NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
NVC_plus =
NVC_Eq*((exp(((sigma_thin.*(Stress_GBy+Stress_GBx))*Omega)/(k*T_max))+1)/2)
; % Molar concentration of vacancies increased due to tensile stress
(adding x and y fields)
%% Matrix formation
% Boundary Conditions
for xx = 1:m
    for yy = 1:m
if P_matrix(xx,yy) == 1
C(xx,yy) =  NVC_plus(xx,yy);
end
if P_matrix(xx,yy) == 3
    C(xx,yy) = NVC_Eq;
end
if P_matrix(xx,yy) == 0
C(xx,yy) = C_new(xx,yy);
end
    end
end

% Corner boundary conditions
% Average out corners
```

```
C(1,1)=(NVC_Eq+NVC_plus(1,1))/2;
C(m,1)=(NVC_Eq+NVC_plus(m,1))/2;
C(1,m)=(NVC_Eq+NVC_plus(1,m))/2;
C(m,m)=(NVC_Eq+NVC_plus(m,m))/2;


for i = 1:m
% For loop in x direction
for j = 1:m
% For loop in y direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in x direction
                    D*dt/(dy^2)*(Sigma_y(i+1,j)-2*Sigma_y(i,j)+Sigma_y(i-
1,j))*(Av_no/E_mol)*Omega+...
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1))+...;
% flow in y direction
                    D*dt/(dx^2)*(Sigma_x(i,j+1)-2*Sigma_x(i,j)+Sigma_x(i,j-
1))*(Av_no/E_mol)*Omega;
    C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end
P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_matrix2(2:m-1,1) = 0;
% Make sides zero
P_matrix2(2:m-1,m) = 0;
% Make sides zero
for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
% Layer above precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii+1,jj) = (C(ii+1,jj));
            end
        end
        if jj >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj-1)== 0
% Layer left of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj-1) = (C(ii,jj-1));
            end
        end
        if ii >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
% Layer below precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii-1,jj) = (C(ii-1,jj));
            end
        end
        if jj <= m-1
```

```
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj+1)== 0
% Layer right of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj+1) = (C(ii,jj+1));
            end
        end
    end
end

Flux_x = D*((sum(C(:,1))+sum(C(:,m)))-(sum(C(:,2))+sum(C(:,m-1))))/(dx)...
% Flux to side boundaries for thinning, Ficks 1st law J = D dC/dx
            *(dt)...
% Time step
            *(Av_no);

Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Ficks 1st law J = D dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration times Avogadro's number for number of vacancies



strain_y =    sum(V_Flux) ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Divided by length of grain to give strain


% Halved as it does not match theory.................................
strain_x =    Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix

Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction

% change to dx and dy with thinning and elongation
G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_Sx1;

t=t+1;
end

%% Plot strain vs theoretical strain
```

```
hold on
t = 1:t;
% Time vector
NH = dt*((D*sigma1*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation

xlabel('Time (min)','FontSize',20)
ylabel('Strain','FontSize',20)
plot(t,Strain_y,'b','LineWidth',1.4);%,t,NH,'k','LineWidth',1.4);
% Plot theory vs model
legend('Model Damage','N-H Theory')
set(gca,'FontSize',15)
legendd = legend('Model Damage','N-H Theory');
set(legendd,'FontSize',18);
grid on
```

A.7 G_Creep_Void.m

```
%% Void with no growth

clc
clear all
% close all

%% Constants

m = 100;
% Size of matrix
v_1v = 0.3;
% Poisson's Ratio Matrix
v_2v = 0.3;
% Poisson's Ratio Precipitate
E_1v = 200;
% Young's Modulus Matrix
E_2v = 0;
% Young's Modulus Precipitate
tmax = 10000-1;
dt = 60;

%% Constants

G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
% Grain Width at start
G_Sy = G_S;
% Grain Length at start
% size of each element x dir
dx = G_Sx/(m-1);
% size of each element x dir
dy = G_Sy/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin
E_mol = 5.4*10^3;
%% Vacancy Concentrations Constants

A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
sigma = 30*10^6;
% Applied Stress
Omega = 1.18*10^-29;
% Volume of atom
sigma1 = sigma;
D_gb = D/2;
% Grain boundary diffusion coefficient
```

```
d_gb = 0.5*10^-9;
% Grain boundary thickness
r_p1v = 10;
% Original void size

NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
C = NVC_Eq.*ones(m,m);
% Blank
C_new = C;
% Blank

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix = 0
% Grain boundary = 1
% Phase boundary = 1
% Void = 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop

%% Pre Calculations
% factors account for adding two different field properties together
k_yv =   (E_2v*((3-v_2v)*E_1v+(5+v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% ky factor
k_xv =   (E_2v*((3*v_2v-1)*E_1v+(1-3*v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% kx factor

    %% Imbedded FORloop to Calculate Stress Matrix

Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
% Blank
t = 1;

while t <= tmax

P_matrix = P_matrix_2;
% reset matrix back to original
Face_1m = zeros(m,m);
% Blank
```

```
Face_2m = zeros(m,m);
% Blank
% Precipitate at each corner

r_pv = r_p1v;
if r_pv == 0
    Sigma_yv = ones(m,m);
    Sigma_xv = zeros(m,m);
    Stress_GBy = zeros(m,m);
% Uniform blank field
    Stress_GBx = zeros(m,m);
    Stress_GBy(1,1:m) = 1;
% Stress at top face
    Stress_GBy(m,1:m) = 1;
else
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_posv = 1;
% x position of void
        y_posv = m;
% y position of void
        x_pos2v = 1;
        y_pos2v = 1;
        x_pos3v = m;
        y_pos3v = m;
        x_pos4v = m;
        y_pos4v = 1;
        %%
        rv = sqrt((x-x_posv).^2 +(y-y_posv).^2);
% Distance from centre of void to coordinate(x,y)
        r2v = sqrt((x-x_pos2v).^2 +(y-y_pos2v).^2);
        r3v = sqrt((x-x_pos3v).^2 +(y-y_pos3v).^2);
        r4v = sqrt((x-x_pos4v).^2 +(y-y_pos4v).^2);
        %%
        Fv = (8*(y-y_posv).^2.*(3.*r_pv.^2+2.*(y-y_posv).^2))/rv.^4;
% Recurring Constant
        F2v = (8*(y-y_pos2v).^2.*(3.*r_pv.^2+2.*(y-y_pos2v).^2))/r2v.^4;
        F3v = (8*(y-y_pos3v).^2.*(3.*r_pv.^2+2.*(y-y_pos3v).^2))/r3v.^4;
        F4v = (8*(y-y_pos4v).^2.*(3.*r_pv.^2+2.*(y-y_pos4v).^2))/r4v.^4;
        Gv = (24.*(r_pv.^2.*(y-y_posv).^4))/rv.^6;
% Recurring Constant
        G2v = (24.*(r_pv.^2.*(y-y_pos2v).^4))/r2v.^6;
        G3v = (24.*(r_pv.^2.*(y-y_pos3v).^4))/r3v.^6;
        G4v = (24.*(r_pv.^2.*(y-y_pos4v).^4))/r4v.^6;
        %%
        if rv >= r_pv && r2v >= r_pv && r3v >= r_pv && r4v >= r_pv
% Conditions for Stress Field Outside Void
            Sigma_yv(y,x) =  1 ...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*rv^2)*(1+(3*r_pv^2+10*(y-y_posv)^2)/rv^2-Fv+Gv)-
(k_xv*r_pv^2)/rv^2*(1-(2*(y-y_posv)^2)/rv^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(1+(3*r_pv^2+10*(y-y_pos2v)^2)/r2v^2-F2v+G2v)-
(k_xv*r_pv^2)/r2v^2*(1-(2*(y-y_pos2v)^2)/r2v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(1+(3*r_pv^2+10*(y-y_pos3v)^2)/r3v^2-F3v+G3v)-
(k_xv*r_pv^2)/r3v^2*(1-(2*(y-y_pos3v)^2)/r3v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(1+(3*r_pv^2+10*(y-y_pos4v)^2)/r4v^2-F4v+G4v)-
(k_xv*r_pv^2)/r4v^2*(1-(2*(y-y_pos4v)^2)/r4v^2);
```

```
            Sigma_xv(y,x) =  ((1-k_yv+k_xv)*r_pv^2)/(2*rv^2)*(3-
(3*r_pv^2+18*(y-y_posv)^2)/rv^2+Fv-Gv)+(k_xv*r_pv^2)/rv^2*(1-(2*(y-
y_posv)^2)/rv^2)...
                              +((1-k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(3-
(3*r_pv^2+18*(y-y_pos2v)^2)/r2v^2+F2v-G2v)+(k_xv*r_pv^2)/r2v^2*(1-(2*(y-
y_pos2v)^2)/r2v^2)...
                              +((1-k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(3-
(3*r_pv^2+18*(y-y_pos3v)^2)/r3v^2+F3v-G3v)+(k_xv*r_pv^2)/r3v^2*(1-(2*(y-
y_pos3v)^2)/r3v^2)...
                              +((1-k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(3-
(3*r_pv^2+18*(y-y_pos4v)^2)/r4v^2+F4v-G4v)+(k_xv*r_pv^2)/r4v^2*(1-(2*(y-
y_pos4v)^2)/r4v^2);
%             Sigma_xy(y,x) = ((1-k_y-k_x)*r_p^2*(x-x_pos)*(y-
y_pos))/r^4*(3-(2*(3*r_p^2+4*(y-y_pos)^2))/r^2+(12*r_p^2*(y-y_pos)^2)/r^4)-
(2*k_x*r_p^2*(x-x_pos)*(y-y_pos))/r^4 ...
%                               +((1-k_y-k_x)*r_p^2*(x-x_pos2)*(y-
y_pos2))/r2^4*(3-(2*(3*r_p^2+4*(y-y_pos2)^2))/r2^2+(12*r_p^2*(y-
y_pos2)^2)/r2^4)-(2*k_x*r_p^2*(x-x_pos2)*(y-y_pos2))/r2^4 ...
%                               +((1-k_y-k_x)*r_p^2*(x-x_pos3)*(y-
y_pos3))/r3^4*(3-(2*(3*r_p^2+4*(y-y_pos3)^2))/r3^2+(12*r_p^2*(y-
y_pos3)^2)/r3^4)-(2*k_x*r_p^2*(x-x_pos3)*(y-y_pos3))/r3^4 ...
%                               +((1-k_y-k_x)*r_p^2*(x-x_pos4)*(y-
y_pos4))/r4^4*(3-(2*(3*r_p^2+4*(y-y_pos4)^2))/r4^2+(12*r_p^2*(y-
y_pos4)^2)/r4^4)-(2*k_x*r_p^2*(x-x_pos4)*(y-y_pos4))/r4^4;
        else
% Stress Field Inside Void
            Sigma_yv(y,x) = NaN;
            Sigma_xv(y,x) = NaN;
%             Sigma_xy(y,x) = 0;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3
        end
    end
end

for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        end
        if ii > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        end
        if jj < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        end
```

```
        if jj > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
        end
    end
end

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of void phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Change scarce matrix to get rid of void phase in phase matrix
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(1,m:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_yv;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_xv;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary



% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 1:m
    for jjj = 1:m

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = -(iii-y_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2)));
            end
```

```
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end

matrix_scarcey(matrix_scarcey == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
% matrix_scarcex(matrix_scarcex == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
Stress_GBy = matrix_scarcey.*Stress_GB_unmodifiedy;
% Normal stress around phase boundary X Original stress state present at
boundaries
Stress_GBx = matrix_scarcex.*Stress_GB_unmodifiedx;
% Normal stress around phase boundary X Original stress state present at
boundaries
end
sigma_thin = sigma/(G_Sx/G_S);
% Increase in stress due to thinning
w = (G_S-(G_S-2*(r_pv*10^-8)))/(G_S);
sigma1 = sigma_thin/(1-w);
sigma2 = sigma_thin*(1+(100-sum(Sigma_yv(1,ceil(r_p1v)+1:m-
ceil(r_p1v))))/sum(Sigma_yv(1,ceil(r_p1v)+1:m-ceil(r_p1v))));
NVC_plus =
NVC_Eq*((exp(((sigma2.*(Stress_GBy+Stress_GBx))*Omega)/(k*T_max))+1)/2); %
Molar concentration of vacancies increased due to tensile stress

%% Matrix formation
% Boundary Conditions
for xx = 1:m
    for yy = 1:m
if P_matrix(xx,yy) == 1
```

```
C(xx,yy) =  NVC_plus(xx,yy);
end
if P_matrix(xx,yy) == 3
    C(xx,yy) = NVC_Eq;
end
if P_matrix(xx,yy) == 0
C(xx,yy) = C_new(xx,yy);
end
    end
end

% Corner boundary conditions
% Average out corners
C(1,1)=(NVC_Eq+NVC_plus(1,1))/2;
C(m,1)=(NVC_Eq+NVC_plus(m,1))/2;
C(1,m)=(NVC_Eq+NVC_plus(1,m))/2;
C(m,m)=(NVC_Eq+NVC_plus(m,m))/2;


for i = 1:m
% For loop in x direction
for j = 1:m
% For loop in y direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in x direction
                    D*dt/(dy^2)*(Sigma_yv(i+1,j)-
2*Sigma_yv(i,j)+Sigma_yv(i-1,j))*(Av_no/E_mol)*Omega+...
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1))+...
% flow in y direction
                    D*dt/(dx^2)*(Sigma_xv(i,j+1)-
2*Sigma_xv(i,j)+Sigma_xv(i,j-1))*(Av_no/E_mol)*Omega;
     C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end

P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_vector1 = find(P_matrix(1:m,1)==1);
P_vector2 = find(P_matrix(1:m,m)==1);
P_matrix2(P_vector1(2:end-1),1) = 0;
% Make sides zero
P_matrix2(P_vector2(2:end-1),m) = 0;
% Make sides zero

for ii = 1:m
% Defining faces for flux
    for jj = 1:m
        if ii <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
% Layer above void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii+1,jj) = (C(ii+1,jj));
            end
```

```
        end
        if jj >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj-1)== 0
% Layer left of void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj-1) = (C(ii,jj-1));
            end
        end
        if ii >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
% Layer below void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii-1,jj) = (C(ii-1,jj));
            end
        end
        if jj <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj+1)== 0
% Layer right of void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj+1) = (C(ii,jj+1));
            end
        end
    end
end

Flux_x = D*((sum(C(P_vector1,1))+sum(C(P_vector2,m)))-
(sum(C(P_vector1,2))+sum(C(P_vector2,m-1))))/(dx)...  % Flux to side
boundaries for thinning, Ficks 1st law J = D dC/dx
            *(dt)...
% Time step
            *(Av_no);

Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Ficks 1st law J = D dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration times Avogadro's number for number of vacancies



strain_y =    sum(V_Flux) ...
% Strain at each time step.
              *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix to give strain

strain_x =    Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix

Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction
```

```
G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_Sx1;

t=t+1;
end

hold on
t = 1:t;
% Time vector
NH = dt*((D*sigma*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation
plot(t,Strain_y,'r','LineWidth',1.4);%,t,NH,'k','LineWidth',1.4); %
,'LineWidth',1.4);%
xlabel('Time (min)','FontSize',20)
ylabel('Strain','FontSize',20)
legend('Model Damage','N-H Theory')
grid on
```

A.8 H_Creep_Void_Growth.m

```
%% Void with growth


clc
clear all
% close all

%% Constants

m = 100;
% Size of matrix
v_1v = 0.3;
% Poisson's Ratio Matrix
v_2v = 0.3;
% Poisson's Ratio Precipitate
E_1v = 200;
% Young's Modulus Matrix
E_2v = 0;
% Young's Modulus Precipitate
tmax = 100000-1;%
dt = 60;

%% Constants

G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
% Grain Width at start
G_Sy = G_S;
% Grain Length at start
% size of each element x dir
dx = G_Sx/(m-1);
% size of each element x dir
dy = G_Sy/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin
E_mol = 5.4*10^3;
%% Vacancy Concentrations Constants

A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
sigma = 30*10^6;
% Applied Stress
Omega = 1.18*10^-29;
% Volume of atom
sigma1 = sigma;
R = 8.3144598;
```

```
D_gb = D/(exp(-125500/(R*T_max)));
% Grain boundary diffusion coefficient
d_gb = 0.5*10^-9;
% Grain boundary thickness
r_p1v = 2;
% Original void size

NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
C = NVC_Eq.*ones(m,m);
% Blank
C_new = C;
% Blank

%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix = 0
% Grain boundary = 1
% Phase boundary = 1
% Void = 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop

%% Pre Calculations
% factors account for adding two different field properties together
k_yv =   (E_2v*((3-v_2v)*E_1v+(5+v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% ky factor
k_xv =   (E_2v*((3*v_2v-1)*E_1v+(1-3*v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% kx factor

    %% Imbedded FORloop to Calculate Stress Matrix

Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
% Blank
t = 1;

while t <= tmax

P_matrix = P_matrix_2;
% reset matrix back to original
```

```
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
% 1 void at each corner

r_pv = r_p1v+(D_gb*d_gb*(sigma1)*Omega*dt)/(2*k*T_max*G_Sx*r_p1v)*10^11;%
r_p1v = r_pv;
if r_pv == 0
    Sigma_yv = ones(m,m);
    Sigma_xv = zeros(m,m);
    Stress_GBy = zeros(m,m);
% Uniform blank field
    Stress_GBx = zeros(m,m);
    Stress_GBy(1,1:m) = 1;
% Stress at top face
    Stress_GBy(m,1:m) = 1;
else
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
        x_posv = 1;
% x position of void
        y_posv = m;
% y position of void
        x_pos2v = 1;
        y_pos2v = 1;
        x_pos3v = m;
        y_pos3v = m;
        x_pos4v = m;
        y_pos4v = 1;
        %%
        rv = sqrt((x-x_posv).^2 +(y-y_posv).^2);
% Distance from centre of void to coordinate(x,y)
        r2v = sqrt((x-x_pos2v).^2 +(y-y_pos2v).^2);
        r3v = sqrt((x-x_pos3v).^2 +(y-y_pos3v).^2);
        r4v = sqrt((x-x_pos4v).^2 +(y-y_pos4v).^2);
        %%
        Fv = (8*(y-y_posv).^2.*(3.*r_pv.^2+2.*(y-y_posv).^2))/rv.^4;
% Recurring Constant
        F2v = (8*(y-y_pos2v).^2.*(3.*r_pv.^2+2.*(y-y_pos2v).^2))/r2v.^4;
        F3v = (8*(y-y_pos3v).^2.*(3.*r_pv.^2+2.*(y-y_pos3v).^2))/r3v.^4;
        F4v = (8*(y-y_pos4v).^2.*(3.*r_pv.^2+2.*(y-y_pos4v).^2))/r4v.^4;
        Gv = (24.*(r_pv.^2.*(y-y_posv).^4))/rv.^6;
% Recurring Constant
        G2v = (24.*(r_pv.^2.*(y-y_pos2v).^4))/r2v.^6;
        G3v = (24.*(r_pv.^2.*(y-y_pos3v).^4))/r3v.^6;
        G4v = (24.*(r_pv.^2.*(y-y_pos4v).^4))/r4v.^6;
        %%
        if rv >= r_pv && r2v >= r_pv && r3v >= r_pv && r4v >= r_pv
% Conditions for Stress Field Outside Void
            Sigma_yv(y,x) =  1 ...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*rv^2)*(1+(3*r_pv^2+10*(y-y_posv)^2)/rv^2-Fv+Gv)-
(k_xv*r_pv^2)/rv^2*(1-(2*(y-y_posv)^2)/rv^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(1+(3*r_pv^2+10*(y-y_pos2v)^2)/r2v^2-F2v+G2v)-
(k_xv*r_pv^2)/r2v^2*(1-(2*(y-y_pos2v)^2)/r2v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(1+(3*r_pv^2+10*(y-y_pos3v)^2)/r3v^2-F3v+G3v)-
(k_xv*r_pv^2)/r3v^2*(1-(2*(y-y_pos3v)^2)/r3v^2)...
```

```
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(1+(3*r_pv^2+10*(y-y_pos4v)^2)/r4v^2-F4v+G4v)-
(k_xv*r_pv^2)/r4v^2*(1-(2*(y-y_pos4v)^2)/r4v^2);
            Sigma_xv(y,x) =  ((1-k_yv+k_xv)*r_pv^2)/(2*rv^2)*(3-
(3*r_pv^2+18*(y-y_posv)^2)/rv^2+Fv-Gv)+(k_xv*r_pv^2)/rv^2*(1-(2*(y-
y_posv)^2)/rv^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(3-
(3*r_pv^2+18*(y-y_pos2v)^2)/r2v^2+F2v-G2v)+(k_xv*r_pv^2)/r2v^2*(1-(2*(y-
y_pos2v)^2)/r2v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(3-
(3*r_pv^2+18*(y-y_pos3v)^2)/r3v^2+F3v-G3v)+(k_xv*r_pv^2)/r3v^2*(1-(2*(y-
y_pos3v)^2)/r3v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(3-
(3*r_pv^2+18*(y-y_pos4v)^2)/r4v^2+F4v-G4v)+(k_xv*r_pv^2)/r4v^2*(1-(2*(y-
y_pos4v)^2)/r4v^2);
%            Sigma_xy(y,x) = ((1-k_y-k_x)*r_p^2*(x-x_pos)*(y-
y_pos))/r^4*(3-(2*(3*r_p^2+4*(y-y_pos)^2))/r^2+(12*r_p^2*(y-y_pos)^2)/r^4)-
(2*k_x*r_p^2*(x-x_pos)*(y-y_pos))/r^4 ...
%                              +((1-k_y-k_x)*r_p^2*(x-x_pos2)*(y-
y_pos2))/r2^4*(3-(2*(3*r_p^2+4*(y-y_pos2)^2))/r2^2+(12*r_p^2*(y-
y_pos2)^2)/r2^4)-(2*k_x*r_p^2*(x-x_pos2)*(y-y_pos2))/r2^4 ...
%                              +((1-k_y-k_x)*r_p^2*(x-x_pos3)*(y-
y_pos3))/r3^4*(3-(2*(3*r_p^2+4*(y-y_pos3)^2))/r3^2+(12*r_p^2*(y-
y_pos3)^2)/r3^4)-(2*k_x*r_p^2*(x-x_pos3)*(y-y_pos3))/r3^4 ...
%                              +((1-k_y-k_x)*r_p^2*(x-x_pos4)*(y-
y_pos4))/r4^4*(3-(2*(3*r_p^2+4*(y-y_pos4)^2))/r4^2+(12*r_p^2*(y-
y_pos4)^2)/r4^4)-(2*k_x*r_p^2*(x-x_pos4)*(y-y_pos4))/r4^4;
        else
% Stress Field Inside Void
            Sigma_yv(y,x) = NaN;
            Sigma_xv(y,x) = NaN;
%             Sigma_xy(y,x) = 0;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3
        end
    end
end

for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        end
        if ii > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        end
        if jj < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
```

```
           P_matrix(ii,jj+1) = 1;
        end
        end
        if jj > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
        end
    end
end

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of void phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Change scarce matrix to get rid of void phase in phase matrix
% Get rid of left and right boundary as they have no effect
matrix_scarcey(1:m,1) = 0;
% Left Boundary
matrix_scarcey(1:m,m) = 0;
% Right boundary
matrix_scarcex(1,1:m) = 0;
% Left Boundary
matrix_scarcex(1,m:m) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_yv;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_xv;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary



% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 1:m
    for jjj = 1:m

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = -(iii-y_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2));
```

```
            matrix_scarcex (iii,jjj) = abs((jjj-x_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2)));
            end
            if jjj >= m/2
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2)));
            end

        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end

matrix_scarcey(matrix_scarcey == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
% matrix_scarcex(matrix_scarcex == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
Stress_GBy = matrix_scarcey.*Stress_GB_unmodifiedy;
% Normal stress around phase boundary X Original stress state present at
boundaries
Stress_GBx = matrix_scarcex.*Stress_GB_unmodifiedx;
% Normal stress around phase boundary X Original stress state present at
boundaries
end
sigma_thin = sigma/(G_Sx/G_S);
% Increase in stress due to thinning
w = (G_S-(G_S-2*(r_pv*10^-8)))/(G_S);
sigma1 = sigma_thin/(1-w);
if r_p1v <= 49
sigma2 = sigma_thin*(1+(100-sum(Sigma_yv(1,ceil(r_p1v)+1:m-
ceil(r_p1v))))/sum(Sigma_yv(1,ceil(r_p1v)+1:m-ceil(r_p1v))));
else
    sigma2 = sigma_thin/(1-w);
end
NVC_plus =
NVC_Eq*((exp(((sigma2.*(Stress_GBy+Stress_GBx))*Omega)/(k*T_max))+1)/2); %
Molar concentration of vacancies increased due to temsile stress
```

```
%% Matrix formation
% Boundary Conditions
for xx = 1:m
    for yy = 1:m
if P_matrix(xx,yy) == 1
C(xx,yy) =  NVC_plus(xx,yy);
end
if P_matrix(xx,yy) == 3
    C(xx,yy) = NVC_Eq;
end
if P_matrix(xx,yy) == 0
C(xx,yy) = C_new(xx,yy);
end
    end
end

% Corner boundary conditions
% Average out corners
C(1,1)=(NVC_Eq+NVC_plus(1,1))/2;
C(m,1)=(NVC_Eq+NVC_plus(m,1))/2;
C(1,m)=(NVC_Eq+NVC_plus(1,m))/2;
C(m,m)=(NVC_Eq+NVC_plus(m,m))/2;


for i = 1:m
% For loop in x direction
for j = 1:m
% For loop in y direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in x direction
                    D*dt/(dy^2)*(Sigma_yv(i+1,j)-
2*Sigma_yv(i,j)+Sigma_yv(i-1,j))*(Av_no/E_mol)*Omega+...
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1))+...
% flow in y direction
                    D*dt/(dx^2)*(Sigma_xv(i,j+1)-
2*Sigma_xv(i,j)+Sigma_xv(i,j-1))*(Av_no/E_mol)*Omega;
     C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end

P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_vector1 = find(P_matrix(1:m,1)==1);
P_vector2 = find(P_matrix(1:m,m)==1);
P_matrix2(P_vector1(2:end-1),1) = 0;
% Make sides zero
P_matrix2(P_vector2(2:end-1),m) = 0;
% Make sides zero

for ii = 1:m
% Defining faces for flux
    for jj = 1:m
```

```
        if ii <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
% Layer above void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii+1,jj) = (C(ii+1,jj));
            end
        end
        if jj >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj-1)== 0
% Layer left of void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj-1) = (C(ii,jj-1));
            end
        end
        if ii >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
% Layer below void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii-1,jj) = (C(ii-1,jj));
            end
        end
        if jj <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj+1)== 0
% Layer right of void
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj+1) = (C(ii,jj+1));
            end
        end
    end
end

Flux_x = D*((sum(C(P_vector1,1))+sum(C(P_vector2,m)))-
(sum(C(P_vector1,2))+sum(C(P_vector2,m-1))))/(dx)...  % Flux to side
boundaries for thinning, Ficks 1st law J = D dC/dx
            *(dt)...
% Time step
            *(Av_no);

Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Ficks 1st law J = D dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration times Avogadro's number for number of vacancies



strain_y =    sum(V_Flux) ...
% Strain at each time step.
              *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
              *m^2;
% Account for size of matrix to give strain

strain_x =    Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
              *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
```

```
            *m^2;
% Account for size of matrix

Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction

G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_S;

t=t+1;
end

hold on
t = 1:t;
% Time vector
NH = dt*((D*sigma*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation

xlabel('Time (min)','FontSize',20)
ylabel('Strain','FontSize',20)
plot(t,Strain_y,'r',t,NH,'k','LineWidth',1.4); %,'LineWidth',1.4);%
set(gca,'FontSize',15)
% Axes font size
xlabel('Time (Minutes)','FontSize',20)
ylabel('Strain','FontSize',20)
legendd = legend('Model Void Damage','N-H Theory');
set(legendd,'FontSize',18)
grid on
```

A.9 I_Creep_Precipitate_Void.m

```
%% Precipitate and void in the same system

clc
clear all
% close all

%% Constants - Stress fields
m = 100;
% Size of matrix
v_1 = 0.3;
% Poisson's Ratio Matrix
v_2 = 0.3;
% Poisson's Ratio Precipitate
E_1 = 200;
% Young's Modulus Matrix (iron)
E_2 = 430;
% Young's Modulus Precipitate (carbide)

v_1v = 0.3;
% Poisson's Ratio Matrix
v_2v = 0.3;
% Poisson's Ratio Precipitate
E_1v = 200;
% Young's Modulus Matrix
E_2v = 0;
% Young's Modulus Precipitate
%% Constants- Coarsening
N1 = 1;
tmax = 100000-1;
K_p = 5*10^-28;
% Constant
Q_p = 286.8;
% Activation energy
R_g =8.31451;
% Gas Constant
T = 600+293;
% Thermodynamic Temperature

%% Constants

G_S = 1*10^-6;
% Grain Size
G_Sx = G_S;
G_Sy = G_S;
dx = G_S/(m-1);
% size of each element x dir
dy = G_S/(m-1);
% size of each element y dir
D = 2.85*10^-19;
% Diffusion coefficient
T_input = 600;
% input temp in degrees Celsius
T_max = 273+T_input;
% Temperature in Kelvin

E_mol = 5.4*10^3;
%% Vacancy Concentrations Constants
```

```
A_v = 3;
% Factor for vacancy formation
Av_no = 6.02*10^23;
% Avogadro's number
E_v = 135000;
% Formation Enthalpy for FCC
k = 1.3806488*10^-23;
% Boltzmann Constant
sigma = 30*10^6;
% Applied Stress
sigma1 = sigma;
Omega = 1.18*10^-29;
% Volume of atom

%% Void growth constants
d_gb = 0.5*10^-9;
% grain boundary thickness
D_gb = D/(exp(-125500/(R_g*T_max)));
% diffusivity of grain boundary
% P = hydrostatic pressure if any?
lambda = G_S;
% distance between voids
r_p1v = 2;
%% Phase Matrix
% the creation of a phase matrix defines different phases present in the
% grain, such as precipitates, matrix, grain boundary and interphase
% boundary(considered same as grain boundary for now)

% Matrix == 0
% Grain boundary == 1
% Phase boundary == 1
% Voids will be == 3 (gets rid of an indexing error to make voids 3 as
well)
% Precipitate == 3

P_matrix = zeros(m,m);

% representing the grain boundary
P_matrix(1,1:m) = 1;
% Top boundary
P_matrix(m,1:m) = 1;
% Bottom boundary
P_matrix(1:m,1) = 1;
% Left Boundary
P_matrix(1:m,m) = 1;
% Right boundary
P_matrix_2 = P_matrix;
% Duplicate matrix for later use in while loop
%% Pre Calculations
% factors account for adding two different field properties together
k_y =   (E_2*((3-v_2)*E_1+(5+v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% ky factor
k_x =   (E_2*((3*v_2-1)*E_1+(1-3*v_1)*E_2))...
        /((E_1+2*E_2)^2-(v_2*E_1+(1-v_1)*E_2)^2);
% kx factor

k_yv =   (E_2v*((3-v_2v)*E_1v+(5+v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% ky factor
```

```
k_xv =   (E_2v*((3*v_2v-1)*E_1v+(1-3*v_1v)*E_2v))...
        /((E_1v+2*E_2v)^2-(v_2v*E_1v+(1-v_1v)*E_2v)^2);
% kx factor

dt = 60;
% Starting points

NVC_Eq = A_v*exp((-E_v/Av_no)/(k*T_max));
% Equilibrium Molar concentration of vacancies
C = NVC_Eq.*ones(m,m);
% Blank
C_new = C;
% Blank
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
Flux_y = zeros(1,m);
% Blank
V_Flux = zeros(1,m);
% Blank
Strain_y = zeros(1,tmax);
% Blank
Strain_x = zeros(1,tmax);
Sigma_x = zeros(m,m);
% Blank
Sigma_y = zeros(m,m);
% Blank
Sigma_xv = zeros(m,m);
% Blank
Sigma_yv = zeros(m,m);
% Blank
Sigma_xt = zeros(m,m);
% Blank
Sigma_yt = zeros(m,m);
% Blank
% Sigma_xy = zeros(m,m);
% Blank
%% Imbedded FORloop to Calculate Stress Matrix
% Current consideration for 1, 2 or 3 precipitates on each boundary, later
% plans to write up for any amount of precipitates
% a = 0;
N = N1;
% Number of precipitates on top and bottom boundary
t = 1;

while t <= tmax
    % While loop for coarsening

P_matrix = P_matrix_2;
% reset matrix back to original
Face_1m = zeros(m,m);
% Blank
Face_2m = zeros(m,m);
% Blank
% If there is no precipitate at the boundary
if N == 0

Stress_GBy = zeros(m,m);
% Uniform blank field
```

```
Stress_GBx = zeros(m,m);
Stress_GBy(1,1:m) = 1;
% Stress at top face
Stress_GBy(m,1:m) = 1;
% Stress at bottom face
r_pv = 10;
a = 0;
% Random constant for bypassing code

% If there is only one precipitate at the boundary
elseif N == 1
a = 1;
% Random constant for bypassing code
r_pv = r_p1v+(D_gb*d_gb*(sigma1)*Omega*dt)/(2*k*T_max*G_Sx*r_p1v)*10^12;%
r_p1v = r_pv;

r_p = (sqrt((m*m*0.1)/pi/2));
% Radius of Precipitate
Vol = 2*pi*(r_p)^2;
% Total volume of precipitates
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p2 = ((Vol/pi)- r_p1^2)^(1/2);
% Shrinking precipitate
% need if statement for stopping coarsening^^^

if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to halt coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end

for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = m/2+0.00001;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = m/2+0.00001;
        y_pos2 = 1;

        x_posv = 1;
% x position of void
        y_posv = m;
% y position of void
        x_pos2v = 1;
        y_pos2v = 1;
        x_pos3v = m;
        y_pos3v = m;
        x_pos4v = m;
        y_pos4v = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
```

```
        rv = sqrt((x-x_posv).^2 +(y-y_posv).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2v = sqrt((x-x_pos2v).^2 +(y-y_pos2v).^2);
        r3v = sqrt((x-x_pos3v).^2 +(y-y_pos3v).^2);
        r4v = sqrt((x-x_pos4v).^2 +(y-y_pos4v).^2);

        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;

        Fv = (8*(y-y_posv).^2.*(3.*r_pv.^2+2.*(y-y_posv).^2))/rv.^4;
% Recurring Constant
        F2v = (8*(y-y_pos2v).^2.*(3.*r_pv.^2+2.*(y-y_pos2v).^2))/r2v.^4;
        F3v = (8*(y-y_pos3v).^2.*(3.*r_pv.^2+2.*(y-y_pos3v).^2))/r3v.^4;
        F4v = (8*(y-y_pos4v).^2.*(3.*r_pv.^2+2.*(y-y_pos4v).^2))/r4v.^4;
        Gv = (24.*(r_pv.^2.*(y-y_posv).^4))/rv.^6;
% Recurring Constant
        G2v = (24.*(r_pv.^2.*(y-y_pos2v).^4))/r2v.^6;
        G3v = (24.*(r_pv.^2.*(y-y_pos3v).^4))/r3v.^6;
        G4v = (24.*(r_pv.^2.*(y-y_pos4v).^4))/r4v.^6;
        %%
        if r >= r_p1 && r2 >= r_p2
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3
        end
        if r_pv == 0
            continue
        elseif rv >= r_pv && r2v >= r_pv && r3v >= r_pv && r4v >= r_pv
% Conditions for Stress Field Outside Void
            Sigma_yv(y,x) =  ((1-
k_yv+k_xv)*r_pv^2)/(2*rv^2)*(1+(3*r_pv^2+10*(y-y_posv)^2)/rv^2-Fv+Gv)-
(k_xv*r_pv^2)/rv^2*(1-(2*(y-y_posv)^2)/rv^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(1+(3*r_pv^2+10*(y-y_pos2v)^2)/r2v^2-F2v+G2v)-
(k_xv*r_pv^2)/r2v^2*(1-(2*(y-y_pos2v)^2)/r2v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(1+(3*r_pv^2+10*(y-y_pos3v)^2)/r3v^2-F3v+G3v)-
(k_xv*r_pv^2)/r3v^2*(1-(2*(y-y_pos3v)^2)/r3v^2)...
```

```
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(1+(3*r_pv^2+10*(y-y_pos4v)^2)/r4v^2-F4v+G4v)-
(k_xv*r_pv^2)/r4v^2*(1-(2*(y-y_pos4v)^2)/r4v^2);
            Sigma_xv(y,x) =  ((1-k_yv+k_xv)*r_pv^2)/(2*rv^2)*(3-
(3*r_pv^2+18*(y-y_posv)^2)/rv^2+Fv-Gv)+(k_xv*r_pv^2)/rv^2*(1-(2*(y-
y_posv)^2)/rv^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(3-
(3*r_pv^2+18*(y-y_pos2v)^2)/r2v^2+F2v-G2v)+(k_xv*r_pv^2)/r2v^2*(1-(2*(y-
y_pos2v)^2)/r2v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(3-
(3*r_pv^2+18*(y-y_pos3v)^2)/r3v^2+F3v-G3v)+(k_xv*r_pv^2)/r3v^2*(1-(2*(y-
y_pos3v)^2)/r3v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(3-
(3*r_pv^2+18*(y-y_pos4v)^2)/r4v^2+F4v-G4v)+(k_xv*r_pv^2)/r4v^2*(1-(2*(y-
y_pos4v)^2)/r4v^2);
        else
% Stress Field Inside Void
            Sigma_yv(y,x) = NaN;
            Sigma_xv(y,x) = NaN;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3

        end

    end
end
for yy = 1:m
% Y vector
    for xx = 1:m
        if P_matrix(yy,xx)==1 || P_matrix(yy,xx)== 0
            Sigma_yt(yy,xx) = Sigma_y(yy,xx) + Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx) + Sigma_xv(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_y(yy,xx) == k_y
            Sigma_yt(yy,xx) = Sigma_y(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_yv(yy,xx) ~= 0
            Sigma_yt(yy,xx) = Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_xv(yy,xx);
        end
    end
end
% Defining phase boundary with grain boundary value
for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        end
        if ii > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        end
```

```
        if jj < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        end
        if jj > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
        end
    end
end
if a == 1

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(2:m-1,1) = 0;
% Left Boundary
matrix_scarcey(2:m-1,m) = 0;
% Right boundary
matrix_scarcex(1,2:m-1) = 0;
% Left Boundary
matrix_scarcex(m,2:m-1) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_yt;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_xt;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 1:m
    for jjj = 1:m
        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= m/2+r_p1+1 && jjj >= m/2-r_p1-1
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj <= 1+r_pv
```

```
            matrix_scarcey (iii,jjj) = -(iii-y_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2)));
            end
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2)));
            end
        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= m/2+r_p2+1 && jjj >= m/2-r_p2-1
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj <= 1+r_pv
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2)));
            end
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2)));
            end
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end
% If there is two precipitates at the boundary

elseif N == 2
a = 2;

r_pv = r_p1v+(D_gb*d_gb*(sigma1)*Omega*dt)/(2*k*T_max*G_Sx*r_p1v)*10^12;%
r_p1v = r_pv;

r_p = sqrt((m*m*0.1)/pi/4);
% Radius of Precipitate
Vol = 4*pi*(r_p)^2;
% Total volume of precipitates
r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p2 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
r_p3 = ((Vol/2/pi)- r_p1^2)^(1/2);
r_p4 = ((Vol/2/pi)- r_p2^2)^(1/2);
```

```
if r_p1 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi/2);
    r_p3 = 0;
end
if r_p2 >= sqrt(((m*m*0.1))/pi/2)
    r_p2 = sqrt(((m*m*0.1))/pi/2);
    r_p4 = 0;
end

if r_p3 == 0 && r_p4 == 0
    r_p1 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p2 = ((Vol/pi)- r_p1^2)^(1/2);
if r_p1 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p1 = sqrt(((m*m*0.1))/pi);
    r_p2 = 0;
end
end



for y = 1:m
% Y vector
    for x = 1:m
% X vector
        %%
        x_pos = round((m-r_p*4)/3+r_p)+0.0001;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = round((m-r_p*4)/3+r_p)+0.0001;
        y_pos2 = 1;
        x_pos3 = round(m-((m-r_p*4)/3+r_p))+0.0001;
        y_pos3 = m;
        x_pos4 = round(m-((m-r_p*4)/3+r_p))+0.0001;
        y_pos4 = 1;

        x_posv = 1;
% x position of precipitate
        y_posv = m;
% y position of precipitate
        x_pos2v = 1;
        y_pos2v = 1;
        x_pos3v = m;
        y_pos3v = m;
        x_pos4v = m;
        y_pos4v = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);

        rv = sqrt((x-x_posv).^2 +(y-y_posv).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2v = sqrt((x-x_pos2v).^2 +(y-y_pos2v).^2);
        r3v = sqrt((x-x_pos3v).^2 +(y-y_pos3v).^2);
        r4v = sqrt((x-x_pos4v).^2 +(y-y_pos4v).^2);
        %%
```

```
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;

        Fv = (8*(y-y_posv).^2.*(3.*r_pv.^2+2.*(y-y_posv).^2))/rv.^4;
% Recurring Constant
        F2v = (8*(y-y_pos2v).^2.*(3.*r_pv.^2+2.*(y-y_pos2v).^2))/r2v.^4;
        F3v = (8*(y-y_pos3v).^2.*(3.*r_pv.^2+2.*(y-y_pos3v).^2))/r3v.^4;
        F4v = (8*(y-y_pos4v).^2.*(3.*r_pv.^2+2.*(y-y_pos4v).^2))/r4v.^4;
        Gv = (24.*(r_pv.^2.*(y-y_posv).^4))/rv.^6;
% Recurring Constant
        G2v = (24.*(r_pv.^2.*(y-y_pos2v).^4))/r2v.^6;
        G3v = (24.*(r_pv.^2.*(y-y_pos3v).^4))/r3v.^6;
        G4v = (24.*(r_pv.^2.*(y-y_pos4v).^4))/r4v.^6;
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4
% Conditions for Stress Field Outside Precipitate
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2);
%
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2);
        else
% Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Precipitate phase represented with value 3

        end
        if r_pv == 0
            continue
```

```
        elseif rv >= r_pv && r2v >= r_pv && r3v >= r_pv && r4v >= r_pv
% Conditions for Stress Field Outside Void
            Sigma_yv(y,x) =  ((1-
k_yv+k_xv)*r_pv^2)/(2*rv^2)*(1+(3*r_pv^2+10*(y-y_posv)^2)/rv^2-Fv+Gv)-
(k_xv*r_pv^2)/rv^2*(1-(2*(y-y_posv)^2)/rv^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(1+(3*r_pv^2+10*(y-y_pos2v)^2)/r2v^2-F2v+G2v)-
(k_xv*r_pv^2)/r2v^2*(1-(2*(y-y_pos2v)^2)/r2v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(1+(3*r_pv^2+10*(y-y_pos3v)^2)/r3v^2-F3v+G3v)-
(k_xv*r_pv^2)/r3v^2*(1-(2*(y-y_pos3v)^2)/r3v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(1+(3*r_pv^2+10*(y-y_pos4v)^2)/r4v^2-F4v+G4v)-
(k_xv*r_pv^2)/r4v^2*(1-(2*(y-y_pos4v)^2)/r4v^2);
            Sigma_xv(y,x) =  ((1-k_yv+k_xv)*r_pv^2)/(2*rv^2)*(3-
(3*r_pv^2+18*(y-y_posv)^2)/rv^2+Fv-Gv)+(k_xv*r_pv^2)/rv^2*(1-(2*(y-
y_posv)^2)/rv^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(3-
(3*r_pv^2+18*(y-y_pos2v)^2)/r2v^2+F2v-G2v)+(k_xv*r_pv^2)/r2v^2*(1-(2*(y-
y_pos2v)^2)/r2v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(3-
(3*r_pv^2+18*(y-y_pos3v)^2)/r3v^2+F3v-G3v)+(k_xv*r_pv^2)/r3v^2*(1-(2*(y-
y_pos3v)^2)/r3v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(3-
(3*r_pv^2+18*(y-y_pos4v)^2)/r4v^2+F4v-G4v)+(k_xv*r_pv^2)/r4v^2*(1-(2*(y-
y_pos4v)^2)/r4v^2);
        else
% Stress Field Inside Void
            Sigma_yv(y,x) = NaN;
            Sigma_xv(y,x) = NaN;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3

        end

    end

end
for yy = 1:m
% Y vector
    for xx = 1:m
        if P_matrix(yy,xx)==1 || P_matrix(yy,xx)== 0
            Sigma_yt(yy,xx) = Sigma_y(yy,xx) + Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx) + Sigma_xv(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_y(yy,xx) == k_y
            Sigma_yt(yy,xx) = Sigma_y(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_yv(yy,xx) ~= 0
            Sigma_yt(yy,xx) = Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_xv(yy,xx);
        end
    end
end

for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
```

```
        if ii < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        end
        if ii > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
        end
        if jj < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        end
        if jj > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
        end
    end
end
if a == 2
matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;

% Get rid of left and right boundary as they have no effect
matrix_scarcey(2:m-1,1) = 0;
% Left Boundary
matrix_scarcey(2:m-1,m) = 0;
% Right boundary
matrix_scarcex(1,2:m-1) = 0;
% Left Boundary
matrix_scarcex(m,2:m-1) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_yt;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_xt;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary
```

```
% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= ((m-r_p*4)/3+r_p)+r_p1+1 && jjj >= ((m-r_p*4)/3+r_p)-
r_p1-1                                                  % Left half
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj <= (m-((m-r_p*4)/3+r_p))+r_p3+1 && jjj >= (m-((m-
r_p*4)/3+r_p))-r_p3-1                                                  %
Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end
            if jjj <= 1+r_pv
            matrix_scarcey (iii,jjj) = -(iii-y_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2)));
            end
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= ((m-r_p*4)/3+r_p)+r_p1+1 && jjj >= ((m-r_p*4)/3+r_p)-
r_p1-1                                                  % Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
           end
            if jjj <= (m-((m-r_p*4)/3+r_p))+r_p3+1 && jjj >= (m-((m-
r_p*4)/3+r_p))-r_p3-1                                                  %
Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end
            if jjj <= 1+r_pv
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2)));
            end
```

```
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2)));
            end
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end

% If there is three precipitates at the boundary

elseif N == 3
a = 3;

r_pv = r_p1v+(D_gb*d_gb*(sigma1)*Omega*dt)/(2*k*T_max*G_Sx*r_p1v)*10^12;%
r_p1v = r_pv;

r_p = sqrt((m*m*0.1)/pi/6);
% Radius of Precipitate
Vol = 6*pi*(r_p)^2;
% Total volume of precipitates
r_p4 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Growing precipitate
r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
% Shrinking precipitate
r_p2 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p1 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);
r_p6 = ((Vol/(4*pi))- (r_p4^2)/2)^(1/2);
r_p5 = ((Vol/(4*pi))- (r_p3^2)/2)^(1/2);

if r_p4 >= sqrt(((m*m*0.1))/pi/2)
% If statement to half coarsening if only one precipitate left
    r_p4 = sqrt(((m*m*0.1))/pi/2);
    r_p2 = 0;
    r_p6 = 0;
end
if r_p3 >= sqrt(((m*m*0.1))/pi/2)
    r_p3 = sqrt(((m*m*0.1))/pi/2);
    r_p1 = 0;
    r_p5 = 0;
end
if r_p1 == 0 && r_p2 == 0 && r_p5 == 0 && r_p6 == 0
    r_p3 = ((K_p.*t*dt+(r_p*10^-6)^3).^(1/3))*10^6;
    r_p4 = ((Vol/pi)- r_p3^2)^(1/2);
if r_p3 >= sqrt(((m*m*0.1))/pi)
% If statement to half coarsening if only one precipitate left
    r_p3 = sqrt(((m*m*0.1))/pi);
    r_p4 = 0;
end
end
for y = 1:m         % Y vector
    for x = 1:m     % X vector
        %%
```

```
        x_pos = (m-r_p*6)/4+r_p;
% x position of precipitate
        y_pos = m;
% y position of precipitate
        x_pos2 = (m-r_p*6)/4+r_p;
        y_pos2 = 1;
        x_pos3 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos3 = m;
        x_pos4 = ((m-r_p*6)/4+r_p)*2+r_p;
        y_pos4 = 1;
        x_pos5 = m-((m-r_p*6)/4+r_p);
        y_pos5 = m;
        x_pos6 = m-((m-r_p*6)/4+r_p);
        y_pos6 = 1;

        x_posv = 1;
% x position of precipitate
        y_posv = m;
% y position of precipitate
        x_pos2v = 1;
        y_pos2v = 1;
        x_pos3v = m;
        y_pos3v = m;
        x_pos4v = m;
        y_pos4v = 1;
        %%
        r = sqrt((x-x_pos).^2 +(y-y_pos).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2 = sqrt((x-x_pos2).^2 +(y-y_pos2).^2);
        r3 = sqrt((x-x_pos3).^2 +(y-y_pos3).^2);
        r4 = sqrt((x-x_pos4).^2 +(y-y_pos4).^2);
        r5 = sqrt((x-x_pos5).^2 +(y-y_pos5).^2);
        r6 = sqrt((x-x_pos6).^2 +(y-y_pos6).^2);

        rv = sqrt((x-x_posv).^2 +(y-y_posv).^2);
% Distance from centre of precipitate to coordinate(x,y)
        r2v = sqrt((x-x_pos2v).^2 +(y-y_pos2v).^2);
        r3v = sqrt((x-x_pos3v).^2 +(y-y_pos3v).^2);
        r4v = sqrt((x-x_pos4v).^2 +(y-y_pos4v).^2);
        %%
        F = (8*(y-y_pos).^2.*(3.*r_p1.^2+2.*(y-y_pos).^2))/r.^4;
% Recurring Constant
        F2 = (8*(y-y_pos2).^2.*(3.*r_p2.^2+2.*(y-y_pos2).^2))/r2.^4;
        F3 = (8*(y-y_pos3).^2.*(3.*r_p3.^2+2.*(y-y_pos3).^2))/r3.^4;
        F4 = (8*(y-y_pos4).^2.*(3.*r_p4.^2+2.*(y-y_pos4).^2))/r4.^4;
        F5 = (8*(y-y_pos5).^2.*(3.*r_p5.^2+2.*(y-y_pos5).^2))/r5.^4;
        F6 = (8*(y-y_pos6).^2.*(3.*r_p6.^2+2.*(y-y_pos6).^2))/r6.^4;
        G = (24.*(r_p1.^2.*(y-y_pos).^4))/r.^6;
% Recurring Constant
        G2 = (24.*(r_p2.^2.*(y-y_pos2).^4))/r2.^6;
        G3 = (24.*(r_p3.^2.*(y-y_pos3).^4))/r3.^6;
        G4 = (24.*(r_p4.^2.*(y-y_pos4).^4))/r4.^6;
        G5 = (24.*(r_p5.^2.*(y-y_pos5).^4))/r5.^6;
        G6 = (24.*(r_p6.^2.*(y-y_pos6).^4))/r6.^6;

        Fv = (8*(y-y_posv).^2.*(3.*r_pv.^2+2.*(y-y_posv).^2))/rv.^4;
% Recurring Constant
        F2v = (8*(y-y_pos2v).^2.*(3.*r_pv.^2+2.*(y-y_pos2v).^2))/r2v.^4;
        F3v = (8*(y-y_pos3v).^2.*(3.*r_pv.^2+2.*(y-y_pos3v).^2))/r3v.^4;
        F4v = (8*(y-y_pos4v).^2.*(3.*r_pv.^2+2.*(y-y_pos4v).^2))/r4v.^4;
```

```
        Gv = (24.*(r_pv.^2.*(y-y_posv).^4))/rv.^6;
% Recurring Constant
        G2v = (24.*(r_pv.^2.*(y-y_pos2v).^4))/r2v.^6;
        G3v = (24.*(r_pv.^2.*(y-y_pos3v).^4))/r3v.^6;
        G4v = (24.*(r_pv.^2.*(y-y_pos4v).^4))/r4v.^6;
        %%
        if r >= r_p1 && r2 >= r_p2 && r3 >= r_p3 && r4 >= r_p4 ...
% Conditions for Stress Field Outside Precipitate
                && r5 >= r_p5 && r6 >= r_p6
            Sigma_y(y,x) =  1 ...
                            +((1-
k_y+k_x)*r_p1^2)/(2*r^2)*(1+(3*r_p1^2+10*(y-y_pos)^2)/r^2-F+G)-
(k_x*r_p1^2)/r^2*(1-(2*(y-y_pos)^2)/r^2)...
                            +((1-
k_y+k_x)*r_p2^2)/(2*r2^2)*(1+(3*r_p2^2+10*(y-y_pos2)^2)/r2^2-F2+G2)-
(k_x*r_p2^2)/r2^2*(1-(2*(y-y_pos2)^2)/r2^2)...
                            +((1-
k_y+k_x)*r_p3^2)/(2*r3^2)*(1+(3*r_p3^2+10*(y-y_pos3)^2)/r3^2-F3+G3)-
(k_x*r_p3^2)/r3^2*(1-(2*(y-y_pos3)^2)/r3^2)...
                            +((1-
k_y+k_x)*r_p4^2)/(2*r4^2)*(1+(3*r_p4^2+10*(y-y_pos4)^2)/r4^2-F4+G4)-
(k_x*r_p4^2)/r4^2*(1-(2*(y-y_pos4)^2)/r4^2)...;
                            +((1-
k_y+k_x)*r_p5^2)/(2*r5^2)*(1+(3*r_p5^2+10*(y-y_pos5)^2)/r5^2-F5+G5)-
(k_x*r_p5^2)/r5^2*(1-(2*(y-y_pos5)^2)/r5^2)...;
                            +((1-
k_y+k_x)*r_p6^2)/(2*r6^2)*(1+(3*r_p6^2+10*(y-y_pos6)^2)/r6^2-F6+G6)-
(k_x*r_p6^2)/r6^2*(1-(2*(y-y_pos6)^2)/r6^2);
            Sigma_x(y,x) =  ((1-k_y+k_x)*r_p1^2)/(2*r^2)*(3-
(3*r_p1^2+18*(y-y_pos)^2)/r^2+F-G)+(k_x*r_p1^2)/r^2*(1-(2*(y-
y_pos)^2)/r^2)...
                            +((1-k_y+k_x)*r_p2^2)/(2*r2^2)*(3-
(3*r_p2^2+18*(y-y_pos2)^2)/r2^2+F2-G2)+(k_x*r_p2^2)/r2^2*(1-(2*(y-
y_pos2)^2)/r2^2)...
                            +((1-k_y+k_x)*r_p3^2)/(2*r3^2)*(3-
(3*r_p3^2+18*(y-y_pos3)^2)/r3^2+F3-G3)+(k_x*r_p3^2)/r3^2*(1-(2*(y-
y_pos3)^2)/r3^2)...
                            +((1-k_y+k_x)*r_p4^2)/(2*r4^2)*(3-
(3*r_p4^2+18*(y-y_pos4)^2)/r4^2+F4-G4)+(k_x*r_p4^2)/r4^2*(1-(2*(y-
y_pos4)^2)/r4^2)...
                            +((1-k_y+k_x)*r_p5^2)/(2*r5^2)*(3-
(3*r_p5^2+18*(y-y_pos5)^2)/r5^2+F5-G5)+(k_x*r_p5^2)/r5^2*(1-(2*(y-
y_pos5)^2)/r5^2)...
                            +((1-k_y+k_x)*r_p6^2)/(2*r6^2)*(3-
(3*r_p6^2+18*(y-y_pos6)^2)/r6^2+F6-G6)+(k_x*r_p6^2)/r6^2*(1-(2*(y-
y_pos6)^2)/r6^2);
        else            % Stress Field Inside Precipitate
            Sigma_y(y,x) = k_y;
            Sigma_x(y,x) = k_x;
            P_matrix(y,x) = 3;
        end
        if r_pv == 0
            continue
        elseif rv >= r_pv && r2v >= r_pv && r3v >= r_pv && r4v >= r_pv
% Conditions for Stress Field Outside Void
            Sigma_yv(y,x) =  ((1-
k_yv+k_xv)*r_pv^2)/(2*rv^2)*(1+(3*r_pv^2+10*(y-y_posv)^2)/rv^2-Fv+Gv)-
(k_xv*r_pv^2)/rv^2*(1-(2*(y-y_posv)^2)/rv^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(1+(3*r_pv^2+10*(y-y_pos2v)^2)/r2v^2-F2v+G2v)-
(k_xv*r_pv^2)/r2v^2*(1-(2*(y-y_pos2v)^2)/r2v^2)...
```

```
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(1+(3*r_pv^2+10*(y-y_pos3v)^2)/r3v^2-F3v+G3v)-
(k_xv*r_pv^2)/r3v^2*(1-(2*(y-y_pos3v)^2)/r3v^2)...
                            +((1-
k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(1+(3*r_pv^2+10*(y-y_pos4v)^2)/r4v^2-F4v+G4v)-
(k_xv*r_pv^2)/r4v^2*(1-(2*(y-y_pos4v)^2)/r4v^2);
            Sigma_xv(y,x) =  ((1-k_yv+k_xv)*r_pv^2)/(2*rv^2)*(3-
(3*r_pv^2+18*(y-y_posv)^2)/rv^2+Fv-Gv)+(k_xv*r_pv^2)/rv^2*(1-(2*(y-
y_posv)^2)/rv^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r2v^2)*(3-
(3*r_pv^2+18*(y-y_pos2v)^2)/r2v^2+F2v-G2v)+(k_xv*r_pv^2)/r2v^2*(1-(2*(y-
y_pos2v)^2)/r2v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r3v^2)*(3-
(3*r_pv^2+18*(y-y_pos3v)^2)/r3v^2+F3v-G3v)+(k_xv*r_pv^2)/r3v^2*(1-(2*(y-
y_pos3v)^2)/r3v^2)...
                            +((1-k_yv+k_xv)*r_pv^2)/(2*r4v^2)*(3-
(3*r_pv^2+18*(y-y_pos4v)^2)/r4v^2+F4v-G4v)+(k_xv*r_pv^2)/r4v^2*(1-(2*(y-
y_pos4v)^2)/r4v^2);
        else
% Stress Field Inside Void
            Sigma_yv(y,x) = NaN;
            Sigma_xv(y,x) = NaN;
            % Define precipitate phase in P_matrix
            P_matrix(y,x) = 3;
% Void phase represented with value 3

        end
    end
end
for yy = 1:m
% Y vector
    for xx = 1:m
        if P_matrix(yy,xx)==1 || P_matrix(yy,xx)== 0
            Sigma_yt(yy,xx) = Sigma_y(yy,xx) + Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx) + Sigma_xv(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_y(yy,xx) == k_y
            Sigma_yt(yy,xx) = Sigma_y(yy,xx);
            Sigma_xt(yy,xx) = Sigma_x(yy,xx);
        elseif P_matrix(yy,xx)== 3 && Sigma_yv(yy,xx) ~= 0
            Sigma_yt(yy,xx) = Sigma_yv(yy,xx);
            Sigma_xt(yy,xx) = Sigma_xv(yy,xx);
        end
    end
end
for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii+1,jj)~= 3
% Layer above precipitate
           P_matrix(ii+1,jj) = 1;
        end
        end
        if ii > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii-1,jj)~= 3
% Layer below precipitate
           P_matrix(ii-1,jj) = 1;
        end
```

```
        end
        if jj < m % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj+1)~= 3
% Layer right of precipitate
           P_matrix(ii,jj+1) = 1;
        end
        end
        if jj > 2 % again getting rid of errors caused by really small
precipitates/voids
        if P_matrix (ii,jj)== 3 && P_matrix(ii,jj-1)~= 3
% Layer left of precipitate
           P_matrix(ii,jj-1) = 1;
        end
        end
    end
end
else
    display('too many precipitates for current works')
end
if a == 3

matrix_scarcey = P_matrix;
% Duplicate phase matrix to save original setup
matrix_scarcex = P_matrix;
matrix_scarcey(matrix_scarcey ==3) = 0;
% Change scarce matrix to get rid of precipitate phase in phase matrix
matrix_scarcex(matrix_scarcex ==3) = 0;
% Get rid of left and right boundary as they have no effect
matrix_scarcey(2:m-1,1) = 0;
% Left Boundary
matrix_scarcey(2:m-1,m) = 0;
% Right boundary
matrix_scarcex(1,2:m-1) = 0;
% Left Boundary
matrix_scarcex(m,2:m-1) = 0;
% Right boundary
Stress_GB_unmodifiedy = matrix_scarcey.*Sigma_yt;
% Define the current stress state only at the top and bottom boundaries,
phase boundary included
Stress_GB_unmodifiedx = matrix_scarcex.*Sigma_xt;
% Remove boundary values for modification to normal forces
matrix_scarcey(1,1:m) = 0;
% Top boundary
matrix_scarcey(m,1:m) = 0;
% Bottom boundary
matrix_scarcex(1,1:m) = 0;
% Top boundary
matrix_scarcex(m,1:m) = 0;
% Bottom boundary

% Resolving scarce matrix so that only the normal force to the circular
precipitate is represented
for iii = 2:m-1
    for jjj = 2:m-1

        if matrix_scarcey (iii,jjj)== 1 && iii >= m/2
% Phase boundary in top half
            if jjj <= ((m-r_p*6)/4+r_p)+r_p1+1 && jjj >= ((m-r_p*6)/4+r_p)-
r_p1-1                                          % Lower third
```

```
            matrix_scarcey (iii,jjj) = -(iii-y_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos)/(sqrt((iii-
y_pos)^2+(jjj-x_pos)^2)));
            end
            if jjj <= (((m-r_p*6)/4+r_p)*2+r_p)+r_p3+1 && jjj >= (((m-
r_p*6)/4+r_p)*2+r_p)-r_p3-1                                  % Middle
third
            matrix_scarcey (iii,jjj) = -(iii-y_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3)/(sqrt((iii-
y_pos3)^2+(jjj-x_pos3)^2)));
            end
            if jjj <= (m-((m-r_p*6)/4+r_p))+r_p4+1 && jjj >= (m-((m-
r_p*6)/4+r_p))-r_p4-1                                        %
Upper third
            matrix_scarcey (iii,jjj) = -(iii-y_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos5)/(sqrt((iii-
y_pos5)^2+(jjj-x_pos5)^2)));
            end
            if jjj <= 1+r_pv
            matrix_scarcey (iii,jjj) = -(iii-y_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_posv)/(sqrt((iii-
y_posv)^2+(jjj-x_posv)^2)));
            end
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = -(iii-y_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos3v)/(sqrt((iii-
y_pos3v)^2+(jjj-x_pos3v)^2)));
            end

        elseif matrix_scarcey (iii,jjj)== 1 && iii <= m/2
% Phase boundary in bottom half
            if jjj <= ((m-r_p*6)/4+r_p)+r_p2+1 && jjj >= ((m-r_p*6)/4+r_p)-
r_p2-1                                                     % Lower third
            matrix_scarcey (iii,jjj) = (iii-y_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2)/(sqrt((iii-
y_pos2)^2+(jjj-x_pos2)^2)));
            end
            if jjj <= (((m-r_p*6)/4+r_p)*2+r_p)+r_p4+1 && jjj >= (((m-
r_p*6)/4+r_p)*2+r_p)-r_p4-1                                  % Middle
third
            matrix_scarcey (iii,jjj) = (iii-y_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos4)/(sqrt((iii-
y_pos4)^2+(jjj-x_pos4)^2)));
            end
            if jjj <= (m-((m-r_p*6)/4+r_p))+r_p6+1 && jjj >= (m-((m-
r_p*6)/4+r_p))-r_p6-1                                        %
Upper third
            matrix_scarcey (iii,jjj) = (iii-y_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos6)/(sqrt((iii-
y_pos6)^2+(jjj-x_pos6)^2)));
            end
```

```
            if jjj <= 1+r_pv
% Left half
            matrix_scarcey (iii,jjj) = (iii-y_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2));
            matrix_scarcex (iii,jjj) = abs((jjj-x_pos2v)/(sqrt((iii-
y_pos2v)^2+(jjj-x_pos2v)^2)));
            end
            if jjj >= m-r_pv
% Right half
            matrix_scarcey (iii,jjj) = (iii-y_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2));
            matrix_scarcex (iii,jjj) = abs(-(jjj-x_pos4v)/(sqrt((iii-
y_pos4v)^2+(jjj-x_pos4v)^2)));
            end
        else
% Conditions not met then progress to next step of loop
            continue
        end
    end
end
end

if a == 1 ||a == 2||a == 3
matrix_scarcey(matrix_scarcey == 0) = 1;
% Replace all 0's with 1 so that multiplication doesnt effect matrix or
straight boundaries
% matrix_scarcex(matrix_scarcex == 0) = 1;
Stress_GBy = matrix_scarcey.*Stress_GB_unmodifiedy;
% Normal stress around phase boundary X Original stress state present at
boundaries
Stress_GBx = matrix_scarcex.*Stress_GB_unmodifiedx;

end
% Vacancy Concentrations
% mole fractions
sigma_thin = sigma/(G_Sx/G_S);
w = (m-(m-2*r_pv))/m;
sigma1 = sigma_thin/(1-w);
if r_p1v <= 49
sigma2 = sigma_thin*(1+(100-sum(Sigma_yt(1,ceil(r_p1v)+1:m-
ceil(r_p1v))))/sum(Sigma_yt(1,ceil(r_p1v)+1:m-ceil(r_p1v))));
else
    sigma2 = sigma_thin/(1-w);
end
NVC_plus =
NVC_Eq*((exp(((sigma1.*(Stress_GBy+Stress_GBx))*Omega)/(k*T_max))+1)/2); %
Molar concentration of vacancies increased due to tensile stress
(adding x and y fields)

%% Matrix formation
% Boundary Conditions
for xx = 1:m
    for yy = 1:m
if P_matrix(xx,yy) == 1
C(xx,yy) =  NVC_plus(xx,yy);
end
if P_matrix(xx,yy) == 3
    C(xx,yy) = NaN;
end
if P_matrix(xx,yy) == 0
C(xx,yy) = C_new(xx,yy);
```

```
end
    end
end


% Corner boundary conditions
% Average out corners
C(1,1)=(NVC_Eq+NVC_plus(1,1))/2;
C(m,1)=(NVC_Eq+NVC_plus(m,1))/2;
C(1,m)=(NVC_Eq+NVC_plus(1,m))/2;
C(m,m)=(NVC_Eq+NVC_plus(m,m))/2;


for i = 1:m
% For loop in x direction
for j = 1:m
% For loop in y direction
    if P_matrix(i,j) == 0
    C_new(i,j)  =   C(i,j)+...
% Original Matix
                    D*dt/(dy^2)*(C(i+1,j)-2*C(i,j)+C(i-1,j)) +...
% flow in x direction
                    D*dt/(dy^2)*(Sigma_yt(i+1,j)-
2*Sigma_yt(i,j)+Sigma_yt(i-1,j))*(Av_no/E_mol)*Omega+...
                    D*dt/(dx^2)*(C(i,j+1)-2*C(i,j)+C(i,j-1))+...
% flow in y direction
                    D*dt/(dx^2)*(Sigma_xt(i,j+1)-
2*Sigma_xt(i,j)+Sigma_xt(i,j-1))*(Av_no/E_mol)*Omega;
    C(i,j)      =   C_new(i,j);
% Renaming matrix for next iteration
    else
        continue
    end
end
end

P_matrix2 = P_matrix;
% Replicating matrix to avoid any issues
P_vector1 = find(P_matrix(1:m,1)==1);
P_vector2 = find(P_matrix(1:m,m)==1);
P_matrix2(P_vector1(2:end-1),1) = 0;
% Make sides zero
P_matrix2(P_vector2(2:end-1),m) = 0;
% Make sides zero
for ii = 1:m
% Defining the phase boundary, given the same value as Grain boundary ( =1
)
    for jj = 1:m
        if ii <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii+1,jj)== 0
% Layer above precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii+1,jj) = (C(ii+1,jj));
            end
        end
        if jj >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj-1)== 0
% Layer left of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj-1) = (C(ii,jj-1));
```

```
            end
        end
        if ii >= 2
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii-1,jj)== 0
% Layer below precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii-1,jj) = (C(ii-1,jj));
            end
        end
        if jj <= m-1
            if P_matrix2 (ii,jj)== 1 && P_matrix2(ii,jj+1)== 0
% Layer right of precipitate
                Face_1m(ii,jj) = (C(ii,jj));
                Face_2m(ii,jj+1) = (C(ii,jj+1));
            end
        end
    end
end

% A non-perfect correction for when voids and precipitates are one element
apart, without
% it the code would become unstable at that point due to lost data.
for ij = 2:m-1
    for ji = 2:m-1
        if      Face_1m (ij,ji)== 0 && Face_1m(ij,ji-1)~= 0 &&
Face_1m(ij,ji+1) ~= 0 && Face_1m(ij+1,ji) == 0
                    Face_2m(ij,ji) =  Face_2m(ij,ji).*2;
        end
    end
end


Flux_x = D*((sum(C(P_vector1,1))+sum(C(P_vector2,m)))-
(sum(C(P_vector1,2))+sum(C(P_vector2,m-1))))/(dx)*(dt)*(Av_no);

Flux_y =  D*((sum(Face_1m)-sum(Face_2m)))/(dy)...
% Ficks 1st law J = D dC/dx
            *dt;
% adding time step for accelerating creep

        V_Flux = Flux_y*Av_no;
% Molar concentration times Avogadro's number for number of vacancies

strain_y =    sum(V_Flux) ...
% Strain at each time step. Total vacancy flux divided by number of points
                *Omega/G_Sy^2 ...
% Volume of each vacancy divided by area of face
                *m^2;
% Account for size of matrix to give strain

strain_x =    Flux_x ...
% Strain at each time step. Total vacancy flux divided by number of points
                *Omega/G_Sx^2 ...
% Volume of each vacancy divided by area of face
                *m^2;
% Account for size of matrix

Strain_y(t+1)= Strain_y(t) + strain_y;
% Total strain over time in y direction
```

```
Strain_x(t+1)= Strain_x(t) + strain_x;
% Total strain over time in x direction

G_Sy1 = G_S*(1+Strain_y(t+1));
G_Sx1 = G_S*(1+Strain_x(t+1));
dy = G_Sy1/(m-1);
dx = G_Sx1/(m-1);
G_Sy = G_Sy1;
G_Sx = G_Sx1;

t=t+1;

end


%% Plot surface

% x =0:dx:G_Sx;
% Range of x(0,2) and specifying the grid points
% y =0:dy:G_Sy;
% Range of y(0,2) and specifying the grid points
% surf(x,y,C)
% % shading interp
% title({['2-D Diffusion with {\nu} = ',num2str(D)];['time (\itt) =
',num2str(t)]})
% xlabel('Spatial co-ordinate (x) \rightarrow')
% ylabel('{\leftarrow} Spatial co-ordinate (y)')
% zlabel('Transport property profile (u) \rightarrow')
%% Plot strain vs theoretical strain

hold on
t = 1:t;
% Time vector
NH = dt*((D*sigma*Omega)/(k*T_max*(G_S^2)))*t;
% NH equation
plot(t,Strain_y,'r',t,NH,'k','LineWidth',1.4);
% Plot theory vs model
% title({'STRAIN Vs TIME'},'FontSize',20)
xlabel('Time (min)','FontSize',20)
ylabel('Strain','FontSize',20)
legendd = legend('Model Void Damage','N-H Theory');
set(legendd,'FontSize',12)
grid on
```

Appendix B: Ethics Form

EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zakiya Chikte (Zakiya.chitke@uct.ac.za; New EBE Building, Ph 021 650 5739).

NB: A copy of this signed form must be included with the thesis/dissertation/report when it is submitted for examination

Name of Principal Researcher/Student: Royden Weyer **Department:** Mechanical Engineering

If a Student: **Degree:** MSc. Eng.(Materials) **Supervisor:** Prof Robert Knutsen

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: Modelling of Damage due to Diffusional Creep in High Chromium Steels

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO ✓
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES	NO ✓
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	NO ✓
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO ✓

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	Date
Principal Researcher/Student:	Royden Luke Weyer Signed by candidate ~~Signature Removed~~	22/04/2016

This application is approved by:

Supervisor (if applicable):	Signed by candidate ~~Signatures Removed~~	22/04/2016
HOD (or delegated nominee): *Final authority for all assessments with NO to all questions and for all undergraduate research.*	Signed by candidate Robert. D. Knutsen	22/04/2016
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		

ADDENDUM 1:
Please append a copy of the research proposal here, as well as any interview schedules or questionnaires:

ADDENDUM 2: To be completed if you answered YES to Question 2:

It is assumed that you have read the UCT Code for Research involving Human Subjects (available at http://web.uct.ac.za/depts/educate/download/uctcodeforresearchinvolvinghumansubjects.pdf) in order to be able to answer the questions in this addendum.

2.1 Does the research discriminate against participation by individuals, or differentiate between participants, on the grounds of gender, race or ethnic group, age range, religion, income, handicap, illness or any similar classification?	YES	NO
2.2 Does the research require the participation of socially or physically vulnerable people (children, aged, disabled, etc) or legally restricted groups?	YES	NO
2.3 Will you not be able to secure the informed consent of all participants in the research? (In the case of children, will you not be able to obtain the consent of their guardians or parents?)	YES	NO
2.4 Will any confidential data be collected or will identifiable records of individuals be kept?	YES	NO
2.5 In reporting on this research is there any possibility that you will not be able to keep the identities of the individuals involved anonymous?	YES	NO
2.6 Are there any foreseeable risks of physical, psychological or social harm to participants that might occur in the course of the research?	YES	NO
2.7 Does the research include making payments or giving gifts to any participants?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

ADDENDUM 3: To be completed if you answered YES to Question 3:

3.1 Is the community expected to make decisions for, during or based on the research?	YES	NO
3.2 At the end of the research will any economic or social process be terminated or left unsupported, or equipment or facilities used in the research be recovered from the participants or community?	YES	NO
3.3 Will any service be provided at a level below the generally accepted standards?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

ADDENDUM 4: To be completed if you answered YES to Question 4

4.1 Is there any existing or potential conflict of interest between a research sponsor, academic supervisor, other researchers or participants?	YES	NO
4.2 Will information that reveals the identity of participants be supplied to a research sponsor, other than with the permission of the individuals?	YES	NO
4.3 Does the proposed research potentially conflict with the research of any other individual or group within the University?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues: